CL-002

Jeddak：データビジネスにおけるプライバシー強化技術の活用
Jeddak: The Application of Privacy Enhanced Technology in Practical Data Business

洪　爵† 張 祺智 † 呉　燁† 韓 東力‡
Jue Hong Qizhi Zhang Ye Wu Dongli Han

1. Introduction

In modern monetization and business, data circulation and sharing are key to improving efficiency and revenue. However, due to the arising data privacy concerns and data protection regulations like GDPR, the direct sharing of data, especially that containing Personally Identifiable Information (PII) and commercial confidentiality, is risky and is regarded as unacceptable.

As a promising solution, Privacy Enhanced Technologies (PETs) are widely used to protect user privacy and meet legal compliance requirements. PETs enable users to share and compute data invisibly (e.g., in ciphertext or within unbreakable sandboxes), obtaining only the final results without exposure to any intermediate or input data. However, the high computational and communication overhead makes PETs impractical in real applications, especially when dealing with data volumes in the billions. Furthermore, developing privacy-preserving applications using PETs requires expertise in cryptography or trusted hardware, which makes it challenging for most enterprises.

Recent work on PETs mainly focuses on designing efficient algorithms or protocols, with few discussing the guidelines for designing an easy-to-use PET system or the application patterns for PETs. In this paper, we present the Jeddak system, which provides capabilities for privacy-preserving data processing and sharing using comprehensive PETs. Jeddak is designed as a unified system that combines various PETs and can be integrated with existing data and AI infrastructures in modern companies. Notably, by designing SQL-oriented protocols and introducing asynchronous learning mechanisms, Jeddak can process data on a billion-scale.

Out contributions are summarized as follow:

1) We present Jeddak, a practical privacy-preserving data-circulation system. Its design can be regarded as guidelines for developing a practical PET-enabled application.
2) We show how Jeddak integrates with data and AI platforms in modern companies, providing smooth usability for legacy business systems.
3) We then describe the algorithmic designs of Secure Multi-party Computation and Federated Learning, which enable Jeddak to handle billion-scale data tasks.
4) We present a series of real-world application use cases of PETs, demonstrating how PETs can assist real businesses.

In the rest of this paper, we first briefly describe several typical privacy enhanced technologies in Section 2, then we present the design and architecture of our Jeddak platform in Section 3. In Section 4 and 5 we dive deeper into the details of how Jeddak is integrated with existing infrastructures, and how it handles the challenges of billion-scale data. In Section 6 we give application examples of Jeddak. We conclude this paper in Section 7.

2. Overview of Typical PETs

In this section, we introduce several key Privacy Enhanced Technologies (PETs), namely Secure Multi-party Computation (MPC), Homomorphic Encryption (HE), Differential Privacy (DP), Federated Learning (FL), and Trusted Execution Environment (TEE).

Secure Multi-Party Computation [1]-[6] enables a group of mutually distrustful participants to collaboratively compute an agreed-upon function without revealing their individual input data, while ensuring that the entire process does not leak any input or intermediate data. MPC has rigorous mathematical proofs and a comprehensive theoretical framework, making it suitable for applications in fields with high security requirements, such as finance, government, and healthcare. It can address the collaborative computation issues of sensitive data, such as 'multi-source integration.' Additionally, MPC offers good computational efficiency and has a wide range of applications in the field of classical computing, such as Private Set Intersection (PSI), Private Information Retrieval (PIR), and joint statistics.

Homomorphic Encryption [7]-[12] allows executing arithmetic operations such as addition and multiplication directly on encrypted data without the need for decryption. This capability can be extended to perform a variety of computations. It is suitable for conducting data analysis and complex reasoning on encrypted data, providing post-quantum security. It is considered a foundational technology for secure privacy computing under " zero trust" conditions and is hailed as a cornerstone for the security of data in future cloud computing.

