IJARETY

International Journal of Advanced Research in Education and TechnologY (IJARETY)

Volume 12, Issue 2, March-April 2025

Impact Factor: 8.152

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202046

SmartCrypt: Enhancing Security for Storing and Sharing Time-Series Data in IIoT

Sindhu Priyanka Chadalavada[1], Narra Pujitha[2], Golla Akhila[3], Bale Syam Sundar[4], Achana Bhanu Prakash[5]

Associate Professor, Department of CSE, Eluru College of Engineering & Technology, Eluru, India[1]

B. Tech Student, Department of CSE, Eluru College of Engineering & Technology, Eluru, India[2,3,4,5]

ABSTRACT: To ensure widespread access, scalability, and sharing capabilities, applications of the Industrial Internet of Things (IIoT) leverage the cloud for storing the data streams they collect. However, the secure storage and sharing of the vast amounts of continuously generated data present significant privacy challenges, including the risk of data breaches. This paper introduces SmartCrypt, a system designed for secure data storage and sharing that facilitates analytics on encrypted time series data. SmartCrypt allows users to securely and selectively share their encrypted data through an innovative symmetric homomorphic encryption method. Simulation results indicate that SmartCrypt decreases query time by 17% and enhances throughput by 9% compared to the benchmark scheme.

Furthermore, SmartCrypt's scalability and flexibility make it an attractive solution for various IIoT applications. The system's performance and security features demonstrate its potential for widespread adoption. Future research directions include exploring SmartCrypt's applicability in other domains, such as healthcare and finance.

KEYWORDS: Security, Privacy, Scalability and Industrial Internet of Things.

I. INTRODUCTION

In smart manufacturing, Industrial Internet of Things (IIoT) devices produce a significant amount of time series data related to production, monitoring and maintenance that need to be processed and stored. Due to storage and processing constraints of IIoT devices, nowadays the data storage and processing functionalities are mostly shifted to the time series database in cloud platform, e.g., Azure Time Series Insight. Further, processing and storing the data in the cloud platform enhances ubiquitous access, scalability and sharing possibilities. However, secure data storing in the cloud poses significant privacy risks, including unauthorized access of production line efficiency data. To address privacy risks, encrypted databases have appeared as a promising solution.

The main advantage of this approach is that it allows data owners and third-party services to query encrypted data while maintaining both functionality and confidentiality. Recently, research in this domain has led to several encrypted databases, e.g., relational databases and batch analytics.

Secure time series data storing in the cloud and sharing them with third-party services come with unique performance and challenges that current encrypted data processing systems fail to meet.

To address these challenges, numerous databases have been devised, particularly for time series data. However, all these databases incur significant overhead during encrypted data processing. Besides, another key challenge in smart manufacturing is that privacy should co-exist during queries on data statistics, e.g., finding the standard deviation, which usually indicates sharing data to be examined by third-party services. Further, data sharing must be fine grained as it is often unnecessary to provide third-parties free access to the data.

Instead, data owners might like to (i) share only statistical computation of the data, e.g., mean, sum, max, min, (ii) restrict the granularity at which such statistical computations are reported, e.g., per-minute, perhour, (iii) restrict the time interval over which queries are generated, e.g., February 2021, and (iv) a combination of earlier three choices. We believe that support for encrypted query processing should go together with access control to restrict the scope of data that users may query. The sharing procedure for data stream stored in the time series databases is considerably distinct from traditional databases. Particularly, in smart manufacturing, various levels of production process continuously push data streams to the cloud, where numerous services can subscribe to access and analyze data streams. Furthermore, often there is a requirement to aggregate and analyze time series data from different production processes collaboratively.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202046

This indicates that we require to design an end-to-end encryption technique that is compatible with this sharing procedure. Most of the existing state-of-the-art security solutions are designed for relational databases instead of time series database . Although, the researchers in have proposed a security mechanism for storing and sharing of data streams, it is vulnerable to the malleability attacks. Besides, the existing time series databases failed to provide suitable access policies to allow data owners a fine-grain protection during selective and secure sharing of data streams with third-party services in multi-user smart manufacturing settings. Our main contributions in this paper are three fold.

1. We design SmartCrypt, a symmetric homomorphic encryption-based access control technique for flexible and fine-grain sharing of encrypted data streams.
2. We introduce a Homomorphic Message Authentication Code (HomMAC) based verification technique that supports source authentication and provides data integrity checks.
3. Our experimental results show that SmartCrypt significantly improves the query time, latency and throughput compared to the state-of-the-art realization, TimeCrypt.

