HE-MAN – Homomorphically Encrypted MAchine learning with oNnx models

Martin Nocker

ONNX

Trusted AI Committee Meeting
July 27th, 2023

Outline

- Motivation
- Homomorphic Encryption
- HE-MAN framework & ONNX

Machine Learning Applications

Spam Detection

Recommender Systems

Chatbots

Autonomous Cars

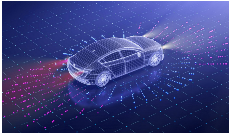


image source: rd.com/article/self-driving-cars

Machine Learning Applications

Sensitive Input

MA

Please comment and write the documentation for the following codeblock:

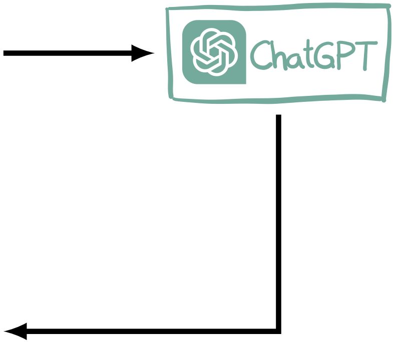
```
void super_secret_function() {
  ...
}
```

Certainly! Here's the code and its documentation:

Code:

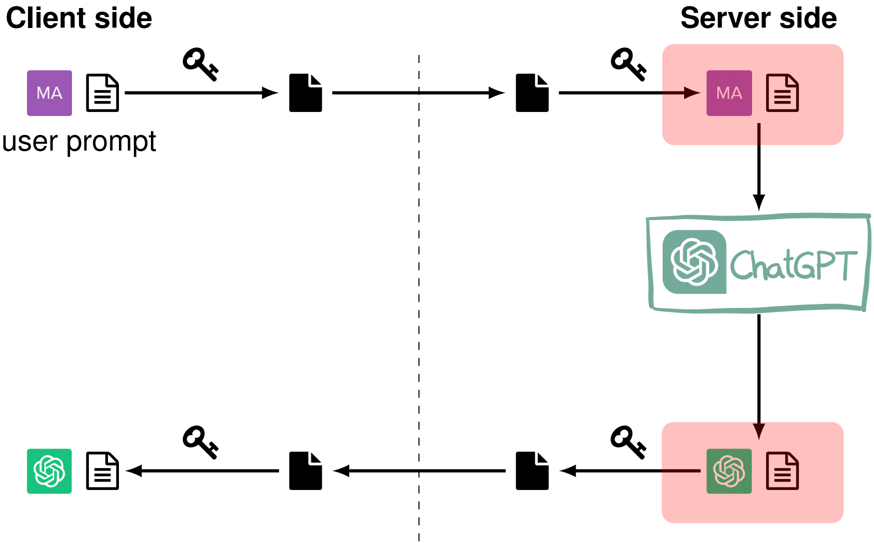
cpp Copy code

```cpp
// Function: super_secret_function
void super_secret_function() {
    // ... Implementation details intentionally omitted ...
}
```

