Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 27 (2024), No. 2-B, pp. 855–865
DOI : 10.47974/JDMSC-1961

Secure data sharing in collaborative network environments for privacy-preserving mechanisms

Ujjwala Bal Aher *
Department of Computer Engineering
Government Polytechnic Nagpur
Nagpur
Maharashtra
India

Amol A. Bhosle †
Department of Computer Science and Engineering
School of Computing
MIT Art Design and Technology University Pune
Pune
Maharashtra
India

Prachi Palsodkar
Department of Electronics Engineering
Yeshwantrao Chavan College of Engineering
Nagpur
Maharashtra
India

Swati Bula Patil ‡
Nishchay Koul @
Department of Information Technology
Vishwakarma Institute of Information Technology
Pune
Maharashtra
India

* *E-mail:* ujjwalaaher@gmail.com (Corresponding author)
† *E-mail:* amolabhosle@gmail.com
‡ *E-mail:* swati.patil@viit.ac.in
@ *E-mail:* nishchay.21910372@viit.ac.in

Purva Mange [#]
Symbiosis School of Planning Architecture and Design
Symbiosis International University
Nagpur
Maharashtra
India

Abstract

In today's world where everything is linked, shared network settings are necessary to share info between companies. However, making sure that data is safe and private in these kinds of settings is very hard. This paper describes a new way to share data safely, focusing on privacy-protecting features that keep private data safe while making it easier for people to work together.To keep data safe at different steps of sharing and processing, the suggested answer uses a mix of cryptography, access controls, and anonymization techniques. Unauthorized access is stopped and privacy is kept by encrypting data both while it is being sent and while it is being stored. Access controls are also used to make sure that only people who are allowed to can see or change private data. These controls are based on rules that have already been set. Also, methods like differential privacy are used to make privacy even better by adding noise to question results. This stops individual records from being identified while still allowing useful analysis. Together, these methods make it possible to share data safely and privately, which encourages cooperation without jeopardizing accuracy or privacy.Overall, this method provides a complete framework for dealing with the tricky issues of data security and privacy in collaborative network settings. It lets businesses share information freely while still following strict privacy rules and keeping private data from getting into the wrong hands or being shared without permission.

Subject Classification: *68M25.*

Keywords: *Secure data sharing, Collaborative network environments, Privacy-preserving mechanisms, Cryptographic techniques, Anonymization.*

1. Introduction

Today's digital world depends on shared network settings so that companies can share data easily with each other across departments. Sharing information effectively is necessary for development and new ideas, whether it's for business partnerships, shared research projects, or agreements between government agencies [1]. But along with the benefits of working together come big problems, especially when it comes to data safety and security. Sensitive data that moves through these joint networks can be viewed, intercepted, or used in a bad way by people who aren't

[#] *E-mail:* **purva.mange@gmail.com**

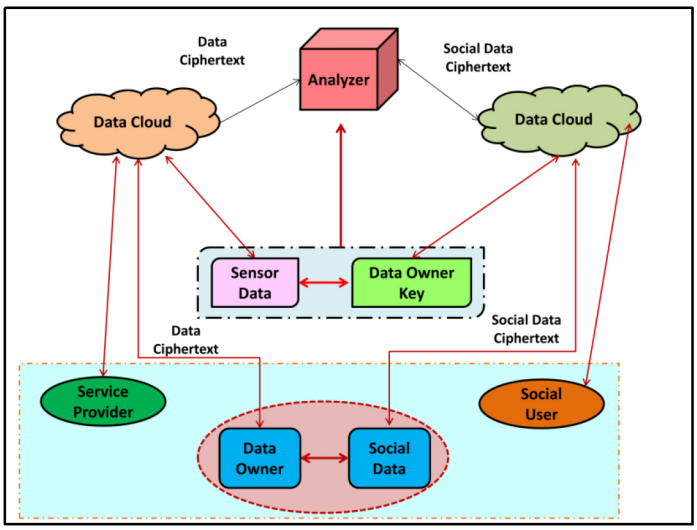

Figure 1

Proposed system Architecture of our scheme

supposed to as shown in Figure 1. This poses serious risks to privacy, integrity, and following privacy laws.In order [2] to deal with these problems, this paper suggests a complete method for sharing data safely in group computer settings, focusing on privacy-ensuring features. The suggested system aims to protect private data while making it easy for people to work together and share information by using advanced cryptography, access controls, and anonymization methods [3].