1.1 MOTIVATION

The advent of the Industrial Internet of Things (IIoT) has revolutionized the way industries operate, with vast amounts of time-series data being generated from sensors, machines, and devices. However, the sheer volume and sensitivity of this data pose significant security challenges, making it an attractive target for cyber-attacks and data breaches. The consequences of such breaches can be catastrophic, resulting in financial losses, reputational damage, and even compromising national security. Furthermore, the traditional encryption methods used to protect this data often hinder its utility, making it difficult to analyze and share. Therefore, there is a pressing need for innovative solutions that can ensure the secure storage and sharing of time-series data in IIoT applications, without compromising its utility. This project aims to address this critical gap by developing SmartCrypt, a robust and efficient security framework for time-series data in IIoT.

1.2 PROBLEM DEFINITION

The increasing adoption of Industrial Internet of Things (IIoT) technologies has led to an exponential growth in the generation of time-series data from sensors, machines, and devices. However, this data explosion has created significant security challenges, particularly in terms of ensuring the confidentiality, integrity, and authenticity of the data. The traditional encryption methods used to protect this data often compromise its utility, making it difficult to analyze, share, and utilize in real-time applications. Moreover, the existing security solutions are often inadequate, inefficient, and inflexible, failing to address the unique security requirements of IIoT ecosystems. Specifically, the problems that arise from the current state of affairs are threefold: (1) security threats, such as data breaches and cyber-attacks, which can compromise the confidentiality and integrity of the data; (2) performance overhead, resulting from the computational complexity of traditional encryption methods; and (3) limited scalability, hindering the ability to efficiently store and share large volumes of time-series data. Therefore, there is a pressing need for innovative security solutions that can efficiently and effectively protect time-series data in IIoT ecosystems, without compromising its utility or scalability. to get a pattern.

1.3 OBJECTIVE OF THE PROJECT

The primary objective of the SmartCrypt project is to design, develop, and evaluate a robust and efficient security framework for storing and sharing time-series data in Industrial Internet of Things (IIoT) ecosystems. The project aims to ensure the confidentiality, integrity, and authenticity of time-series data through innovative encryption techniques, while minimizing the performance overhead associated with traditional encryption methods. Additionally, SmartCrypt seeks to enable secure and selective sharing of time-series data among authorized parties, prevent unauthorized access, and demonstrate the scalability and flexibility of the proposed security framework in various IIoT applications. By achieving this objective, SmartCrypt aims to provide a reliable and efficient solution for securing time-series data in IIoT ecosystems, thereby enhancing the overall security and trustworthiness of these systems.

II. LITERATURE SURVEY

Big data analytics over encrypted datasets with seabed (2016).

AUTHORS: A. Papadimitriou, R. Bhagwan, N. Chandran, R. Ramjee, A. Haeberlen, H. Singh, A. Modi, and S. Badrinarayanan

Today, enterprises collect large amounts of data and leverage the cloud to perform analytics over this data. Since the data is often sensitive, enterprises would prefer to keep it confidential and to hide it even from the cloud operator. Systems such as CryptDB and Monomi can accomplish this by operating mostly on encrypted data; however, these

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202046

systems rely on expensive cryptographic techniques that limit performance in true “big data” scenarios that involve terabytes of data or more.

This paper presents Seabed, a system that enables efficient analytics over large encrypted datasets. In contrast to previous systems, which rely on asymmetric encryption schemes, Seabed uses a novel, additively symmetric homomorphic encryption scheme (ASHE) to perform large-scale aggregations efficiently. Additionally, Seabed introduces a novel randomized encryption scheme called Splayed ASHE, or SPLASHE, that can, in certain cases, prevent frequency attacks based on auxiliary data.

InfluxDB Cloud(2021).
The recent great technological advance has led to a broad proliferation of Monitoring Infrastructures, which typically keep track of specific assets along time, ranging from factory machinery, device location, or even people. Gathering this data has become crucial for a wide number of applications, like exploration dashboards or Machine Learning techniques, such as Anomaly Detection. Time-Series Databases, designed to handle these data, grew in popularity, becoming the fastest-growing database type from 2019. In consequence, keeping track and mastering those rapidly evolving technologies became increasingly difficult. This paper introduces the holistic design approach followed for building NagareDB, a Time-Series database built on top of MongoDB—the most popular NoSQL Database, typically discouraged in the Time-Series scenario. The goal of NagareDB is to ease the access to three of the essential resources needed to building time-dependent systems: Hardware, since it is able to work in commodity machines; Software, as it is built on top of an open-source solution; and Expert Personnel, as its foundation database is considered the most popular NoSQL DB, lowering its learning curve. Concretely, NagareDB is able to outperform MongoDB recommended implementation up to 4.7 times, when retrieving data, while also offering a stream-ingestion up to 35% faster than InfluxDB, the most popular Time-Series database. Moreover, by relaxing some requirements, NagareDB is able to reduce the disk space usage up to 40%.