Differential Privacy [13]-[15] provides a rigorous and provable means of privacy protection, ensuring the privacy-compliant collection, querying, and publishing of sensitive data. Its strength lies in the independence of privacy protection from an attacker's background knowledge. Centralized differential privacy techniques involve a trusted data owner adding noise to statistical results, ensuring that the modification of a single record does not significantly affect the outcome, thus meeting privacy protection requirements. This addresses user privacy leakage during interactive queries and dataset publication. Local differential privacy techniques add noise to data before collection, preventing privacy leakage at the data collection stage.

Federated Learning [16]-[18] is a method where different institutions or entities train models on their local data without sharing it outside their domain. During the iterative process, they exchange intermediate parameters such as gradients and residuals, leveraging privacy-preserving technologies like homomorphic encryption, ultimately achieving a modeling effect equivalent to centralized data training. This approach solves the 'data silo' problem and enables efficient AI training and inference under privacy protection. It is widely applied in fields such as joint

† ByteDance
‡ Central University of Finance and Economics

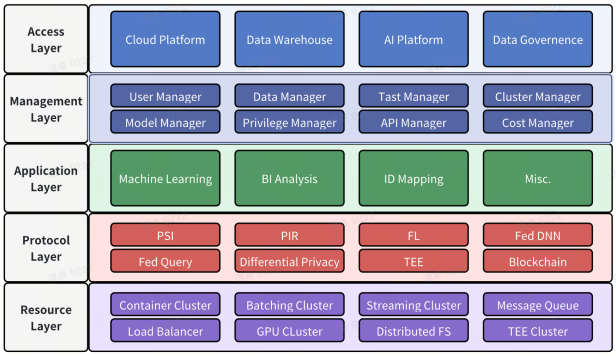

Figure 2

financial risk control and joint marketing in advertising. Its characteristic is that it is highly suitable for building complex models with large datasets.

A Trusted Execution Environment [19] is a secure and isolated environment based on hardware. One example is Intel SGX, which uses Intel CPUs to create an isolated and encrypted secure area in memory for applications. The TEE provides hardware-level confidentiality and integrity protection for the code and data within, preventing eavesdropping or tampering. It ensures the trustworthiness of the execution environment and code through remote attestation services.

3. Design of Jeddak

Based on modern PETs, we have developed Jeddak, a secure and privacy-preserving data-circulation platform. In the following sections, we initially present the architecture and system design of Jeddak. Subsequently, we summarize the application patterns of PETs in real-world business scenarios, which serve as the foundational design principles for Jeddak.

3.1 Architecture

The architecture of Jeddak is structured into five distinct layers, each collaborating to offer a comprehensive, secure data-sharing service, as shown in Figure 2:

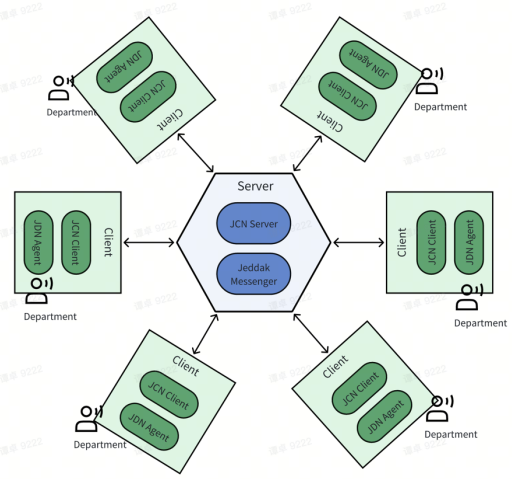

Figure 1

- **Access Layer**: It offers a range of access interfaces to business systems, designed to preserve user conventions and reduce the costs associated with integration and adaptation.
- **Management Layer**: This layer provides a suite of tools for platform management, encompassing user, data, task, and privilege management.
- **Application Layer**: It comprises application-specific protocols focused on privacy-preserving machine learning, business intelligence analysis, and more.
- **Protocol Layer**: It encompasses the fundamental libraries of privacy-preserving algorithms, including PSI, PIR, FL, among others.
- **Resource Layer**: This layer delivers industrial-grade computation, communication, and storage resources to the entire platform, naturally inheriting benefits such as large-scale capabilities, high availability, and elastic scalability.

