# Enhancing Privacy, Integrity and Security in Federated Learning: Intel SGX-based Multi-Party Computation Platform

XIN JIN
xiji@kth.se

10th August 2023

## 1 Thesis title

Enhancing Privacy, Integrity and Security in Federated Learning: Intel SGX-based Multi-Party Computation Platform

## 2 Background

With the booming development of large dataset collection and deep learning algorithm's success into people's life, Federated Learning(FL), which enables guests to train the desired effective model without sharing privacy with other clients, is raising more and more importance at various scenarios[1]. However, certain deep learning models, such as Generative Adversarial Networks (GANs), pose privacy risks in FL, as they can inadvertently reveal sensitive information from separate devices [2]. This poses a potential threat to users' privacy and necessitates the development of robust privacy-preserving techniques.

### Privacy Challenges in Federated Learning

For example, the loss function employed in certain deep learning models can inadvertently reveal logic information, providing insights into the correctness of an attacker's modified inputs. This presents a privacy risk in scenarios where malicious actors intentionally misrepresent data. For instance, if an image belonging to Alice is deliberately declared as belonging to Eve, the local victim device may continuously contribute Alice's information to correct the model, leading to unintended data leakage. This underscores the need for robust privacy-preserving techniques in FL to prevent such unauthorized access and protect the privacy of individual users' data. Privacy-preserving techniques are essential to safeguard the privacy of individual users and maintain the integrity of the FL ecosystem.

### Multiple Party Computation

To bolster privacy and security in FL, one effective technique is Multi-Party Computation (MPC). MPC enables secure computation between multiple parties without disclosing sensitive data [3]. Although MPC involves some overhead due to interaction protocol security, its speed and reliability make it an attractive option for FL when compared to alternatives like Homomorphic Encryption (HE) and Differential Privacy(DP)[4].

### Trusted Execution Environment(TEEs) - Intel SGX

Despite the advantages of MPC, it can still suffer from communication overhead, especially in scenarios involving large datasets and complex machine learning models. Trusted Execution Environments (TEEs), such as Intel Software Guard Extensions (SGX), offer a robust solution to ensure privacy-preserving execution [5]. TEEs provide a secure enclave where cryptography operations and sensitive functions can be executed without exposure to external entities, including the trusted server CPU itself[6]. SGX allows for the isolation of privacy data and code, thus preventing unauthorized access and potential information leakage[7]. Intel SGX provides a separated area for privacy data and code, which can not be accessed outside the SGX secure enclave, that the unencrypted data and functions inside SGX enclave are not visible to any entity including the trusted server CPU itself holding the SGX.

## 3 Research question

How can an SGX-based framework be designed and implemented to enable secure multi-party computation (MPC) for basic machine learning functions, such as Maxpool and ReLU, using data from multiple parties? Additionally, how does SGX accelerate and simplify MPC secure computations?

During the thesis work, the research will explore possibilities to optimize the security and speed of trusted MPC machine learning from the following perspectives:

- Investigate the integration of GPU integrated technology to identify the faster and most efficient SGX secure approach for machine learning[8].
- Evaluate methods to enhance output privacy in the SGX-based MPC framework[9].

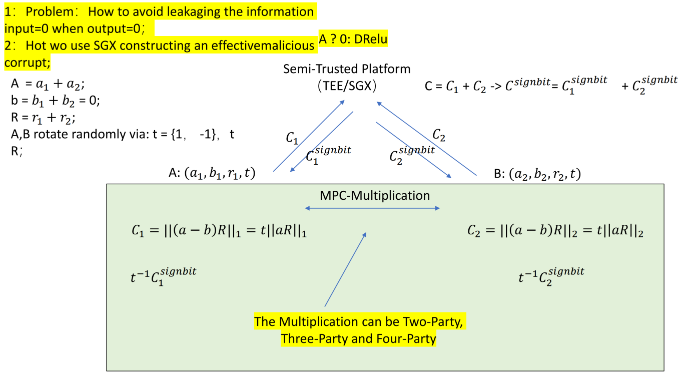
1：Problem：How to avoid leakaging the information input=0 when output=0；
2：Hot wo use SGX constructing an effectivemalicious corrupt;
A ? 0: DRelu
Semi-Trusted Platform (TEE/SGX)
$C = C_1 + C_2$ -> $C^{signbit} = C_1^{signbit} + C_2^{signbit}$
A = $a_1 + a_2$;
b = $b_1 + b_2$ = 0;
R = $r_1 + r_2$;
A,B rotate randomly via: t = {1， -1}， t R；
$C_1$
$C_1^{signbit}$
$C_2$
$C_2^{signbit}$
A: $(a_1, b_1, r_1, t)$
B: $(a_2, b_2, r_2, t)$
MPC-Multiplication
$C_1 = ||(a-b)R||_1 = t||aR||_1$
$C_2 = ||(a-b)R||_2 = t||aR||_2$
$t^{-1}C_1^{signbit}$
$t^{-1}C_2^{signbit}$
The Multiplication can be Two-Party, Three-Party and Four-Party

Figure 1: The output privacy idea diagram

- Explore techniques for tamper-proof authentication in the context of secure multi-party computation for machine learning.