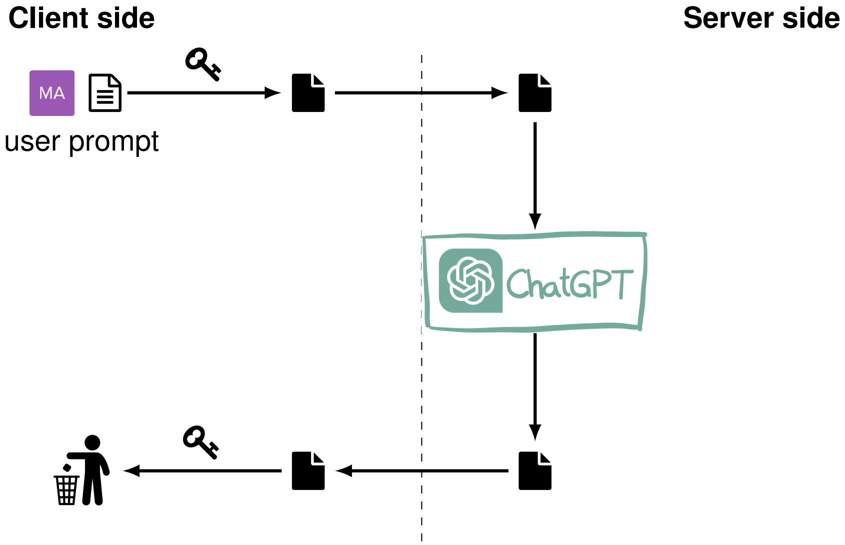


Machine Learning Applications

Classical cryptosystems

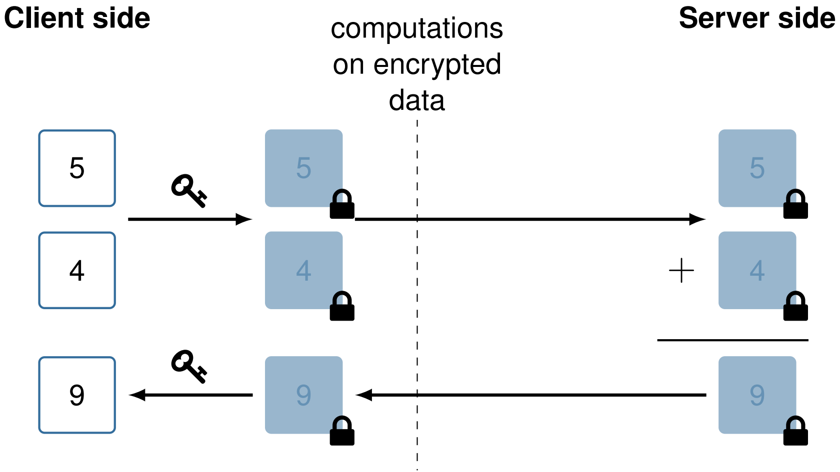


Machine Learning Applications
Classical cryptosystems



thenounproject.com/icon/garbage-2034076

Fully Homomorphic Encryption (FHE)



- $+$ <u>and</u> $\times$ $\Rightarrow$ **Fully** Homomorphic Encryption (FHE)

Fully Homomorphic Encryption (FHE)

Definition

Homomorphism: structure-preserving map

$f : A \to B$

$f(a + b) = f(a) \oplus f(b)$

Example

$f(x) = |x|$

$f(a \cdot b) = f(a) \cdot f(b)$

Example

RSA: $c = m^e \mod N$

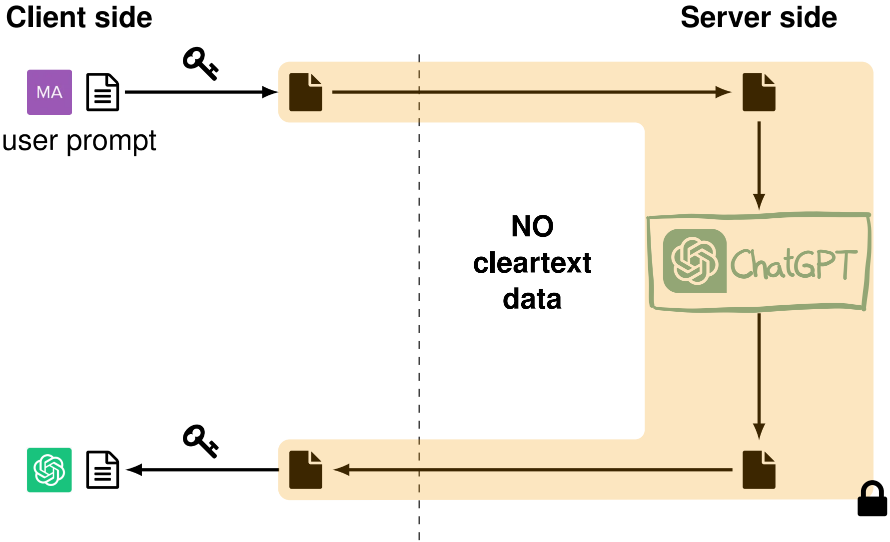
$\prod_i c_i = \prod_i m_i^e = (\prod_i m_i)^e \mod N$

Definition

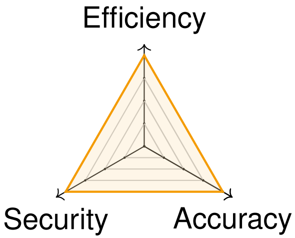
Fully Homomorphic Encryption (FHE) Scheme:

- $\mathbb{D}(\mathbb{E}(a) \oplus \mathbb{E}(b)) = a + b$
- $\mathbb{D}(\mathbb{E}(a) \otimes \mathbb{E}(b)) = a \times b$

