

Poster: Secure SqueezeNet inference in 4 minutes

Ehud Aharoni 


IBM Research - Haifa, IL

Moran Baruch 

IBM Research - Haifa, IL and
Bar Ilan University, IL

Nir Drucker 


IBM Research - Haifa, IL

Gilad Ezov 

IBM Research - Haifa, IL

Eyal Kushnir 

IBM Research - Haifa, IL

Guy Moshkovich 

IBM Research - Haifa, IL

Omri Soceanu 

IBM Research - Haifa, IL

Abstract—Privacy-preserving machine learning (PPML) solutions often use multi-party computation or client-assisted homomorphic encryption (HE) techniques, which require a substantial communication overhead. In contrast, non-interactive solutions are considered slow and are only practical for small neural networks or those with limited security guarantees. We show that, for the first time, it is possible to evaluate a large HE-friendly SqueezeNet model on large images in a non-interactive setting using HE with 128-bit security parameters and no reductions in the number of layers. This evaluation takes only 4 minutes when running on a GPU and 6 minutes when running on a CPU.

Index Terms—Fully Homomorphic Encryption, HEaAN, CKKS, SqueezeNet, Privacy Preserving Machine Learning, Machine Learning Inference, Tile Tensors, HELayers

I. INTRODUCTION

Practical non-interactive privacy-preserving machine learning solutions are useful when outsourcing sensitive data to a third-party cloud environment. Such environments often require adherence to privacy regulations such as the GDPR [1]. However, these solutions are hard to design and implement because they cannot use standard multi-party computation techniques. homomorphic encryption (HE) is one cryptographic method that does not require extra communication beyond the input and output.

HE is an encryption scheme that allows the evaluation of any algorithm on encrypted data [2]. Common HE schemes involve four methods: *Gen*, *Enc*, *Dec*, *Eval*. The *Gen* function generates a secret-key public-key pair. The *Enc* function uses the public key to encrypt a message m , which can be a vector $m[s]$ of s integer elements, and returns a ciphertext. Its “inverse” is the *Dec* function, which receives a ciphertext and returns an s -dimensional vector. We distinguish between exact HE schemes, where $m = Dec(Enc(m))$, and approximate HE schemes such as CKKS [3], where $Dec(Enc(m)) = m + \epsilon$, for a small error ϵ .

The *Eval* function receives a function and a vector of ciphertexts, and evaluates the function on these ciphertexts. Specifically, it allows us to perform the operations *Add*, *Mul*, and *Rot*, which are defined as

$$\begin{aligned} Dec(Add(Enc(m_1), Enc(m_2))) &= m_1 + m_2 \\ Dec(Mul(Enc(m_1), Enc(m_2))) &= m_1 * m_2 \\ Dec(Rot(Enc(m_1), n))[i] &= m_1((i + n) \bmod s) \end{aligned}$$

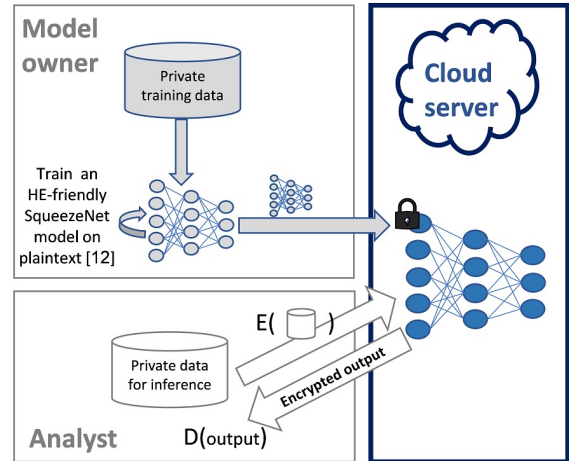


Figure 1: A typical flow for running a neural network (NN) over HE spans over three entities: a model owner, a cloud server, and an analyst. This work considers the cloud operation.

Using HE to evaluate complex functions that require extensive computations is still considered slow and thus impractical. For example, it would not be practical to evaluate a classification algorithm that involves running a deep NN model on encrypted data due to the lengthy time required for inference. One reason for this is that when using HE, only a limited number of operations are allowed on ciphertexts before an expensive bootstrapping operation is needed. Recently, two works demonstrated the use of HE without bootstrapping on medium-size networks: CHET [4] uses SqueezeNet-CIFAR [5], and HELayers [6] uses AlexNet [7]. All inference evaluations run in less than 10 minutes on these networks.

Recently, the performance of the CKKS bootstrapping was drastically improved, which allowed [8] to evaluate an even larger network: ResNet-20 using CKKS in 10,602 seconds (176 minutes) but with only 111.6 bits of security. In this work, we combine the fast CKKS bootstrap implementation of HEaAN [9] with the AI optimizations provided by the HELayers library [6], [10] to achieve the first HE-Friendly full SqueezeNet [11], [12] implementation with 128-bit security. This network requires 40 multiplications, which is much more than the ~ 20 multiplications that are required for the AlexNet

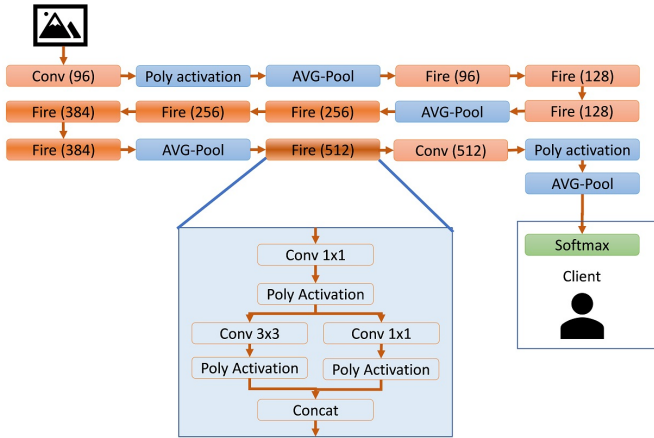


Figure 2: An HE-Friendly SqueezeNet with one convolutional layer, 8 fire layers, and 4 average pooling layers. It receives an encrypted image and returns an encrypted vector to the user, who decrypts it and performs the last Softmax layer to get the classification results.