## 4 Hypothesis

- The SGX implementation would dynamically suit for 2-party, 3-party and 4-party MPC protocol.
- There would be only one trusted party working as an SGX server. And the other parties are assumed as malicious/semi-honest.
- The outcome of the project would be an SGX version platform for MPC Machine Learning preparing for FL. And optimizations of security/integrity/speed for SGX Machine Learning Computation if available.

# 5 Research method

Since the main session of the project is to develop the SGX Machine Learning MPC Platform firstly, the most important project method would be C++ and C++ traditional code optimization methods, such as how to modify the SGX code easier to inherit. The security and performance of SGX code.

The second challenge is to optimize the algorithms from different aspects mentioned in module 3, the methods can be engineering such as GPU acceleration, or applied mathematical such as cryptography operations hiding possible leakage information. The overall performance of the project would be measured by the memory size, execution time and attack face analysis.

There are mainly 5 steps:

- MPC Protocol selection and design. Present pseudo-code of the whole process when necessary.
- Code design: Classes and Methods.
- Implementation.
- Benchmarking. To compare with pure CPU/GPU solutions, or make pure technical analysis such as the memory space usage, execution time, security level etc.
- Optimization and extensions as mentioned in Module 3.

## 5.1 The project code design

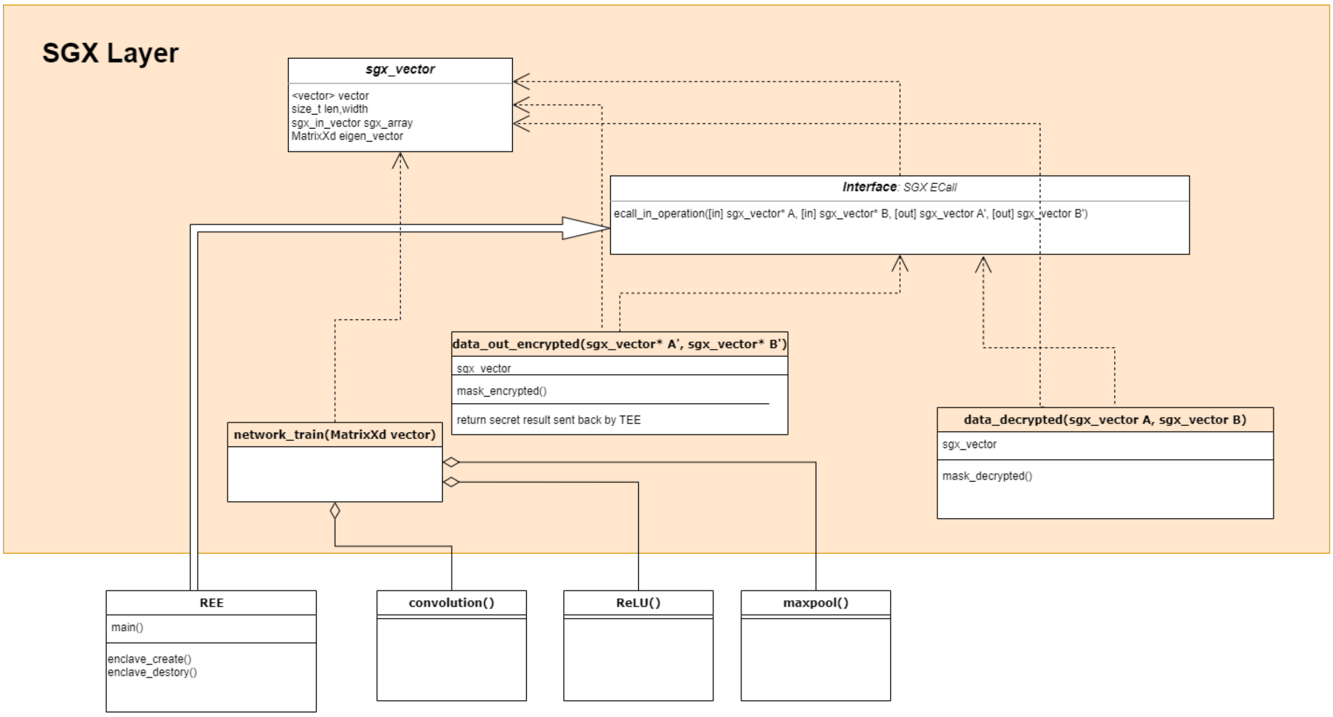
SGX Layer
sgx_vector
<vector> vector
size_t len,width
sgx_in_vector sgx_array
MatrixXd eigen_vector
Interface: SGX ECall
ecall_in_operation([in] sgx_vector* A, [in] sgx_vector* B, [out] sgx_vector A', [out] sgx_vector B')
data_out_encrypted(sgx_vector* A', sgx_vector* B')
sgx_vector
mask_encrypted()
return secret result sent back by TEE
network_train(MatrixXd vector)
data_decrypted(sgx_vector A, sgx_vector B)
sgx_vector
mask_decrypted()
REE
main()
enclave_create()
enclave_destory()
convolution()
ReLU()
maxpool()