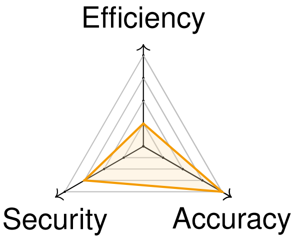
Machine Learning Applications
FHE



Fully Homomorphic Encryption (FHE)



Ideal World

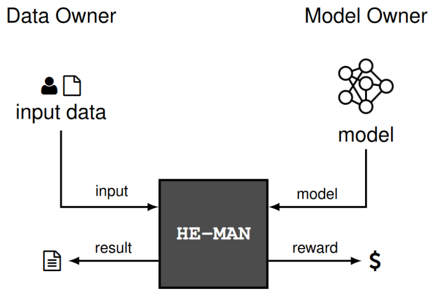


Real World

Further challenges:

- FHE operations are orders of magnitude more complex
- Only additions and multiplications of ciphertexts are possible

Privacy-Preserving ML



Design Goals

- Broad model support
- Abstraction of cryptographic details

Previous work

- NN inference for specific networks [BGBE19]
- Include other techniques, e.g. SMPC [HLHD22, LMSP21]
- Individual ML framework support: TensorFlow [RRK+20], PyTorch [KVH+21]

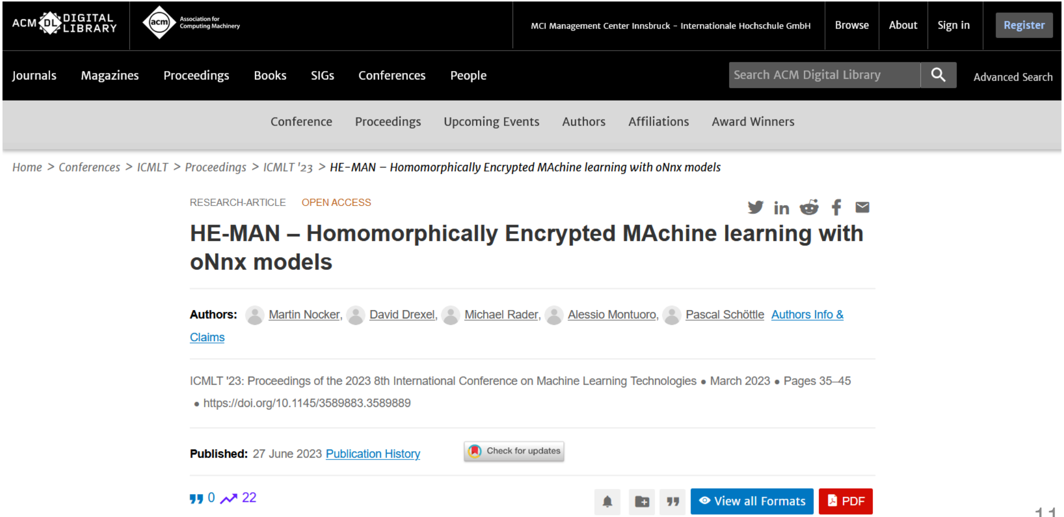
HE-MAN

- ONNX model input format
- FHE engineering
- Crypto details are abstracted away from the user