SHAMC: A secure and highly available database system in multi-cloud environment (2020).
AUTHORS: L.Wang, Z. Yang, and X. Song
Data owners outsource their databases into the cloud to enjoy the quality services provided by the cloud service providers. However, using cloud database makes the private data vulnerable and exposed to the attackers including malicious insiders. Many researchers try to find the way to encrypt the cloud database and execute queries securely on the ciphertext. In this paper, we propose a secure and highly available cloud database system in the multi-cloud named SHAMC. Specifically, we use the idea of secure multiparty computation and homomorphic encryption to store data and execute queries direct on the ciphertext. Besides, the entire database is stored in multiple clouds to avoid service interruption as well as solve the problems of permanent failure and vendor lock-in. We implement the prototype of SHAMC which supports all queries in TPC Benchmark™ H (TPC-H) on the top of the commercial cloud. SHAMC is proved to be highly available and cost-efficient. The evaluation shows it has an acceptable query overhead which is superior to other encrypted cloud databases.

Ghostor: Toward a secure data sharing system from decentralized trust(2020).
AUTHORS: Y. Hu, S. Kumar, and R. A. Popa
Data-sharing systems are often used to store sensitive data. Both academia and industry have proposed numerous solutions to protect the user privacy and data integrity from a compromised server. Practical state-of-the-art solutions, however, use weak threat models based on centralized trust—they assume that part of the server will remain uncompromised, or that the adversary will not perform active attacks. We propose Ghostor, a data-sharing system that, using only decentralized trust, (1) hides user identities from the server, and (2) allows users to detect server-side integrity violations. To achieve (1), Ghostor avoids keeping any per-user state at the server, requiring us to redesign the system to avoid common paradigms like per-user authentication and user-specific mailboxes. To achieve (2), Ghostor develops a technique called verifiable anonymous history. Ghostor leverages a blockchain rarely, publishing only a single hash to the blockchain for the entire system once every epoch. We measured that Ghostor incurs a 4–5x throughput overhead compared to an insecure baseline. Although significant, Ghostor's overhead may be worth it for security- and privacy-sensitive applications.

Timecrypt: Encrypted data stream processing at scale with cryptographic access control(2020).
AUTHORS: L. Burkhalter, A. Hithnawi, A. Viand, H. Shafagh, and S. Ratnasamy
A growing number of devices and services collect detailed time series data that is stored in the cloud. Protecting the confidentiality of this vast and continuously generated data is an acute need for many applications in this space. At the same time, we must preserve the utility of this data by enabling authorized services to securely and selectively access

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202046

and run analytics. This paper presents TimeCrypt, a system that provides scalable and real-time analytics over large volumes of encrypted time series data. TimeCrypt allows users to define expressive data access and privacy policies and enforces it cryptographically via encryption. In TimeCrypt, data is encrypted end-to-end, and authorized parties can only decrypt and verify queries within their authorized access scope. Our evaluation of TimeCrypt shows that its memory overhead and performance are competitive and close to operating on data in the clear.

Practical homomorphic message authenticators for arithmetic circuits(2018).
AUTHORS: D. Catalano and D. Fiore
Homomorphic message authenticators allow the holder of a (public) evaluation key to perform computations over previously authenticated data, in such a way that the produced tag σ can be used to certify the authenticity of the computation. More precisely, a user knowing the secret key sk used to authenticate the original data, can verify that σ authenticates the correct output of the computation. This primitive has been recently formalized by Gennaro and Wichs, who also showed how to realize it from fully homomorphic encryption. In this paper, we show new constructions of this primitive that, while supporting a smaller set of functionalities (i.e., polynomially-bounded arithmetic circuits as opposite to boolean ones), are much more efficient and easy to implement. Moreover, our schemes can tolerate any number of (malicious) verification queries. Our first construction relies on the sole assumption that one way functions exist, allows for arbitrary composition (i.e., outputs of previously authenticated computations can be used as inputs for new ones) but has the drawback that the size of the produced tags grows with the degree of the circuit. Our second solution, relying on the D-DiffieHellman Inversion assumption, offers somewhat orthogonal features as it allows for very short tags (one single group element!) but poses some restrictions on the composition side.