3.2 System Design

The Jeddak platform consists of the following five main components as depicted in Figure 1:

- **JCN (Jeddak Control Network) Server**: It manages global, non-sensitive user information such as user IDs and task configurations, serving as a central repository for metadata without compromising user privacy.
- **Jeddak Messenger**: It acts as a communication middleware for the execution of privacy computing tasks, ensuring secure and isolated channels for communication. It supports unicast, multicast, and broadcast to facilitate various communication patterns.
- **JCN Client**: This component manages local, sensitive user information, including metadata and model artifacts.
- **JDN (Jeddak Data Network) Agent**: It is responsible for the execution of specific privacy computing algorithms. Scheduled as tasks by the JCN Server, these algorithms are executed in collaboration with a peer JDN Agent from other departments.

These five components work in concert to create a secure and efficient environment for privacy computing, ensuring that sensitive data is processed without information leakage.

3.3 Application Patterns

PETs offer innovative capabilities focused on data applications, including data security, privacy protection, and asset management assistance. Based on extensive deployment in real-world business scenarios, we have identified three key application forms of PETs that inform the design of Jeddak:

- **Integration with Data Infrastructure**: Starting from the source of data usage, privacy computing capabilities are integrated into the data management platform, fully managing the data usage process throughout the entire lifecycle of business applications.
- **Integration with Data Products**: Privacy computing functions are integrated into commonly used data suites, ensuring easy user access. For instance, in scenarios involving data circulation and sharing, when sensitive data is processed with data products, privacy-enhancing functions such as order-preserving encryption and homomorphic encryption can be readily activated to maintain data privacy.

- **Integration with Data Development Tools**: This approach assists users in developing and expanding domain-specific applications. For example, as demonstrated later, Jeddak's federated learning has been integrated into our company's AI platform, enabling businesses to conveniently incorporate federated learning capabilities and construct their own AI privacy computing applications.

4. Integration with Infrastructure

In this section we demonstrate how Jeddak being integrated with practical business infrastructure, e.g., the data platform and the AI platform.

4.1 Integrate with Data Platform

A modern company's data platform typically consists of five major systems: data collection, processing, storage, governance, and operations. Figure 3 illustrates how Jeddak, a data circulation system based on PETs, interacts with the existing data platform to enhance the security and privacy of data circulation.

- **Data Governance**: The governance system oversees data catalogs, relationships, permissions, and model management. By interacting with this system, Jeddak can identify data involved in circulation, obtain classification and permissions, and register newly produced data back into the governance framework.
- **Data Storage**: The storage system comprises various facilities such as object storage, databases, and data warehouses. Jeddak loads data from the heterogeneous storage, leverages them in the privacy computing process, and persists results for future use in downstream business scenarios.
- **Data Operation**: Traditional data operation systems manage data exposure in an insecure manner, failing to maintain data ownership integrity. Jeddak supplants these systems by providing secure data application interfaces that prioritize data circulation security.

This integration exemplifies how Jeddak can seamlessly integrate into the existing data platform, augmenting security and efficiency through its privacy computing features.

4.2 Integrate with AI Platform

Federated learning, despite its frequent use, encounters several challenges in practical applications, primarily due to existing platforms being standalone systems that do not integrate well with a company's AI infrastructure. This situation leads to two main issues:

- **High Migration Costs**: Current modeling tasks operate on various AI platforms within a company, tightly integrated in terms of data access, model development, distributed training scheduling, and online inference. Adopting an independent federated learning platform would require these tasks to be redeveloped and adapted to the new platform, leading to prohibitive migration costs.
- **Non-reusability of Existing Models**: Businesses have often trained numerous models using traditional AI platforms. Migration to a federated learning platform renders these models unusable.