Figure 2: The project code design

## 5.2 The project structure

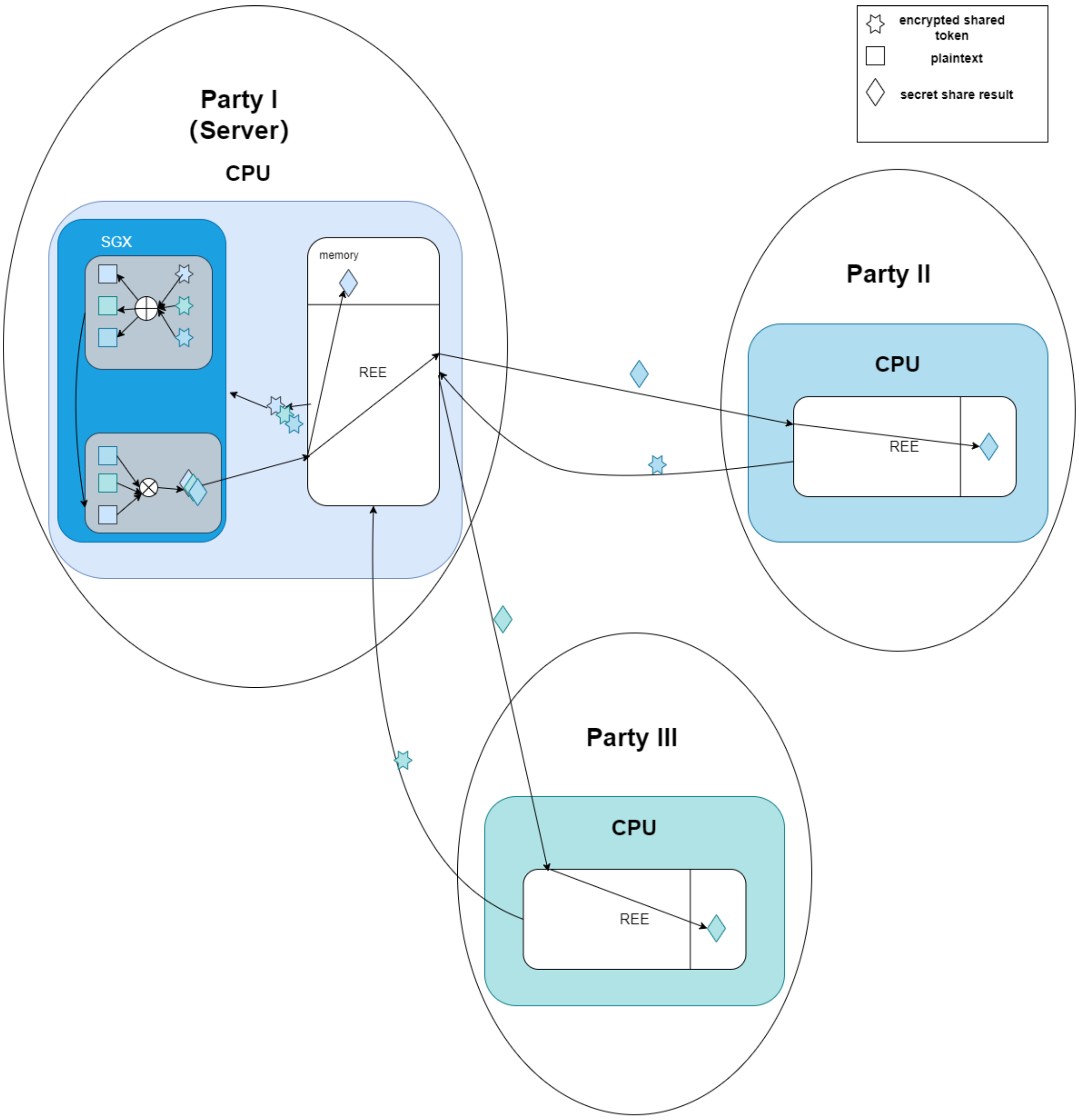

Figure 3: The project structure overview

## 5.3 Raising the derivative of ReLU(DReLU) as an example

### 2-Parties Computation

Assume the condition of 2-Party Privacy Computation, let $P_0, P_1$ participate the computation. $P_0$ works as the trusted server which holds the SGX TEEs, that the privacy share data from different parties would be sent to SGX at $P_0$ to compute the desired result. DReLU(x) is the desired result, whose inputs are privacy share data tokens, and return the result persistently in a share data token form to the parties. x $\in \mathbf{R}^{m,n}$ is a two-dimensional privacy data matrix, and $P_0$ holds the privacy data share token $x^0 \in \mathbf{R}^{m,n}$, and $P_1$ holds the privacy data share token $x^1$, such that:

$$x_{ij} = x_{ij}^0 + x_{ij}^1 \tag{1}$$

### DReLU function[10]

$$\forall x \in \mathbf{R}^{m,n}, \quad DReLU(x) = \begin{cases} 1, & x_{ij} \geq 0, \\ 0, & x_{ij} < 0. \end{cases} \tag{2}$$

**MPC protocol for DReLU**

$$\langle x\rangle_L \text{ denotes the share data token sent and stored at 2 parties.} \tag{3}$$

$$\Pi_{DReLU} \text{ is the MPC protocol to compute the derivative of Rectified Linear Unit(DReLU).} \tag{4}$$
$$(\Pi_{DReLU}(\langle x\rangle_L) \rightarrow \langle \Pi_{DReLU}(x)\rangle_L) \text{ is the MPC privacy computation process,} \tag{5}$$