Results: accuracies close to cleartext numbers, at increased runtime

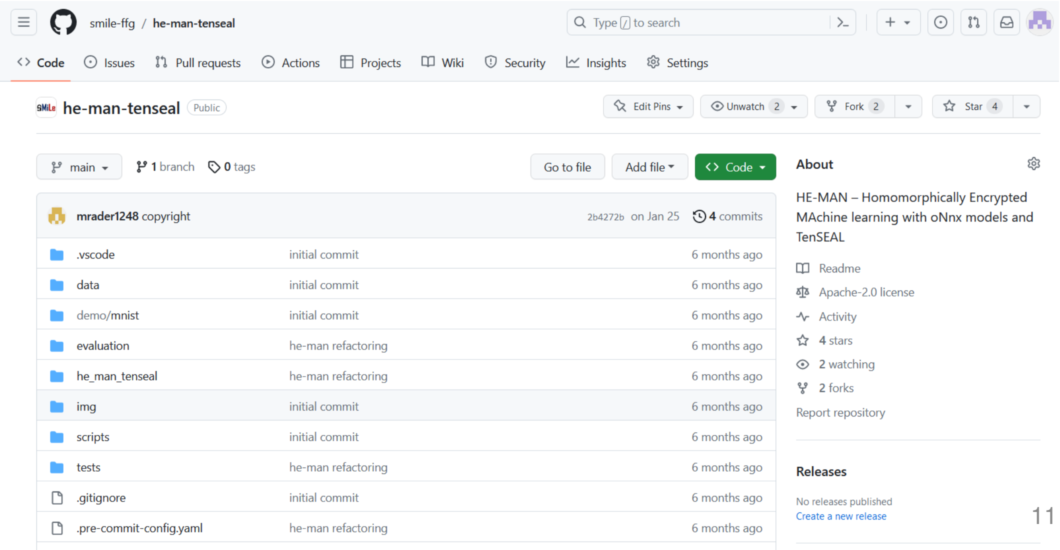
HE-MAN

https://dl.acm.org/doi/10.1145/3589883.3589889

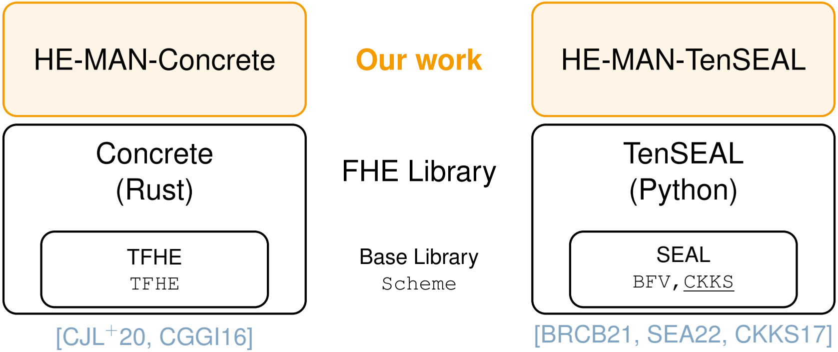


HE-MAN

https://github.com/smile-ffg/he-man-tenseal



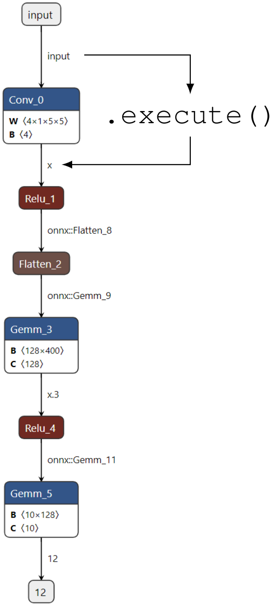
HE-MAN
Architecture



ONNX in HE-MAN

So far implemented

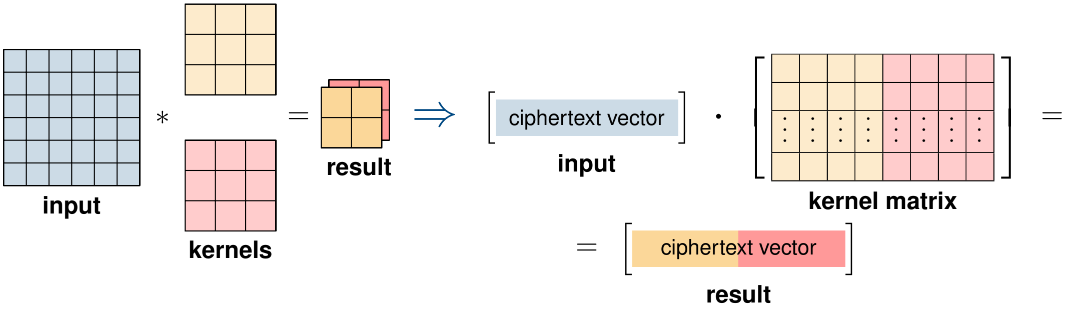
- AddOperator
- AveragePoolOperator
- ConstantOperator
- ConvOperator
- FlattenOperator
- GemmOperator
- MatMulOperator
- MulOperator
- PadOperator
- ReluOperator
- ReshapeOperator
- SubOperator



Linear operations in HE-MAN-TenSEAL

Convolution

- Ciphertext = vector of encrypted values
- Linear operations via vector-matrix multiplication



Other tools

README.md

ZAMA
Concrete ML

Read documentation | Community support

release v1.1.0 | Learn Tutorials and demos | Contribute Zama Bounty Program

---

Concrete ML is a Privacy-Preserving Machine Learning (PPML) open-source set of tools built on top of Concrete by Zama. It aims to simplify the use of fully homomorphic encryption (FHE) for data scientists to help them automatically turn machine learning models into their homomorphic equivalent. Concrete ML was designed with ease-of-use in mind, so that data scientists can use it without knowledge of cryptography. Notably, the Concrete ML model classes are similar to those in scikit-learn and it is also possible to convert PyTorch models to FHE.