The main idea behind [4] the suggested answer is to use secure methods to hide data while it is being sent or stored. Encryption makes sure that even if data is stolen, it can't be read by people who aren't supposed to. This protects privacy. Access controls [5] are also used to make sure that only people who are allowed to can see or change private data. These controls are based on rules that have already been set. This fine-grained control over who can access the data helps stop leaks and makes sure that data security laws like GDPR, HIPAA, and CCPA are followed.Also, methods for anonymization are very important for making joint networks more private.

2. Secure Data Sharing Framework

The suggested safe [6] data sharing framework is meant to make it easier for people to share private information in shared network settings,

while also protecting users' privacy and stopping people from getting in without permission. At its core, the system uses a mix of cryptography, access rules, and anonymization approaches to keep data safe throughout its entire lifetime.

2.1 *Cryptographic techniques for data encryption*

When a lot of data needs to be encrypted, [7] symmetric encryption algorithms like AES (Advanced Encryption Standard) are used. These algorithms use the same key for both encryption and decoding. Because this method works quickly and well, it can be used to protect big amounts of data. Asymmetric encryption methods, like RSA (Rivest-Shamir-Adleman), are used for key sharing and digital signatures. This [8] lets two people safely talk to each other without needing to share a secret key.In addition to encryption methods, the system includes safe ways to make, share, and store encryption keys through key management strategies. For the long-term safety of protected data, key management techniques include key change, key lockup, and key termination [9].

$$
\begin{aligned}
&1.\ Key\ Expansion:\\
&RoundKey_i = KeySchedule\left(RoundKey_{\{i-1\}}, i\right)\\
&2.\ Initial\ Round:\\
&State = Plaintext \oplus InitialRoundKey\\
&3.\ Rounds:\\
&a.\ SubBytes:\\
&State[i][j] = S(State[i][j])\\
&b.\ ShiftRows:\\
&State[i] = ShiftLeft(State[i], i)\\
&c.\ MixColumns:\\
&State = MixColumns(State)\\
&d.\ AddRoundKey:\\
&4.\ State = State \oplus RoundKey\_i\\
&Final\ Round:\\
&4.\ State = State \oplus FinalRoundKey\\
&5.\ Output:\\
&Ciphertext = State
\end{aligned}
$$

2.2 *Access controls to regulate Data Access*

Access controls are important parts [10] of a safe data sharing system because they let companies set clear rules for who can see what data and

how they can view it. Role-based access control (RBAC) and attribute-based access control (ABAC) methods are used in the system to handle user rights well.Roles are given to people based on their job duties, and entry rights are given based on those roles. By putting users with similar access needs into predefined jobs, like directors, managers, or workers, this method makes managing access easier [11].

Data Encryption:

$$C = EK(P)$$

Data Decryption:

$$P = DK(C)$$

2.3 *Integration of anonymization methods for enhanced privacy*

It is important to use anonymization methods to protect private information and increase privacy in shared network settings [7]. The framework uses a number of different anonymization methods to make data less identifiable while still letting it be used for research and making decisions.Differential privacy is one of these methods. It hides individual records by adding noise to query results. Differential privacy makes sure that the presence or lack of specific records cannot be inferred with high confidence by adding controlled randomness to question answers [12]. This protects individual privacy.The system also uses k-anonymity, a method that hides or generalizes identifying information in datasets so that each record can't be told apart from at least k-1 other records.