To bridge the gap between federated learning and existing AI platforms, we have proposed and implemented the Native-FL (Native Federated Learning) schema within a large internet company. The Native-FL schema primarily involves adapting the model scripting language and the model training process.

The Native-FL schema provides APIs that align with the existing AI platform's format and definition, covering both federated training operations and private data preprocessing. This allows developers to either write new federated models using

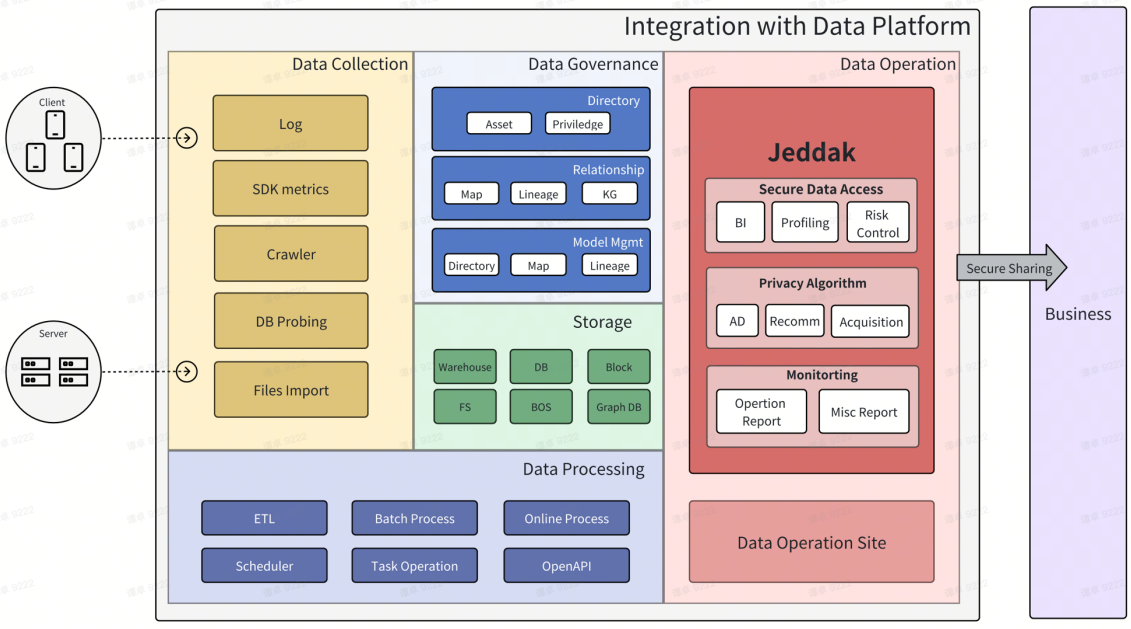

Figure 3

familiar APIs or convert existing models into a federated version using the provided private learning APIs.

Once a federated model is composed using AI platform-compatible APIs, the training process can be scheduled and executed by the platform without modification. For instance, the AI platform can retrieve data with its original data-source component, supply it to the federated model on each participant, initiate training tasks across multiple workers, and aggregate and update model weights on parameter server clusters. It's important to note that the Native-FL coordinator assists in aligning data segments and pairing tasks between worker clusters in each party, ensuring the accuracy of the training process.

5. Handling Billion-Scale Data

In real-world application scenarios at modern internet companies, data size frequently exceeds the billion-row mark. For instance, a daily influx of 500 billion rows, amounting to 100 terabytes of data, is processed for cross-departmental AI modeling and business intelligence analysis, posing a significant challenge to Jeddak. The challenge is twofold: reducing the substantial computational costs associated with PETs and ensuring the successful completion of long-duration PET tasks, which can range from several to tens of hours. Below, we demonstrate how Jeddak tackles these challenges.