$$\text{where } \Pi_{DReLU}(\langle x\rangle_L) \text{ is the computation process for share data ,} \tag{6}$$
$$\text{and } \langle \Pi_{DReLU}(x)\rangle_L \text{ are the desired share data results returning back to the 2 parties.} \tag{7}$$

The share data token $\langle x\rangle_L$ sent to the trusted server would be:

$$\langle x\rangle_L = \alpha x \tag{8}$$

$$\langle x^0\rangle_L = \alpha x^0 + r_0 \tag{9}$$

$$\langle x^1\rangle_L = \alpha x^1 + r_1 \tag{10}$$

where $r_0$ and $r_1$ are the randomized numbers that:

$$r_0 + r_1 = 0,$$

and $\alpha \in R^+$ is also randomized. Because $\alpha$ and $r$ are unknown as random numbers, inputs $\langle x\rangle_L$ originally stored at $P_0$ and $P_1$ would be privacy data secreted for parties. And the result of DReLU will keep unchanged:

$$\begin{aligned}
&DReLU(x) = DReLU(x_0 + x_1)\\
&= DReLU(\alpha x_0 + \alpha x_1)\\
&= DReLU(\alpha x_0 + \alpha x_1 + r_0 + r_1) \qquad where \quad r_0 + r_1 = 0\\
&= DReLU(\alpha x_0 + r_0 + \alpha x_1 + r_1)\\
&= DReLU(\langle x^0\rangle_L + \langle x^1\rangle_L)\\
&= DReLU(\langle x\rangle_L)
\end{aligned}$$

Then split $DReLU(\langle x\rangle_L)$ randomly to:

$$DReLU(\langle x\rangle_L) = DReLU(\langle x\rangle_L)^0 + DReLU(\langle x\rangle_L)^1$$

and the returned result share data tokens are $\langle y^0\rangle_L$ and $\langle y^1\rangle_L$ such that:

$$\begin{aligned}
\langle y^0\rangle_L &= DReLU(\langle x\rangle_L)^0 + r_0'\\
\langle y^1\rangle_L &= DReLU(\langle x\rangle_L)^1 + r_1'
\end{aligned}$$