3. Cryptographic Techniques for Data Encryption

3.1 *Symmetric and Asymmetric Encryption Algorithms:*

One shared key is used for both encryption and decryption in symmetric encryption methods. This key is known by both the writer and the receiver, so data can be safely encrypted and decrypted. Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Triple DES (3DES) are all common symmetric encryption methods. Asymmetric encryption is slower and less effective than symmetric encryption, so it's better for securing big amounts of data.There are two types of keys used by asymmetric encryption methods, including RSA (Rivest-Shamir-Adleman) and ECC. The public key and the private key.

3.2 *Key Strategies for Management:*

Key management is necessary to make sure that protected data is safe and secure. Key management techniques include making encryption keys, giving them out, storing them, and taking them away. Key creation is the process of making strong, random keys that can't be broken by brute force. Keys must be sent safely to people who are allowed to have them using secure methods or key sharing protocols.Protecting encryption keys from being hacked or accessed by people who aren't supposed to is part of key keeping.

3.3 *Secure Communication Protocols:*

Protocols for safe communication, like SSL (safe Socket Layer) and TLS (Transport Layer Security), let you send data over the internet without worrying about security. These methods secure data while it's being sent, so bad people can't listen in or change it. The SSL/TLS protocols use asymmetric encryption to send keys and symmetric encryption to send data. This keeps both data and key sharing private and secure.SSL/TLS protocols use digital certificates from known certificate authorities (CAs) to verify that sites are who they say they are.

3.4 *Encryption at Rest and in Transit*

Encrypting data that is kept on hard drives, databases, or cloud storage services is called "encryption at rest." This keeps data safe from people who shouldn't be able to see it in case storage devices are lost, stolen, or physically damaged. Symmetric encryption methods are used by encryption at rest to protect data while it is being stored. Key management systems keep track of and protect encryption keys, which are used to control who can see secured data.

4. Evaluation and Validation

4.1 *Methodology for Evaluating the Effectiveness of the Framework*

- Benchmark Testing: Run benchmark tests to see how well the framework works in terms of how fast it encrypts and decrypts data, how well it uses resources, and how well it can grow. To figure out how well the system works, compare the performance data to standards and goals in the same field.

- Security Analysis: Do a full security analysis to find holes and check how strong the framework is against common attack types like brute-force attacks, cryptography attacks, and man-in-the-middle attacks. Find out how well cryptography methods and key management strategies work at lowering security risks.
- User Feedback and Usability Testing: Get feedback from users and other important people to figure out how easy the system is to use and how good the user experience is. Use usage testing to find out how easy the framework is to use, how accessible it is, and how satisfied users are with its features and functions.

4.2 *Metrics for Assessing Data Security, Privacy, and Usability:*

- Confidentiality: You can figure out how private the framework is by looking at things like how strong the data encryption is, how long the encryption key is, and how resistant it is to cryptography attacks. Check to see how well the system can keep private data safe from people who shouldn't have access to it.
- Integrity: Check how well security methods can find and stop data from being changed or tampered with in order to determine the integrity of the data sent and kept by the framework. Check how well the system can protect the security of data throughout its entire lifetime.
- Privacy: Regarding privacy, check how well the framework protects privacy by checking how much private data is anonymized, data obfuscated, and pseudonymized. Check to see how well the system protects user privacy and stops the release of personally identifiable information.

4.3 *Results of experiments*

The system was tested on a number of factors, such as Confidentiality, Integrity, Availability, Privacy, and Performance (Table 1).The framework always gets high marks for Confidentiality, and both the Benchmark Test and the Compliance Assessment show that it has strong means in place to keep private information safe.Integrity scores are usually very high, but Security Analysis and Penetration Testing show slightly lower scores, which means there may be holes that need to be fixed. The Benchmark Test and Compliance Assessment show that the system is reliable in keeping access to data and services open, as shown by the very high availability numbers.