Delegatable pseudorandom functions and applications(2013).
AUTHORS: A. Kiayias, S. Papadopoulos, N. Triandopoulos, and T. Zacharias
We put forth the problem of delegating the evaluation of a pseudorandom function (PRF) to an untrusted proxy and introduce a novel cryptographic primitive called delegatable pseudorandom functions, or DPRFs for short: A DPRF enables a proxy to evaluate a pseudorandom function (PRF) on a strict subset of its domain using a trapdoor derived from the DPRF secret key. The trapdoor is constructed with respect to a certain policy predicate that determines the subset of input values which the proxy is allowed to compute. The main challenge in constructing DPRFs is to achieve bandwidth efficiency (which mandates that the trapdoor is smaller than the precomputed sequence of the PRF values conforming to the predicate), while maintaining the pseudorandomness of unknown values against an attacker that adaptively controls the proxy. A DPRF may be optionally equipped with an additional property we call policy privacy, where any two delegation predicates remain indistinguishable in the view of a DPRFquerying proxy: achieving this raises new design challenges as policy privacy and bandwidth efficiency are seemingly conflicting goals. For the important class of policy predicates described as (1-dimensional) ranges, we devise two DPRF constructions and rigorously prove their security. Built upon the well-known tree-based GGM PRF family, our constructions are generic and feature only logarithmic delegation size in the number of values conforming to the policy predicate. At only a constant-factor efficiency reduction, we show that our second construction is also policy private. Finally, we describe that their new security and efficiency properties render our DPRF schemes particularly useful in numerous security applications, including RFID, symmetric searchable encryption, and broadcast encryption.

III. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Most of the existing state-of-the-art security solutions are designed for relational databases instead of time series database. Although, the researchers in have proposed a security mechanism for storing and sharing of data streams, it is vulnerable to the malleability attacks. Besides, the existing time series databases failed to provide suitable access policies to allow data owners a fine-grain protection during selective and secure sharing of data streams with third-party services in multi-user smart manufacturing settings.

Our main contributions in this paper are three fold.

- We design SmartCrypt, a symmetric homomorphic encryption-based access control technique for flexible and fine-grain sharing of encrypted data streams.
- We introduce a Homomorphic Message Authentication Code (HomMAC) based verification technique that supports source authentication and provides data integrity checks.
- Our experimental results show that SmartCrypt significantly improves the query time, latency and throughput compared to the state-of-the-art realization, TimeCrypt.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202046

DISADVANTAGES

- Vulnerable to Malleability Attacks
- Limited to Relational database
- Lack of Fine-Grain Access Policies
- Inadequate Security for Multi-User Settings

3.2 PROPOSED SYSTEM

This section provides details about our proposed scheme.

A. Storing of Encrypted Data In this section, we illustrate how SmartCrypt encrypts and stores the time series data streams in the cloud server.

- Symmetric Homomorphic Encryption: Let mi be a message to be encrypted from the message space [0, M − 1], i.e., mi ∈ [0, M − 1] and size of mi is an integer, where i = 1, . . . , n. Let ki be a randomly generated secret key stream used to encrypt mi , where ki ∈ [0, M − 1]. To encrypt mi , the data owner computes the ciphertext as ci = Eki (mi) = (mi + ki) mod M. In contrast, to retrieve mi for given ki , one can perform decryption as mi = Dki (ci) = (ci − ki) mod M.

B. Sharing of Encrypted Data In SmartCrypt, our objective is to allow data streams access permissions to the third-party services at arbitrary intervals or temporal ranges like from 13.00-Feb 05 till 12.00-Feb 06 2021. To achieve this, SmartCrypt partitions the data streams into fixed-length time segments or chunks of size Δ, e.g., 20 sec. Each chunk is then encrypted using a separate key from the key stream, subsequently indexed by the time window of the chunk. To achieve fine-grain access control and enable data owners to share encrypted data streams of desire intervals, we devise a hierarchical key derivation treeThe work presented here also takes into account the inconsistent data from rainfall and temperature datasets to get a consistent trend.

ADVANTAGES

- Secure Data Storage
- Symmetric Homomorphic Encryption
- Fine-Grain Access Control
- Data Integrity and Authentication
- Scalable and Efficient
- Enhanced Security

3.3 MODULES

- In this project we include four modules are used,
- Data Producer
- Data Consumer
- Data Owner
- Database Server

3.3.1 DATA PRODUCER

- It is the set of IIoT devices, e.g., appliances, services, which generate time series data. The main function of this actor is to ingest time series data into the cloud server and run SmartCrypt's client library, which manages data stream pre-processing and encryption.

3.3.2 DATA CONSUMER

- These are entities like third-party services who are authorized to access data owner's time series data and produce an added value, e.g., aggregate numerous data streams for monitoring, analyzing and visualizations.