where $r_0$’ and $r_1$’ are random numbers that $r_0' + r_1' = 0$

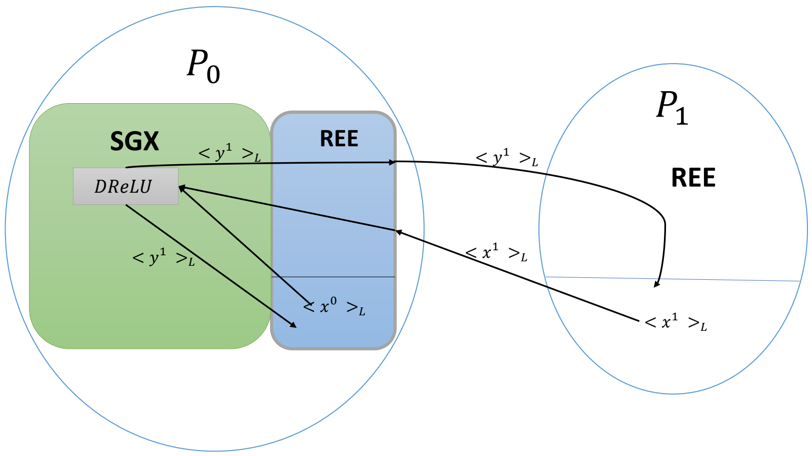

Figure 4: MPC encrypted protocol for DReLU of 2 Party Computation.pdf

## 6 Background of the student

The student obtained double-degree Bachelor of Computer Science and Mathematics, and has completed Electrical Engineering master courses study at TU Berlin and KTH, that she has good programming and mathematical skills for the cryptography engineering project.

## 7 Supervisor at the company

Dr. Dai Tianxiang, he will supervise, direct and co-operate with Jin Xin's project.
Dr. Yong Li, he will supervise, organize and lead Jin Xin's project.

## 8 Suggested examiner at KTH

My degree project examiner is Johan Håstad. Prof.Dr. Johan has agreed to be my examiner.

## 9 Suggested supervisor at KTH

My degree project supervisor is Ming Xiao. Prof.Dr. Xiao has agreed to be my degree project supervisor.

## 10 Resources

Server with Intel® Xeon® Platinum Processor 8360Y
Ubuntu 22.04 LTS, GCC-13
Nvidia RTX A5000 GPU

## 11 Eligibility

Yes, I have completed all master courses study to start the degree project.

# 12 Study Planning

I have completed all the master courses study and obtained enough course credits, except master thesis.

# References

[1] P. M. Mammen, “Federated learning: Opportunities and challenges.” [Online]. Available: http://arxiv.org/abs/2101.05428

[2] B. Hitaj, G. Ateniese, and F. Perez-Cruz, “Deep models under the GAN: Information leakage from collaborative deep learning.” [Online]. Available: http://arxiv.org/abs/1702.07464

[3] O. Goldreich, “Secure multi-party computation.”

[4] R. Kanagavelu, Z. Li, J. Samsudin, Y. Yang, F. Yang, R. S. M. Goh, M. Cheah, P. Wiwatphonthana, K. Akkarajitsakul, and S. Wangz, “Two-phase multi-party computation enabled privacy-preserving federated learning.” [Online]. Available: http://arxiv.org/abs/2005.11901

[5] J. I. Choi and K. R. B. Butler, “Secure multiparty computation and trusted hardware: Examining adoption challenges and opportunities,” vol. 2019, pp. 1–28. doi: 10.1155/2019/1368905. [Online]. Available: https://www.hindawi.com/journals/scn/2019/1368905/

[6] S. Felsen, Kiss, T. Schneider, and C. Weinert, “Secure and private function evaluation with intel SGX,” in *Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop*. ACM. doi: 10.1145/3338466.3358919. ISBN 978-1-4503-6826-1 pp. 165–181. [Online]. Available: https://dl.acm.org/doi/10.1145/3338466.3358919

[7] S. Hui, Y. Zhang, A. Hu, and E. Song, “Horizontal federated learning and secure distributed training for recommendation system with intel sgx,” *arXiv preprint arXiv:2207.05079*, 2022.

[8] Y. Jie, Y. Ren, Q. Wang, Y. Xie, C. Zhang, L. Wei, and J. Liu, “Multi-party secure computation with intel SGX for graph neural networks,” in *ICC 2022 - IEEE International Conference on Communications*. doi: 10.1109/ICC45855.2022.9839282 pp. 528–533, ISSN: 1938-1883.

[9] E. Bresson, D. Catalano, N. Fazio, A. Nicolosi, and M. Yung, “Output privacy in secure multiparty computation.”

[10] Y. Akimoto, K. Fukuchi, Y. Akimoto, and J. Sakuma, “Privformer: Privacy-preserving transformer with MPC,” in *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*. doi: 10.1109/EuroSP57164.2023.00031 pp. 392–410.