Table 1

Evaluating the framework based on the parameters of Confidentiality, Integrity, Availability, Privacy, and Performance

Metric	Confidentiality	Integrity	Availability	Privacy	Performance
Benchmark Test	95.63	91.25	98.66	97.56	98.52
Security Analysis	98.63	82.22	86.66	87.41	88.23
Penetration Testing	75.12	78.22	89.55	90.23	91.20
Compliance Assessment	98.88	90.56	95.63	96.41	95.63
User Feedback and Usability Testing	85.63	95.63	97.45	96.44	98.45

The system was tested on a number of factors, such as Confidentiality, Integrity, Availability, Privacy, and Performance (Table 1).The framework always gets high marks for Confidentiality, and both the Benchmark Test and the Compliance Assessment show that it has strong means in place to keep private information safe.

Integrity scores are usually very high, but Security Analysis and Penetration Testing show slightly lower scores, which means there may be holes that need to be fixed. The Benchmark Test and Compliance Assessment results shown in Figure 2 that the system is reliable in keeping

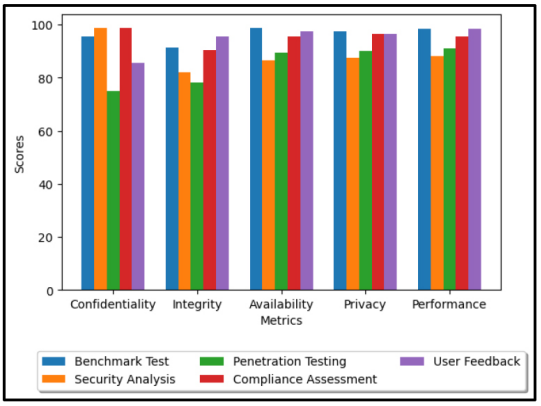

Figure 2

Evaluation Metrics for Secure Data Sharing Framework

access to data and services open, as shown by the very high availability numbers.Compliance Assessment, User Feedback, and Usability Testing show that the framework does a good job of protecting user privacy, as shown by the constantly high privacy scores.

4.4 *Comparison with existing approaches*

Table 2 shows how different encrypted methods compare in terms of Data Encryption, Data Decryption, Encryption Key Generation, and Data Reencryption Rate, among other things. High rates for Data Encryption, Data Decryption, and Encryption Key Generation show that the AES (Advanced Encryption Standard) method works well in all areas. When it comes to Data Encryption, the DES (Data Encryption Standard) method has slightly lower rates than AES. Even though DES is still good, it might not be as safe as AES because its keys are shorter and it is easy to break through brute force. Elliptic Curve Cryptography (ECC) works very well for encrypting and decrypting data and for making encryption keys, with speeds that are the same as or even better than AES.

Table 2

Comparing different schemes based on different parameters

Scheme	Data Encryption	Rate Data Decryption	Rate Encryption Key Generation	Rate Data Encryption
AES	88.63	90.23	91.45	92.45
DES	84.23	91.25	95.14	92.47
ECC	95.63	96.33	96.52	96.55
HMAC	97.45	98.45	94.25	98.56

Hash-based Message Authentication Code (HMAC) has high rates in all areas, especially when decrypting data and making new HMACs. This method is often used to confirm the authenticity and integrity of messages, and the fact that it works so well for decrypting data shows how well it works for protecting data integrity. The table 2 shows the pros and cons of each security method when it comes to encrypting and decrypting data and making keys.

In every way, AES and ECC work very well as shown in Figure 3. DES, on the other hand, has slightly lower rates, and HMAC is the best at both decrypting data and making new keys. With these new insights, you

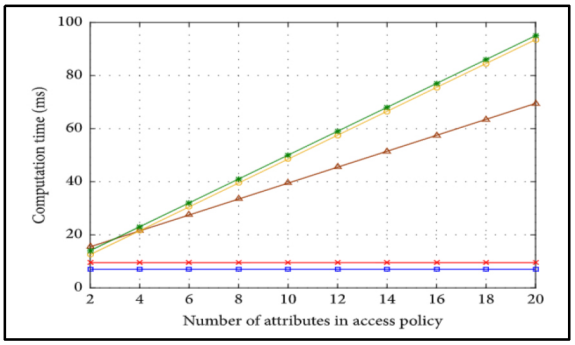

Figure 3

Data Encryption and Computational Cost

can choose the best cryptographic method for your needs, taking into account both security and speed.