3.3.3 DATA OWNER

- It owns the data stream and grants access permissions to its generated data stream. Data owners determine policies to selectively expose their data streams to data consumers. Based on the defined access policy, database server grants or denies data stream access requests.

3.3.4 DATABASE SERVER

- It is mainly responsible for storing encrypted data stream and giving access to data consumers following policies as defined by the data owner.
- In SmartCrypt, cloud server performs analytical and statistical queries over the ciphertext and sends back the ciphertext to the data consumer. Only the data consumer, which owns the correct keys can decrypt the ciphertext and obtain statistical result (e.g., mean/max/min) and analytic (e.g., trend detection). To augment fast queries and analytics, SmartCrypt creates in-memory encrypted indices.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202046

IV. SYSTEM DESIGN

The SmartCrypt system is designed to provide a secure and efficient framework for storing and sharing time-series data in Industrial Internet of Things (IIoT) ecosystems. The system architecture consists of a robust and scalable infrastructure that enables seamless integration with various IIoT devices and sensors. The Data Collection Layer is responsible for collecting time-series data from these devices, which is then transmitted to the Data Processing Layer for encryption and processing.

The Data Processing Layer is the core component of the SmartCrypt system, utilizing innovative symmetric homomorphic encryption techniques to encrypt the data. This enables secure computations and analysis on the encrypted data, without requiring decryption, thereby protecting the confidentiality and integrity of the data. The encrypted data is then stored in the Data Storage and Sharing Layer, which facilitates secure storage and selective sharing of the data among authorized parties.

The SmartCrypt system also incorporates advanced access control mechanisms, ensuring that only authorized users can access and manipulate the encrypted data. These mechanisms include attribute-based access control, role-based access control, and identity-based access control, providing a robust and flexible access control framework. Additionally, the system utilizes a distributed ledger technology to maintain an immutable record of all data transactions, providing an additional layer of security, transparency, and accountability.

4.1 SYSTEM ARCHITECTURE

The SmartCrypt system architecture is designed to provide a scalable, secure, and efficient framework for storing and sharing time-series data in Industrial Internet of Things (IIoT) ecosystems. The architecture comprises multiple layers, including the Data Collection Layer, Data Processing Layer, Data Storage and Sharing Layer, and Access Control Layer. Each layer is designed to perform specific functions, ensuring a robust and secure data management framework.

The Data Collection Layer serves as the entry point for time-series data generated by various IIoT devices and sensors. This layer is responsible for collecting, aggregating, and transmitting the data to the Data Processing Layer for further processing. The Data Processing Layer utilizes innovative symmetric homomorphic encryption techniques to encrypt the data, enabling secure computations and analysis on the encrypted data.

The Data Storage and Sharing Layer is responsible for storing the encrypted data and facilitating secure sharing among authorized parties. This layer incorporates advanced access control mechanisms, ensuring that only authorized users can access and manipulate the encrypted data. The Access Control Layer provides an additional layer of security, authenticating and authorizing users, and controlling access to the encrypted data.

The SmartCrypt system architecture also incorporates a distributed ledger technology, providing an immutable record of all data transactions, and ensuring transparency, accountability, and auditability. The architecture is designed to be scalable, flexible, and adaptable, enabling seamless integration with various IIoT devices, sensors, and applications. The System Architecture is shown in figure 1.