Thank you!

https://github.com/smile-ffg/he-man-concrete

https://github.com/smile-ffg/he-man-tenseal

Paper:

https://dl.acm.org/doi/10.1145/3589883.3589889

References I

[BGBE19] Alon Brutzkus, Ran Gilad-Bachrach, and Oren Elisha.
Low latency privacy preserving inference.
In *International Conference on Machine Learning*, pages 812–821. PMLR, 2019.

[BRCB21] Ayoub Benaissa, Bilal Retiat, Bogdan Cebere, and Alaa Eddine Belfedhal.
TenSEAL: A library for encrypted tensor operations using homomorphic encryption, 2021.

[CGGI16] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène.
Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds.
Cryptology ePrint Archive, Paper 2016/870, 2016.
`https://eprint.iacr.org/2016/870`.

[CJL+20] Ilaria Chillotti, Marc Joye, Damien Ligier, Jean-Baptiste Orfila, and Samuel Tap.
CONCRETE: Concrete Operates oN Ciphertexts Rapidly by Extending TfhE.
In *WAHC 2020–8th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, volume 15, 2020.

[CKKS17] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song.
Homomorphic encryption for arithmetic of approximate numbers.
In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 409–437. Springer, 2017.

References II

[HLHD22] Zhicong Huang, Wenjie Lu, Cheng Hong, and Jiansheng Ding.
Cheetah: Lean and fast secure Two-Party deep neural network inference.
In *31st USENIX Security Symposium (USENIX Security 22)*, pages 809–826, Boston, MA, August 2022. USENIX Association.

[KVH+21] Brian Knott, Shobha Venkataraman, Awni Hannun, Shubho Sengupta, Mark Ibrahim, and Laurens van der Maaten.
Crypten: Secure multi-party computation meets machine learning.
In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 4961–4973. Curran Associates, Inc., 2021.

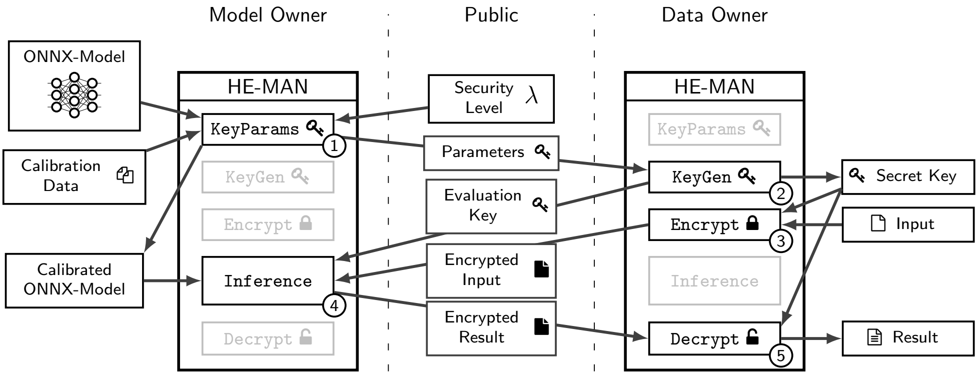
[LMSP21] Ryan Lehmkuhl, Pratyush Mishra, Akshayaram Srinivasan, and Raluca Ada Popa.
Muse: Secure inference resilient to malicious clients.
In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2201–2218. USENIX Association, August 2021.

References III

[RRK+20] Deevashwer Rathee, Mayank Rathee, Nishant Kumar, Nishanth Chandran, Divya Gupta, Aseem Rastogi, and Rahul Sharma.
Cryptflow2: Practical 2-party secure inference.
In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 325–342, New York, NY, USA, 2020. Association for Computing Machinery.

[SEA22] Microsoft SEAL (release 4.0).
`https://github.com/Microsoft/SEAL`, 2022.
Microsoft Research, Redmond, WA.

HE-MAN
Architecture



SMiLe

Secure Machine Learning Application with Homomorphically Encrypted Data