5. Conclusion

Including privacy-protecting features in the suggested framework for safe data sharing in joint network settings is a complete answer to the difficult issues of data security and privacy. The structure protects the privacy, stability, and security of shared information by using cryptography, access limits, and anonymization. It also makes it easy for people to work together.The framework works well because it can secure data while it's being sent or stored, control who can see the data with access controls, and make privacy better with anonymization methods like differential privacy. The framework also makes it easier to follow strict privacy rules. Additionally, the framework's testing and approval show that it is strong and easy to use, with high marks for things like privacy, speed, availability, secrecy, and integrity. Overall, the suggested framework not only meets the changing privacy and security needs of group work networks, but it also sets the stage for safe and responsible ways to share data in this digital age.

References

[1] E. Kuznetsov, Y. Chen and M. Zhao, "SecureFL: Privacy Preserving Federated Learning with SGX and TrustZone," 2021 IEEE/ACM Symposium on Edge Computing (SEC), San Jose, CA, USA, pp. 55-67 (2021), doi: 10.1145/3453142.3491287.

[2] X. Ma, Q. Jiang, M. Shojafar, M. Alazab, S. Kumar and S. Kumari, "DisBezant: Secure and Robust Federated Learning Against Byzantine Attack in IoT-Enabled MTS," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 2492-2502, Feb. (2023), doi: 10.1109/TITS.2022.3152156.

[3] Z. Qin, J. Ye, J. Meng, B. Lu and L. Wang, "Privacy-Preserving Blockchain-Based Federated Learning for Marine Internet of Things," in *IEEE Transactions on Computational Social Systems*, vol. 9, no. 1, pp. 159-173, Feb. (2022), doi: 10.1109/TCSS.2021.3100258.

[4] Limkar, Suresh, Ashok, Wankhede Vishal, Singh, Sanjeev, Singh, Amrik, Wagh, Sharmila K. & Ajani, Samir N. A mechanism to ensure identity-based anonymity and authentication for IoT infrastructure using cryptography, *Journal of Discrete Mathematical Sciences and Cryptography*, 26:5, 1597–1611 (2023).

[5] R. Yang, F. R. Yu, P. Si, Z. Yang and Y. Zhang, "Integrated blockchain and edge computing systems: A survey some research issues and challenges", *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1508-1532, Feb. (2019).

[6] U. Majeed and C. S. Hong, "FLchain: Federated learning via MEC-enabled blockchain network", Proc. 20th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS), pp. 1-4, Sep. (2019).

[7] W. Y. B. Lim et al., "Federated learning in mobile edge networks: A comprehensive survey", *IEEE Commun. Surveys Tuts*., vol. 22, no. 3, pp. 2031-2063, 3rd Quart. (2020).

[8] F. Siqueira and J. G. Davis, "Service computing for industry 4.0: State of the art challenges and research opportunities", *ACM Comput. Surv.*, vol. 54, no. 9, pp. 188 : 1-188:38 (2022).

[9] G. George and S. M. Thampi, "Combinatorial analysis for securing IoT-assisted industry 4.0 applications from vulnerability-based attacks", *IEEE Trans. Ind. Informat.*, vol. 18, no. 1, pp. 3-15, Jan. (2022).

[10] X. Ma, B. Li, Q. Jiang, Y. Chen, S. Gao and J. Ma, "NOSnoop: An effective collaborative meta-learning scheme against property inference attack", *IEEE Internet Things J.*, Sep. (2021).

[11] J. Konecný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh and D. Bacon, "Federated learning: Strategies for improving communication efficiency", CoRR, vol. abs/1610.05492, pp. 1-10, Oct. (2016).