5.1 Improving Performance

Here, we briefly describe how Jeddak implements high-performance MPC-based collaborative data analysis and a scalable FL system.

To construct a high-performance FL system, we have implemented the federated algorithm using the parameter-server (PS) model. By leveraging the PS model's scalability, we can divide large datasets into smaller chunks and distribute them across worker clusters to train a shared model in parallel. Building on the PS model, we propose an asynchronous federated split-training algorithm that breaks the traditional algorithm's synchronous interaction barriers, achieving faster training speeds (as shown in Figure 5).

For MPC-based collaborative analysis, we enhance performance by reducing the MPC computation domain and offering a suite of SQL-oriented protocols. Ensuring semantic security, we execute push-down and pull-up operations on the

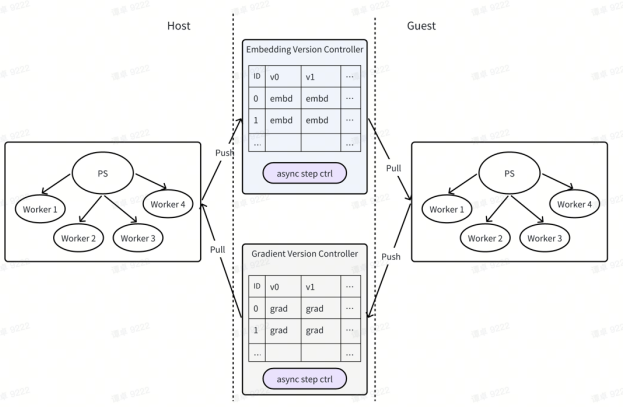

Figure 5

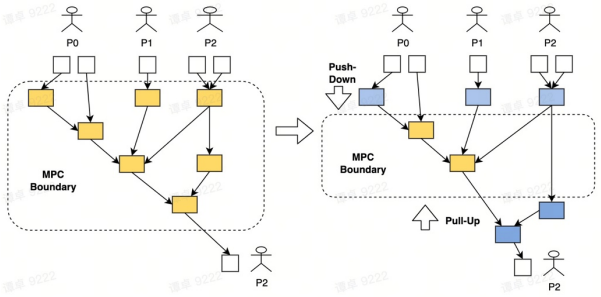

Figure 4

boundaries of MPC computations (Figure 4). Specifically, we decompose and identify computations involving only local data, moving them into the plaintext domain for local computation (push-down). Computations within the MPC that can be inferred from query results are directly converted to plaintext computations, effectively raising the lower boundary of the MPC (pull-up). By combining push-down and pull-up strategies, the computational load of MPC can be significantly reduced.

In the design of SQL-oriented protocols, we utilize an efficient secret-sharing-based Private Set Intersection (PSI) to replace the expensive ciphertext join operations. Subsequently, we employ the one-hot vector data structure, in conjunction with vector multiplication, to achieve efficient grouping and filtering within the MPC framework, thereby avoiding the costly operations of sorting and comparison in the ciphertext domain. By integrating these solutions, aggregations such as sum or count with grouping can be executed within minutes, even at the scale of millions of data entries.

5.2 Achieving High Availability

Maintaining high availability for PET computation systems is crucial, particularly when dealing with billion-scale data that can require tasks to run for several hours to tens of hours. It is clear that without fault-tolerant mechanisms or failure recovery, even a transient network error could lead to the complete failure of a task, resulting in the waste of time and computing resources.