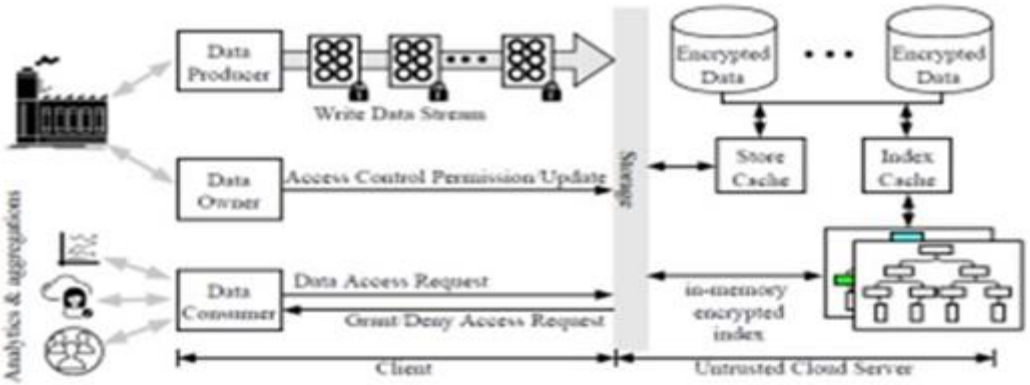


Fig 1: System Architecture

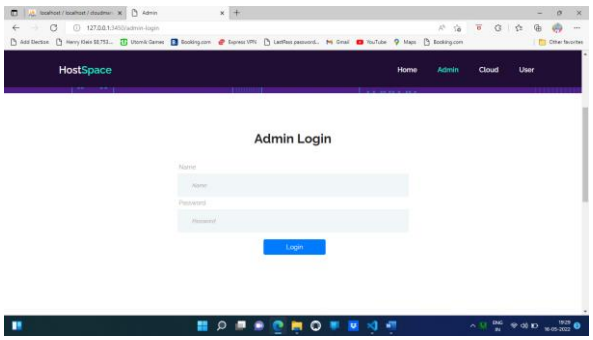
| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

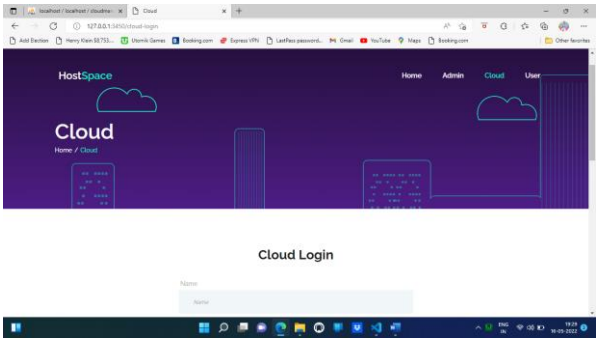
DOI:10.15680/IJARETY.2025.1202046

V. RESULTS

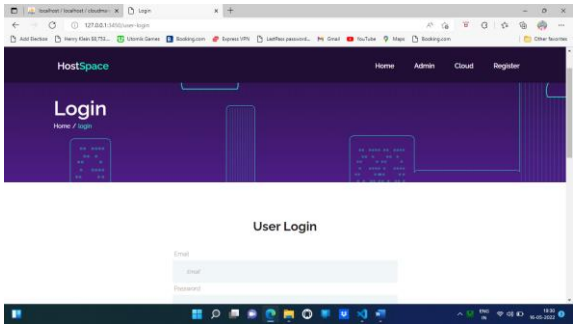
The following figures present the sequence of screenshots of the results.



