Privacy-preserving Data Federation for Trainable, Queryable and Actionable Data

Stavroula Iatropoulou, Theodora Anastasiou, Sophia Karagiorgou,
Petros Petrou, Dimitrios Alexandrou, Thanassis Bouras
Privacy-preserving Distributed Machine Learning Department
UBITECH LIMITED
Limassol, Cyprus
{siatropoulou, tanastasiou, skaragiorgou}@ubitech.eu
{ppetrou, dalexandrou, bouras}@ubitech.eu

***Abstract*—Privacy preservation over federated data has gained its momentum in the era of securing users' sensitive data. Combining and analysing sensitive data from multiple sources offers considerable potential for knowledge discovery. However, there are different constraints which should be fulfilled, such as what are the data to be preserved; what is meant by privacy preservation; what are the constraints on federated computing; and what are the secure mechanisms to train, query and explore data without accuracy loss. We introduce the Protected Federated Query Engine which applies Fully Homomorphic Encryption and Querying Mechanisms over decentralised data sources of diverse schemas and granularities to efficiently collect, align, aggregate and serve Artificial Intelligence Operations (AIOps) and Data Operations (DataOps) without sacrificing accuracy and efficiency.**

***Index Terms*—encrypted data management, privacy-preserving AI models lifecycle, privacy-preserving AI model training and serving over decentralised and heterogeneous data**

I. Introduction

Nowadays, many applications handling sensitive data, coming from the healthcare, insurance, mobility domains and more, require the management of the complete data lifecycle. Such data implies that privacy-preserving infrastructures need to be put in place, so that reliable and secure data operations can be provided.

In addition to this, contemporary applications combine and analyse sensitive data from multiple federated sources for better knowledge discovery and more accurate models' training. In some applications the enforcement of data protection regulations prohibit data centralisation for analysis purposes because of potential privacy risks, such as the accidental disclosure of data to third parties. Besides, there are a number of issues that pose problems for such analyses, including technical barriers, privacy restrictions, data protection compliance to one or more legal jurisdictions, security concerns, and trust issues.

Motivated by these pressing needs, this paper introduces a *Protected Federated Query Engine* with key differentiating factor that the encrypted data are then serving for collection, aggregation, and filtering via *DataOps* and Artificial Intelligence (AI) methods via *AIOps*. The aim is to optimize the complete data path, in terms of efficient, reliable, secure, and trustworthy data processing and federated analysis. We report early results via a promising prototype which enables to extract actionable insights from ubiquitous open breast cancer data [1] in a decentralized way, delivering efficient DataOps and intelligent AIOps, enforcing encryption mechanisms at the expected point of action.

The contributions of this paper are, as follows:

- The presentation of a reproducible, governance- and provenance-rich microservices prototype for protected data federation for trainable, queryable and actionable data which are preserving privacy across the complete data path;
- A mechanism for executing federated queries on top of encrypted data aiming at preventing access to sensible data from unauthorized users, applications and processes;
- A mechanism for Fully Homomorphic Encryption (FHE) which allows cloud servers, edge, and fog nodes to perform Data and Ops over encrypted data, while only authorised clients (i.e., users and applications) are able to see the decrypted data; and
- A *Protected Federated Query Engine* for decentralised query confidentiality which ensures that cross-domain knowledge cannot be generated and propagated between different nodes, resources, and administrative domains.

II. Related Work

Many circumstances necessarily force changes in privacy-preserving for supporting efficient DataOps and AIOps, such as the heterogeneous, decentralised and vast amounts of data to be analysed. Also, in the last years, there is an increased need for secure operations and analyses over the data eliminating their disclosure to third parties.

Therefore, this section briefly covers the most relevant and up-to-date research works, mostly related to homomorphic encryption and privacy-preserving distributed data operations. The literature review shows that either there are ambiguous definitions of privacy and confusion between privacy and security in the field [2], or secure multiparty computation (SMPC) is being performed where a cryptographic protocol calculates a result set, where the data of each participating party remains concealed, or Federated Processing / Learning is being performed in a parallel or incremental manner [3]. *The Protected Federated Query Engine is mostly related with the SMPC approach with the differentiating factor that the encrypted data are then federated in order to serve DataOps*

for collection, aggregation, querying and filtering and AIOps for AI and statistical model training and inference.