and SqueezeNet-CIFAR networks; thus, it requires the use of bootstrap operations. Figure 2 demonstrates the HE-Friendly SqueezeNet architecture. We also evaluated the network on large images of size $224 \times 224 \times 3$, which are larger than the $32 \times 32 \times 3$ CIFAR-10 images used in other works. Our results show that it only takes about 4 minutes to perform network inference over encrypted data. To the best of our knowledge, this is the first practical demonstration of a non-interactive large NN being evaluated over encrypted data with 128-bit security.

Threat model Our threat-model involves three entities: An AI model owner, a cloud server that performs model inference on HE encrypted data using the pre-computed AI model, and an analyst that sends confidential data to the cloud for model inference. See Figure 1 for an illustration. We assume that the model owner allows the cloud to see its model but the users’ data remains private. In addition, any communication between all entities is encrypted using a secure network protocol such as TLS 1.3. Finally, we assume that the cloud is honest-but-curious, i.e., it evaluates the functions provided by the model owner and users without any deviation. Our threat model does not consider privacy attacks, where the users try to extract the model training data through the inference results.

II. EXPERIMENTS

For the experiments, we considered two platforms: 1) an A100 SXM4 80 GB GPU on a server with an AMD® EPYC 7763 2.45GHz machine with 64 cores (128 threads) and 750 GB memory; 2) An Intel® Xeon® CPU E5-2699 v4 @ 2.20GHz machine with 44 cores (88 threads) and 750 GB memory.

We integrated HEaAN [9] into HELayers [6], which provides us with a bootstrap implementation of CKKS [3] for CPUs as well as for GPUs. We configured HELayers (and thus

HEaAN) with parameters that target 128-bit security. All the reported results are the average of 10 runs. We used the trained HE-friendly SqueezeNet model from [12]; this model has an accuracy of 82% when running on datasets with encrypted images of size $224 \times 224 \times 3$.

On platform 1, initializing the HE context took 3 seconds and it took 55 seconds to load the model. These pre-computation steps can be cached and used for future inference operations. The inference operation itself took 4.07 minutes and used 60 GB of RAM. On platform 2, the initialization and loading time took 5 and 21 seconds, respectively, and the inference operation took 6 minutes. On platform 2, we also configured HELayers to run a batch of 16 samples. Here, the inference operation took 39 minutes with an amortized latency (throughput) of only 2.3 minutes.

Our work demonstrates that using non-interactive HE for large tasks is practical. In the future, we aim to continue and test the practicality of non-interactive HE-based solutions on even deeper architectures.

REFERENCES

- [1] EU General Data Protection Regulation, “Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation),” *Official Journal of the European Union*, vol. 119, 2016. [Online]. Available: <http://data.europa.eu/eli/reg/2016/679/oj>
- [2] S. Halevi, “Homomorphic Encryption,” in *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, Y. Lindell, Ed. Cham: Springer International Publishing, 2017, pp. 219–276.
- [3] J. Cheon, A. Kim, M. Kim, and Y. Song, “Homomorphic encryption for arithmetic of approximate numbers,” in *Proceedings of Advances in Cryptology - ASIACRYPT 2017*. Springer Cham, 11 2017, pp. 409–437.
- [4] R. Dathathri, O. Saarikivi, H. Chen, K. Laine, K. Lauter, S. Maleki, M. Musuvathi, and T. Mytkowicz, “Chet: An optimizing compiler for fully-homomorphic neural-network inferencing,” in *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, ser. PLDI 2019. Association for Computing Machinery, 2019, p. 142–156.
- [5] D. Corvoysier, “Experiment with SqueezeNets, commit:2619f730b4e91313057039feb81788c5648e3951,” 2017. [Online]. Available: <https://github.com/kaizouman/tensorsandbox/tree/master/cifar10/models/squeeze>
- [6] E. Aharoni, A. Adir, M. Baruch, N. Drucker, G. Ezov, A. Farkash, L. Greenberg, R. Masalha, G. Moshkovich, D. Murik, H. Shaul, and O. Soceanu, “HELayers: A tile tensors framework for large neural networks on encrypted data,” *CoRR*, vol. abs/2011.01805, 2020.
- [7] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” *Neural Information Processing Systems*, vol. 25, 01 2012.
- [8] J.-W. Lee, H. Kang, Y. Lee, W. Choi, J. Eom, M. Deryabin, E. Lee, J. Lee, D. Yoo, Y.-S. Kim, and J.-S. No, “Privacy-preserving machine learning with fully homomorphic encryption for deep neural network,” *IEEE Access*, vol. 10, pp. 30039–30054, 2022.
- [9] CryptoLab, “HEaAN: Homomorphic Encryption for Arithmetic of Approximate Numbers, version 3.1.4,” 2022. [Online]. Available: <https://www.cryptolab.co.kr/eng/product/heaan.php>
- [10] IBM, “HELayers SDK with a Python API for x86,” 2021. [Online]. Available: <https://hub.docker.com/r/ibmcom/helayers-pylab>
- [11] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size,” 2016.
- [12] M. Baruch, N. Drucker, L. Greenberg, and G. Moshkovich, “Fighting COVID-19 in the Dark: Methodology for Improved Inference Using Homomorphically Encrypted DNN,” 2021.