To ensure high availability in Jeddak, we utilize methodologies from both microservices and cloud-native distributed architectures, as detailed below:

- **Checkpoint**: For long-running privacy computing tasks like AI modeling, we implement distributed checkpoints to save the intermediate state and data. This allows a failed task to resume from the checkpoint instead of starting from scratch.
- **Global Consistency**: The central server of Jeddak (e.g., JCN Server) maintains a global state machine for each privacy computing task and monitors the health status of all participating cluster nodes via heartbeats. For example, if a node fails, the global state transitions from 'RUNNING' back to 'READY', prompting all involved nodes to await the rescheduling and re-execution of the failed task.
- **Failure Recovery**: A failed task is eligible for rescheduling and re-execution for a preset number of retry attempts. During a retry, all involved nodes load the checkpoint from the shared file system and continue from the breakpoint, thereby reducing execution time.

These fault tolerance mechanisms ensure that Jeddak can handle failures gracefully and maintain efficient operations, offering a robust environment for privacy computing tasks.

6. Applications

In this section, we detail several typical applications of Jeddak in real-world businesses, demonstrating how PETs address data privacy and compliance issues, and enhance the utilization of data within businesses.

6.1 Collaborative Query

Consider a collaboration involving multiple institutions, each holding its own databases. Joint queries across these institutions could yield more valuable insights. However, due to privacy concerns, direct data sharing between institutions is not permitted. Meeting privacy compliance requirements while interconnecting these scattered databases has become a challenge for many businesses.

Jeddak has implemented a joint query system grounded in MPC to address this challenge. The core concept relies on the use of provably secure MPC protocols to execute joint query tasks without compromising privacy. The features of Jeddak's joint query system include:

- Secure Foundation: It is built on MPC, offering precise and provably secure capabilities for joint queries.
- Seamless Integration: The system mirrors traditional database query operations, significantly minimizing the learning curve for users.
- Performance Optimization: The MPC-friendly execution plans and fast MPC protocols satisfy real-world business performance requirements.

6.2 Private Set Intersection

Private Set Intersection (PSI) protocols allow two or more set holders to compute the intersection of their data sets without leaking any additional information. As a classic multi-party secure computation protocol, it has been widely applied in scenarios such as mobile contact list intersection and joint profiling of users across multiple parties.

Jeddak provides a fast PSI implementation based on the Oblivious Pseudo-Random Functions (OPRF). It ensures that the original user data does not leave the domain, and the protocol's recipient only obtains the set of elements within the intersection. Moreover, the protocol achieves an effective trade-off between communication and computational complexity. Jeddak's PSI has been applied in various fields such as finance and healthcare.

To further enhance the performance of PSI, Jeddak employs methods such as data sharding and parallel processing. It also caches intermediate pre-computed data, which significantly reduces execution time. As a result, the process of finding intersections for millions of data entries takes only minutes.

6.3 Private Data Collection

Data collectors, such as internet companies, banks, and other service providers, can analyze and mine user characteristics by collecting personal data, thereby enhancing the application experience for users and generating more revenue opportunities. However, the collection of personal data poses the risk of serious privacy leaks and legal implications. For instance, user trajectory data could reveal sensitive information like home addresses and hospital visit records.

Jeddak offers a private data collection tool that employs local differential privacy, known as the Locally Differentially Private Data Collector (LDPDC). It features:

- **T**heoretical Foundation: It is grounded in the theory of local differential privacy, offering robust and verifiable privacy protection.
- Customized Solutions: It provides tailored data collection strategies for various types of personal data, ensuring the efficiency of the data collected.
- Minimal Overhead: It minimizes interactions between the server and the client, resulting in low communication overhead.
- User-Centric Privacy: It includes an interface that allows users to adjust the level of privacy protection according to their individual preferences.

These features make LDPDC a powerful tool for collecting data while maintaining stringent privacy standards.

6.4 Private Database Statistic

It is well known that querying statistical information from a database, such as sums or averages, can potentially lead to serious personal privacy leakage. For instance, in a salary database scenario, a data analyst might use a statistical query to determine that the total salary expenditure for Department A is 3 million. If a new employee, let's call them Employee A, joins Department A, and the analyst then recalculates the total salary expenditure to be 3.75 million, it would be trivial to infer that Employee A's salary is 750,000.