Clifton et al. [4] propose a toolkit of components that can be combined for specific privacy-preserving data mining applications and show how the latter can be used to solve several data mining and privacy preservation problems.

Sun et al. [2] provide a systematic review of privacy-preserving data mining techniques from 231 scientific articles published in the past 20 years. They present a summary of the state of the art, compare the problems they address, and identify the challenges in the field.

Welten et al. [3] support analyses on sensitive patient data by simultaneously complying with local data protection regulations using an approach called the Personal Health Train (PHT). The main principle of the PHT is that the analytical task is brought to the data provider and the data instances remain in their original location. The proposed engine differentiates by supporting secure multiparty homomorphic encryption for multiple parties to jointly compute some function over their own data without revealing the original data to any other parties. In addition, to protect the participants from being attacked by external parties (who are outside of the federated ecosystem), the proposed approach also protects the parties from each other.

A Homomorphic Encryption (HE) scheme allows the evaluation of arbitrary computations on encrypted data without decrypting it. In theory, realizing SMPC through a HE scheme is a simple and efficient approach. However, despite its promising theoretical power, the practical side of the approach remains underdeveloped [5]. In mathematics, homomorphism describes the transformation of one data set into another while preserving relationships between elements in both sets. In other words, an encryption algorithm E and a decryption D should satisfy the following conditions:

- c1 = E(a1), c2 = E(a2)
- D(f (c1, c2)) = f (a1, a2)

In the proposed approach, all the DataOps and AIOps (i.e., the aforementioned f) can be executed as-is in the encrypted data and the result analysis can be performed by the decryption key owner. HE can eliminate obstacles to privacy and enable data sharing.

To the best of our knowledge Goldwasser et al. [6], proposed and introduced the first semantically secure homomorphic encryption scheme by replacing Rivest–Shamir–Adleman's (RSA) trapdoor function [7] with a Probabilistic Encryption. The common types of homomorphic encryption are: Partially Homomorphic, Somewhat Homomorphic, Leveled Homomorphic Encryption and Fully Homomorphic Encryption (FHE). Since FHE has extremely high security, it allows users' data to be protected anytime it is sent to the cloud (e.g., from edge devices) and has made great contributions to privacy protection in cloud computing [8]. FHE can support multipliable operations (currently addition and multiplication), allowing more computation to be performed over encrypted data [9].

Many attempts have been made not only to propose new theoretical encryption algorithms but also be practical. The first FHE algorithm based on ideal lattices which satisfied both additive homomorphism and multiplicative homomorphism demonstrates the feasibility of computing any function on encrypted data. In homomorphic encryption, messages are encrypted with a noise that grows at each homomorphic evaluation of an elementary operation. The number of homomorphic operations is limited, but can be made asymptotically large using bootstrapping. This technical trick allows to evaluate arbitrary functions by essentially evaluating the decryption operation on encrypted secret keys.

Chechulina et al. [10], introduced a new fully homomorphic scheme that does not require massive computation resources and provides acceptable sizes of encryption keys and output values. The approach has an only constraint: the result of all the mathematical operations can not exceed the number of relatively prime numbers set.

Brakerski et al. [11], [12] introduced the concept of leveled homomorphic encryption to overcome the performance issues of bootstrapping. The parameters are chosen sufficiently large to evaluate the entire computation without bootstrapping. Additionally, they added support for Single Instruction, Multiple Data (SIMD)-style batching. This takes advantage of the fact that the plaintext space is a ring of polynomials with numerous coefficients, which may be interpreted as numerous distinct independent slots and allows for the compression of numerous messages (often 213-216) into a single ciphertext. Automorphisms additionally (i.e., HElib [13] implementation) enables homomorphically executable rotations between slots. The Gama-Georgieva-Izabachene (CGGI2) scheme [14], [15] performs bootstrapping in less than 100 milliseconds, while previous implementations/schemes needed several minutes even in efficient implementations. However, fast bootstrapping is incompatible with batching, introducing a trade-off between latency and throughput when compared to second-generation schemes. Al Badawi et al. [16] proposed and implemented (in C++) a new open-source FHE software library (OpenFHE) that incorporates selected design ideas from prior FHE projects, such as PALISADE [17], HElib, and HEAAN [18]. OpenFHE introduces several new design features: (i) the supported FHE schemes also support bootstrapping and scheme switching; (ii) the library can support multiple hardware acceleration backends using a standard Hardware Abstraction Layer (HAL); and (iii) OpenFHE includes both user-friendly modes, where maintenance operations (such as modulus switching, key switching, and bootstrapping) are automatically invoked by the library, and compiler-friendly modes where an external compiler makes these decisions.