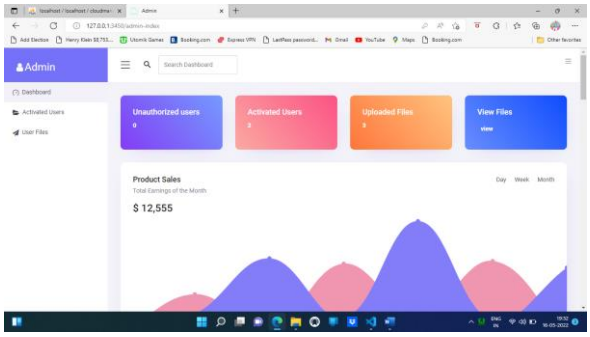
Admin Login



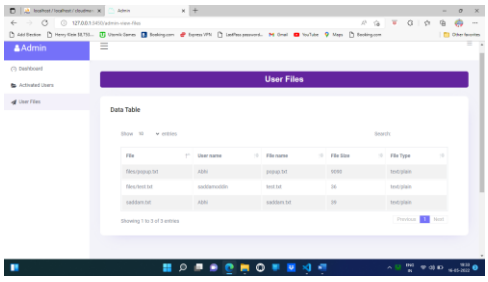
Cloud Login



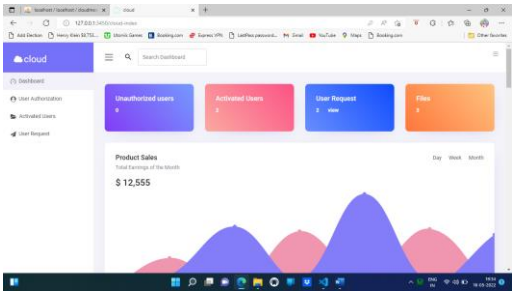
User Login



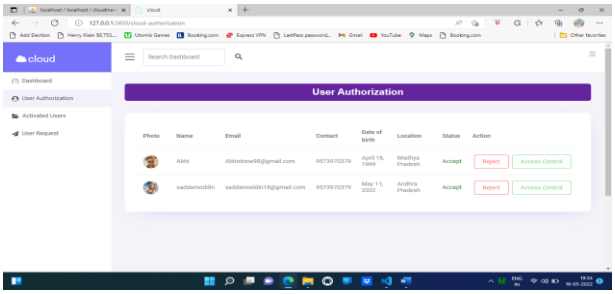
Admin Index



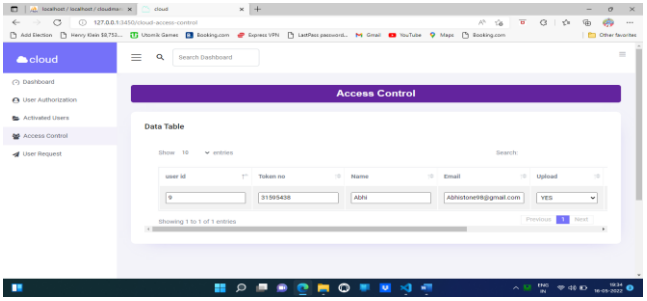
Admin view Files



Cloud Index



Cloud Authentication

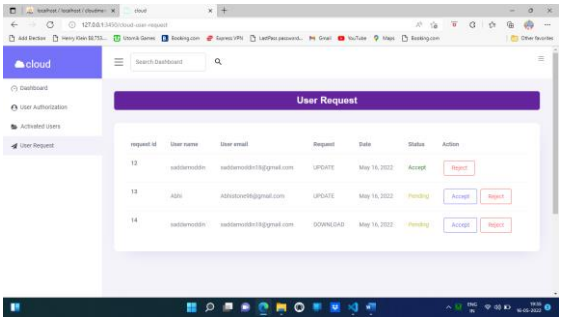


Access Control

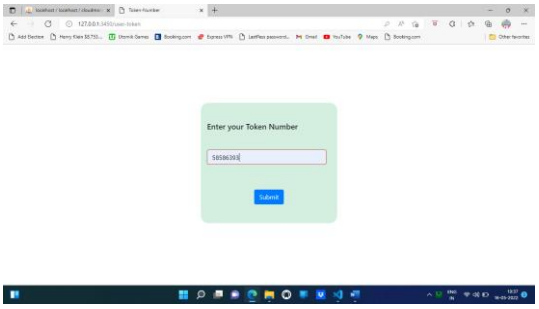
| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