Jeddak has implemented a private query processor utilizing differential privacy, known as Differentially Private SQL (DPSQL), to safeguard against privacy leakage in aggregate queries. DPSQL offers the following features:

- **P**rovable Privacy Protection: It is founded on centralized differential privacy theory, offering stringent and provable privacy protection.
- Low Learning Curve: It requires minimal learning effort, accepting SQL statements as input and producing query results that adhere to differential privacy, in line with standard database usage.
- Wide Compatibility: It is compatible with a range of SQL/NoSQL/analytical databases, including but not limited to Spark, ClickHouse, MySQL, and others.

6.5 Collaborative Marketing Using FL

Collaboration between institutions can typically yield more comprehensive user profiles, facilitating precise ad targeting and thereby improving revenue. Recognizing that data is a critical asset for these institutions and considering compliance requirements, privacy-preserving computation methods must be employed when conducting joint data alignment and model training.

Jeddak provides a federated learning service for a major e-commerce company. With data from both parties remaining within their respective databases, it utilizes algorithms such as xDeepFM and DCN, leveraging distributed clusters to process tens of millions of sparse user features. It completes the iterative update of a model with millions of users and billions of parameters within just 2 hours, achieving an approximate 17% improvement in AUC. This efficiency primarily originates from its innovative encryption algorithms and distributed model optimization techniques.

Jeddak has assisted the e-commerce company in addressing the challenge of training complex models for a vast user base, increasing their conversion rate by over 10%. It has satisfied the requirement of completing model training without the original data ever leaving the database. Furthermore, its collaborative computing approach, federated learning, has also resolved associated data security and privacy concerns.

6.6 Collaborative Risk Control Using FL

Risk models are a crucial risk management strategy for banks when extending loans to businesses. The development of risk models depends on a comprehensive understanding of data related to the enterprises and their controllers, which includes central bank credit reports, tax information, public opinion, trade information with upstream and downstream partners, and financial assets. Typically, banks have access only to central bank credit reports, limiting their ability to fully assess enterprise risk and impacting the non-performing loan ratio.

Jeddak employs federated learning to effectively address this issue. It ensures that banks can perform risk control modeling and reduce the non-performing loan ratio without the need for data to leave their databases. Additionally, it offers a legitimate avenue for data providers to effectively leverage the value of their data.

Equipped with a suite of feature engineering and federated learning algorithms commonly used in risk control, such as federated general linear models, federated decision trees, and federated Xgboost, Jeddak aids banks with corporate legal person credit data and enterprise invoice information in joint modeling efforts. While safeguarding the privacy of both parties' data, the Area Under the ROC Curve (AUC) has improved from 0.63 to 0.76, significantly enhancing the banks' risk control capabilities.

7. Conclusion

In this paper, we introduce the Jeddak system, a prime example of leveraging Privacy Enhanced Technologies to facilitate secure and private data processing and circulation. We detail the design and architecture of Jeddak, highlighting its seamless integration with existing business workflows. This integration is exemplified through two case studies: one with a data platform and another with an AI platform.

We then discuss how Jeddak addresses the performance challenges posed by billion-scale data from real-world applications. It does so by employing fast MPC protocols and asynchronous federated learning algorithms. Furthermore, we illustrate how Jeddak achieves high availability in long-running privacy computation tasks.

Finally, we present a series of application examples that demonstrate the capabilities of PETs in enabling data sharing between businesses while ensuring privacy and regulatory compliance.

参考文献

[1] Ran Canetti, et al., "Adaptively Secure Multi-party", TOC/CIS groups, LCS, MIT (1996), p. 1.

[2] A. Shamir, R. Rivest, and L. Adleman, "Mental Poker", Technical Report LCS/TR-125, Massachusetts Institute of Technology, April 1979.