The Protected Federated Query Engine protects data flows, queries and machine learning results via FHE through a federated, coordinated and programmable mechanism which has been implemented for securely executing queries and preserving the accuracy of the trained models and the inferred results. The federated mechanism has been implemented on top of encrypted data with the aim to preserve access to sensitive information from unauthorized users. The FHE enables cloud servers, edge, and fog nodes to compute arbitrary func-

tions (i.e., aggregations, filters, joins, etc.) over the encrypted data, while only authorized clients (i.e., users, devices, and applications) are able to view the decrypted data.

III. Protected Federated Query Engine

The core purpose of the Protected Federated Query Engine is to protect the data flows from cloud servers, edge and fog nodes with the ability to execute queries in a federated manner and preserve the privacy of the data and the accuracy of the trained models and the inferred results.

A. Conceptual Architecture

The Protected Federated Query Engine aims at protecting data flows by preserving the privacy and ensuring the information on the underlying data and analytic operations performed over the data without sacrificing efficiency of the DataOps and accuracy of the AIOps, respectively. It supports operations enabling federated queries execution on top of encrypted data taking care of the entire decentralised data lifecycle. It supports data sources which may have diverse schemas, granularity and types. The engine manages this heterogeneity via multiple connectors allowing to interact with different databases. The latter introduces a new perspective to the workloads management of online analytics and aggregation, serving as Online Analytical Processing (OLAP).

The Protected Federated Query Engine consists of a *Coordinator* and a customized number of *Worker* nodes which communicate with each other through a REST API. Each query statement concerning the encrypted data is submitted to the coordinator node, which, consequently, parses the statement and then, creates the query with a query plan, which is finally distributed for execution across a series of multiple workers.

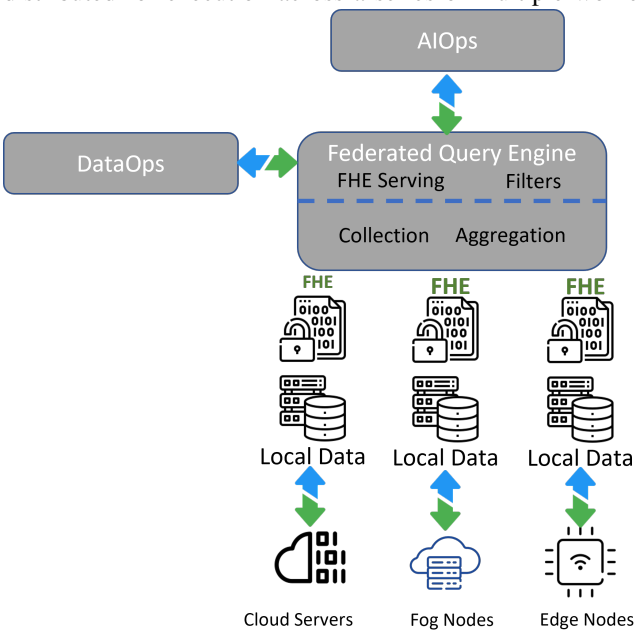

Fig. 1. Protected Federated Query Engine Architecture.

Secure and distributed multiparty computation is realised via a Fully Homomorphic Encryption method, which allows distributed entities to compute arbitrary functions over encrypted data, while only authorized clients (i.e., users and applications) are able to access, analyse and process the decrypted data. The high-level architecture of the proposed Protected Federated Query Engine is depicted in Figure 1.

B. Prototype and Experiments

The existing State-of-the-Art (SotA) FHE libraries facilitate users, with little or no expertise in the cryptography, to fully overcome the hindrances imposed by the complexity of implementing the homomorphic functions.