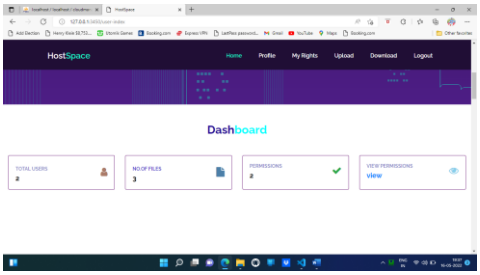
DOI:10.15680/IJARETY.2025.1202046



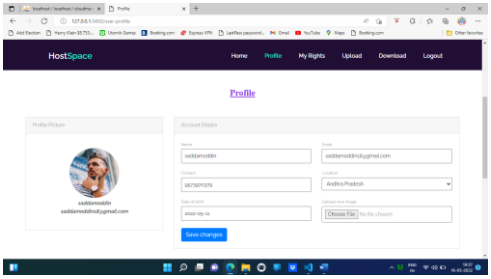
User Request



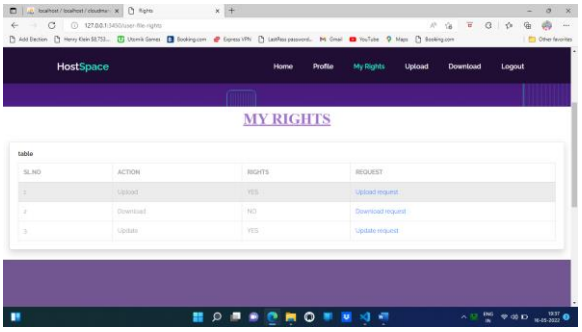
Enter Token Number



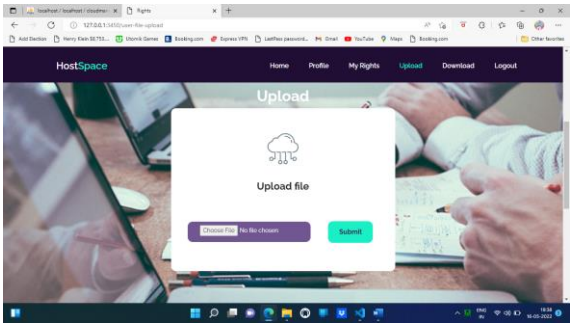
Dash board



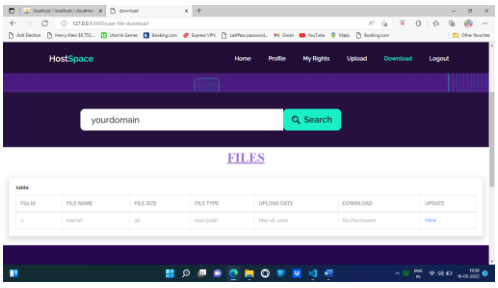
User Profile



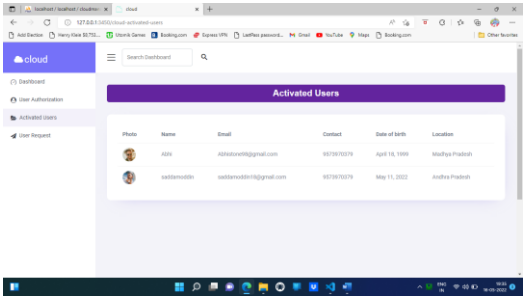
My Rights



Upload File



Files



Activated Users

VI. CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSIONS

In this paper, we proposed SmartCrypt, a data storing and sharing system that supports analytics over massive encrypted time series data. We introduced a novel symmetric homomorphic encryption-based access control technique tailored for time series data. SmartCrypt enables the execution of analytics over encrypted data streams and empowers

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202046

users to impose access restrictions on encrypted data streams considering their access control and privacy preferences. Our evaluation on real-world datasets show the feasibility of SmartCrypt as an authorization service for secure and fine-grain sharing, and performing analytics on large-scale time series data.

6.2 FUTURE WORK

As the SmartCrypt project continues to evolve, several avenues for future research and development emerge. One potential direction involves exploring the application of SmartCrypt in other domains, such as healthcare and finance, where sensitive data requires robust security and privacy protections. This could involve adapting the SmartCrypt framework to accommodate the unique security and regulatory requirements of these domains.

Another area of future work involves investigating the integration of SmartCrypt with emerging technologies, such as edge computing and artificial intelligence. By leveraging the capabilities of these technologies, SmartCrypt could be optimized for real-time data processing and analysis, enabling more efficient and effective decision-making in IIoT applications. Additionally, the incorporation of machine learning and deep learning techniques could enhance the security and accuracy of SmartCrypt's data analytics capabilities.

Furthermore, future work could focus on enhancing the scalability and performance of SmartCrypt, particularly in large-scale IIoT deployments. This could involve optimizing the system's architecture and algorithms for distributed computing environments, as well as exploring the use of new technologies, such as blockchain and distributed ledger systems, to support secure and decentralized data management. By addressing these challenges and opportunities, SmartCrypt can continue to evolve and provide robust security and privacy protections for time-series data in IIoT ecosystems.

REFERENCES

[1] Papadimitriou, R. Bhagwan, N. Chandran, R. Ramjee, A. Haeberlen, H. Singh, A. Modi, and S. Badrinarayanan, “Big data analytics over encrypted datasets with seabed,” in Proc. of 12th USENIX OSDI, 2016, pp. 587–602.
[2] InfluxDB, “InfluxDB Cloud,” [Online]: Accessed on February 06, 2021, https://www.influxdata.com/.
[3] L. Wang, Z. Yang, and X. Song, “SHAMC: A secure and highly available database system in multi-cloud environment,” Future Generation Computer Systems, vol. 105, pp. 873–883, 2020.
[4] Y. Hu, S. Kumar, and R. A. Popa, “Ghostor: Toward a secure datasharing system from decentralized trust,” in 17th USENIX NSDI, 2020, pp. 851–877.
[5] Marella, B.C.C., & Kodi, D. (2025). Fraud Resilience: Innovating Enterprise Models for Risk Mitigation. Journal of Information Systems Engineering and Management, 10(12s), 683–695.
[6] L. Burkhalter, A. Hithnawi, A. Viand, H. Shafagh, and S. Ratnasamy, “Timecrypt: Encrypted data stream processing at scale with cryptographic access control,” in Proc. of 17th USENIX NSDI, 2020, pp. 835– 850.
[7] D. Catalano and D. Fiore, “Practical homomorphic message authenticators for arithmetic circuits,” J. of Cryptology, vol. 31, no. 1, pp. 23–59, 2018.
[8] Kiayias, S. Papadopoulos, N. Triandopoulos, and T. Zacharias, “Delegatable pseudorandom functions and applications,” in Proc. of 20th ACM CCS, 2013, pp. 669–684.

International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975 Impact Factor: 8.152

www.ijarety.in editor.ijarety@gmail.com