[3] Andrew C. Yao, Protocols for secure computations (extended abstract)

[4] Andrew Chi-Chih Yao:How to Generate and Exchange Secrets (Extended Abstract). FOCS 1986: 162-167

[5] Oded Goldreich, Silvio Micali, Avi Wigderson:How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. STOC 1987: 218-229

[6] Zvi Galil, Stuart Haber, Moti Yung: Cryptographic Computation: Secure Fault-Tolerant Protocols and the Public-Key Model. CRYPTO 1987: 135-155

[7] Sellers, Andrew. "Council Post: Everything You Wanted To Know About Homomorphic Encryption (But Were Afraid To Ask)". Forbes. Retrieved 2023-08-18.

[8] Munjal, Kundan; Bhatia, Rekha (2022). "A systematic review of homomorphic encryption and its contributions in healthcare industry". Complex & Intelligent Systems. 9 (4): 3759–3786. doi:10.1007/s40747-022-00756-z. PMC 9062639. PMID 35531323.

[9] Armknecht, Frederik; Boyd, Colin; Gjøsteen, Kristian; Jäschke, Angela; Reuter, Christian; Strand, Martin (2015). "A Guide to Fully Homomorphic Encryption". Cryptology ePrint Archive.

[10] Vinod Vaikuntanathan. "Homomorphic Encryption References".

[11] R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. In Foundations of Secure Computation, 1978.

[12] Sander, Tomas; Young, Adam L.; Yung, Moti (1999). "Non-interactive cryptocomputing for NC/Sup 1/". 40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039). pp. 554–566.

[13] Hilton, M; Cal (2012). "Differential Privacy: A Historical Survey". Semantic Scholar. S2CID 16861132. Retrieved 31 December 2023.

[14] Dwork, Cynthia (2008-04-25). "Differential Privacy: A Survey of Results". In Agrawal, Manindra; Du, Dingzhu; Duan, Zhenhua; Li, Angsheng (eds.). Theory and Applications of Models of Computation. Lecture Notes in Computer Science. Vol. 4978. Springer Berlin Heidelberg. pp. 1–19. doi:10.1007/978-3-540-79228-4_1. ISBN 978-3-540-79227-7. S2CID 2887752.

[15] Kairouz, Peter; McMahan, H. Brendan; Avent, Brendan; Bellet, Aurélien; Bennis, Mehdi; Bhagoji, Arjun Nitin; Bonawitz, Kallista; Charles, Zachary; Cormode, Graham; Cummings, Rachel; D'Oliveira, Rafael G. L.; Eichner, Hubert; Rouayheb, Salim El; Evans, David; Gardner, Josh "Advances and Open Problems in Federated Learning". Foundations and Trends in Machine Learning. 14 (1–2): 1–210.

[16] Pokhrel, Shiva Raj; Choi, Jinho (2020). "Federated Learning with Blockchain for Autonomous Vehicles: Analysis and Design Challenges". IEEE Transactions on Communications. 68 (8): 4734–4746. doi:10.1109/TCOMM.2020.2990686. S2CID 219006840.

[17] Xu, Zirui; Yu, Fuxun; Xiong, Jinjun; Chen, Xiang. "Helios: Heterogeneity-Aware Federated Learning with Dynamically Balanced Collaboration". 2021 58th ACM/IEEE Design Automation Conference (DAC). pp. 997–1002.

[18] Yu, Fuxun; Zhang, Weishan; Qin, Zhuwei; Xu, Zirui; Wang, Di; Liu, Chenchen; Tian, Zhi; Chen, Xiang (2021-08-14). "Fed2". Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. KDD '21. New York, NY, USA: Association for Computing Machinery. pp. 2066–2074.

[19] Sabt, M; Achemlal, M; Bouabdallah, A (2015). "Trusted Execution Environment: What It is, and What It is Not". 2015 IEEE Trustcom/BigDataSE/ISPA (PDF). IEEE. pp. 57–64. doi:10.1109/Trustcom.2015.357. ISBN 978-1-4673-7952-6.