The open-source library used for the implementation of the FHE mechanism is Zama [19] and the framework for the entire data federation lifecycle is Trino [20]. It is used as the federated querying mechanisms on top of distributed, large and diverse data sources. Figure 2 depicts the experimental setup for measuring the AIOps efficiency of the proposed engine. The AIOps efficiency has been measured through the AI model's accuracy. We conducted experiments by measuring a conventional model's accuracy against FHE-enabled model's accuracy, both in the training and the inference phase. DataOps collect, aggregate and filter data from two different data storage systems in the support of the federated querying mechanisms.

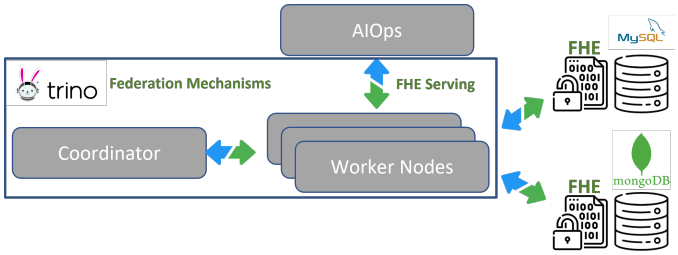

Fig. 2. Experimental Setup.

The supporting data federation mechanisms are written in Java, where a *Coordinator* orchestrates the workloads among a customised number of *Worker* nodes which communicate with each other through a REST API. In this setup, a Coordinator with two Workers have been deployed, in the support of MySQL and MongoDB data storage systems, respectively.

The dataset has been vertically split and stored in order to be queried in a federated manner. A join query is performed over the common 'id' field. The Workers fetch the encrypted data from each connector and exchange intermediate synchronisation data with each other. Finally, the Coordinator returns the results of the aggregated dataset and trains the FHE-enabled model.

We report early experimentation results by using the *Breast Cancer Dataset* [1] of the sklearn library. We selected 10 basic features which are, the: 'mean compactness', 'mean concave points', 'radius error', 'area error', 'worsttexture', 'worstperimeter', 'worstarea', 'worstsmoothness', 'worstconcavepoints' and 'worstsymmetry'. We then trained a Logistic Regression model and measured its accuracy via: (i) a conventional training method; and (ii) a federated FHE-enabled method. The federation mechanism is simulated by joining and aggregating the breast cancer data from a MySQL and

a MongoDB via Trino. DataOps have been then performed for data filtering and normalisation. The conventional training method used is the Logistic Regression from sklearn [21], while the FHE-enabled method used is the Concrete-ML of Zama [19]. Concrete-ML applies an encoding step quantising the data using a given number of bits. The higher the quantisation bit width, the better the precision, and also the more expensive the calculation.

In our experimentation, we used the default values of the Concrete-ML model with quantisation width equivalent to 2 bits. The FHE-enabled model's accuracy both during the training and the inference phase was identical with the conventional model's accuracy, and equal to 89%. Without compromising the privacy and sensitivity of the data, we equally achieve similar results. Figure 3 illustrates the comparative evaluation between the FHE-enabled model's accuracy and the conventional model's accuracy for different data sizes. We conclude that the proposed engine and its underlying mechanisms enable us to support data and AI operations over the encrypted dataset efficiently without sacrificing accuracy.

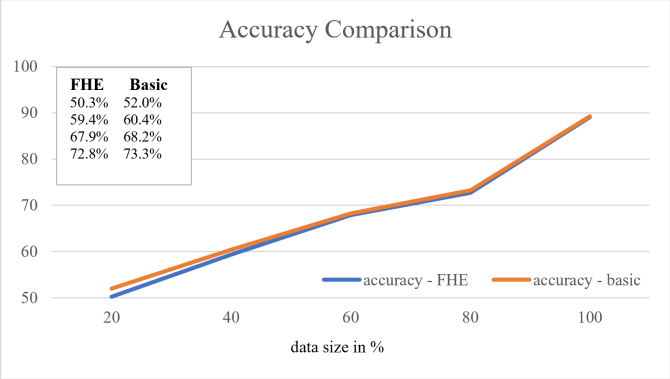

Fig. 3. FHE-enabled vs. Conventional Model's Accuracy Comparison.

IV. Conclusions and Future Work

The purpose of this paper is to present early results regarding the Protected Federated Query Engine which supports decentralised and secure multiparty computation. The engine applies FHE and querying mechanisms over distributed data of diverse schemas and granularities to efficiently collect, align, aggregate and serve AIOps and DataOps without sacrificing accuracy and efficiency, respectively.

In the near future, we plan to validate the solution in privacy-preserving individuals' mobility activities in indoor applications. We also plan to experiment over the computing performance of different FHE functions in order to scale up, parallelise and make more efficient the multiparty key generation.

V. Acknowledgments

This work has received funding by the European Commission project HEU MobiSpaces (https://mobispaces.eu/) under Grant Agreement No. 101070279.

References

[1] Sklearn breast cancer data set. [Online]. Available: https://t.ly/yCoq
[2] C. Sun, L. Ippel, A. Dekker, M. Dumontier, and J. Van Soest, "A systematic review on privacy-preserving distributed data mining," *Data Science*, no. Preprint, pp. 1–30, 2021.
[3] S. Welten, Y. Mou, L. Neumann, M. Jaberansary, Y. Y. Ucer, T. Kirsten, S. Decker, and O. Beyan, "A privacy-preserving distributed analytics platform for health care data," *Methods of Information in Medicine*, 2022.
[4] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, "Tools for privacy preserving distributed data mining," *ACM Sigkdd Explorations Newsletter*, vol. 4, no. 2, pp. 28–34, 2002.
[5] S. M. Ghanem and I. A. Moursy, "Secure multiparty computation via homomorphic encryption library," in *2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*, 2019, pp. 227–232.
[6] S. M. Shaft Goldwasser, "Probabilistic encryption how to play mental poker keeping secret all partial information," in *In Proceedings of the fourteenth annual ACM symposium on Theory of computing*. ACM, 1982, pp. 365–377.
[7] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.
[8] A. F. M. Ferhat Özgür Çatak", "Cpp-elm: cryptographically privacy-preserving extreme learning machine for cloud systems," *International Journal of Computational Intelligence Systems*, vol. 11, no. 1, pp. 33–44, 2018.
[9] R. K. Mark A. Will, *A guide to homomorphic encryption*, ser. The Cloud Security Ecosystem, 2015.
[10] S. K. Darya Chechulina, Kirill Shatilov, "Fully homomorphic encryption for secure computations in protected database." in *In FedCSIS (Position Papers)*. FedCSIS, 2015, pp. 125–131.
[11] C. G. Zvika Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory*, vol. 6, no. 13, pp. 1–36, 2014.
[12] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical gapsvp." in *In: Safavi-Naini, R., Canetti, R. (eds) Advances in Cryptology – CRYPTO 2012*. Springer, Berlin, Heidelberg, 2012, p. 868–886.
[13] S. Halevi and V. Shoup. Helib. [Online]. Available: https://github.com/homenc/HElib
[14] M. G. M. I. Ilaria Chillotti, Nicolas Gama, "Faster packed homomorphic operations and efficient circuit bootstrapping for tfhe," in *In: Takagi, T., Peyrin, T. (eds) Advances in Cryptology – ASIACRYPT 2017*. Springer, 2017, p. 377–408.
[15] I. Chillotti, N. Gama, M. Georgieva, and M. 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. [Online]. Available: https://eprint.iacr.org/2016/870
[16] A. A. Badawi, J. Bates, F. Bergamaschi, D. B. Cousins, S. Erabelli, N. Genise, S. Halevi, H. Hunt, A. Kim, Y. Lee, Z. Liu, D. Micciancio, I. Quah, Y. Polyakov, S. R.V., K. Rohloff, J. Saylor, D. Suponitsky, M. Triplett, V. Vaikuntanathan, and V. Zucca, "Openfhe: Open-source fully homomorphic encryption library," Cryptology ePrint Archive, Paper 2022/915, 2022, https://eprint.iacr.org/2022/915. [Online]. Available: https://eprint.iacr.org/2022/915
[17] G. W. R. Y. Polyakov, R. Rohloff and D. Cousins, "Palisade lattice cryptography library." [Online]. Available: https://palisade-crypto.org/
[18] Heaan. [Online]. Available: https://github.com/snucrypto/HEAAN
[19] Concrete ml. [Online]. Available: https://www.zama.ai/concrete-ml
[20] Trino: Distributed sql query engine for big data. [Online]. Available: https://trino.io/
[21] scikit-learn: machine learning in python. [Online]. Available: https://scikit-learn.org/