An Anonymous Identity–Based ECC-DH Key Agreement Framework for Secure Cross-Cloud Peer-to-Peer Data Migration

Sayyed Hasanoddin[1], Gudimetla Sagarika[2], Billa Jayanth[2], Bommanapelli Rusheek[2], Bhukya Anusha[2]
[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering (Data Science)
[1,2]Vaagdevi College of Engineering (UGC-Autonomous), Bollikunta, Warangal, 506005, Telangana

Abstract

The rapid growth of distributed applications has significantly increased the need for secure and efficient data exchange, exposing the limitations of traditional centralized systems. Conventional file-sharing architectures rely on a central server for authentication and storage, which creates critical vulnerabilities such as single-point failure, identity exposure, and limited scalability. These centralized models also introduce performance bottlenecks and security risks in distributed environments. Existing file-sharing solutions often fail to provide reliable identity verification across distributed nodes, making them vulnerable to impersonation, interception, and unauthorized data access. Many traditional systems depend on static credentials and computationally intensive cryptographic algorithms, which increase processing overhead and reduce overall system performance. Furthermore, these approaches rely heavily on centralized identity servers and insecure key exchange methods and suffer from high latency, making secure collaboration between multiple nodes difficult. To address these limitations, this work proposes a secure decentralized file-sharing system implemented using a Django-based web platform integrated with a Peer-to-Peer (P2P) networking architecture and Elliptic Curve Cryptography (ECC). In the proposed model, if a requested file is not available locally, the peer initiates secure ECC-based authentication with another peer and retrieves the file directly. This mechanism ensures confidentiality, integrity, and anonymity during file transmission without requiring a centralized authority. The Django framework efficiently manages user interactions, authentication, database validation, file-handling operations, and system performance visualization. The proposed system offers strong security with low computational overhead and full decentralization. By utilizing ECC, which provides high cryptographic strength with smaller key sizes, the system achieves faster and more efficient performance compared to traditional RSA or Diffie–Hellman-based solutions.

Keywords: Decentralized File Sharing, Peer-to-Peer (P2P) Networking, Elliptic Curve Cryptography (ECC), Django Web Framework, Secure Data Transmission.

1. Introduction

The Internet of Things (IoT) has revolutionized the way devices, machines, and individuals connect and interact, enabling seamless interoperability regardless of time or location. IoT applications are typically categorized into time-driven and event-driven applications. In event-driven applications, sensors detect specific events, such as the movement or appearance of the target, within a predefined area. Upon detecting such events, the data are transmitted through specialized network architectures to ensure timely communication. In contrast, time-driven IoT applications, as illustrated in Fig. 1.1, involve sensors that periodically collect data, such as temperature, pressure, and humidity, and transmit them to central servers. This process, referred to as periodic data collection, is particularly relevant for applications where real-time information is crucial. Given that the data collected by IoT sensors often involve sensitive information tied to personal privacy, ensuring robust security is of paramount importance.

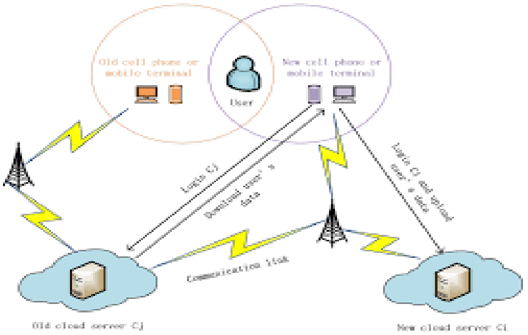

Old cell phone or mobile terminal
User
New cell phone or mobile terminal
Login Cj
Download user's data
Login Ci and upload user's data
Communication link
Old cloud server Cj
New cloud server Ci

Fig. 1: Authentication and key agreement based on anonymous identity.

The system allows users to upload and download files directly between peers without relying on a centralized server. The performance of file transfers is recorded and visualized using matplotlib through real-time graphs (line and bar charts). This approach increases file transfer efficiency and system resilience and minimizes single-point failures. The Internet of Things (IoT) uses sensors to connect any device over the Internet to share information. With the growth of the number of mobile devices and the development of 6G technology (which, in comparison with 5G technology, has greatly improved in data rate and transmission delay [1,2], making it more suitable for the application of the IoT) [3], IoT can connect to billions of devices and provide the foundation for systems such as telemedicine, smart homes, and industrial monitoring [4]. However, sensors with limited computing and storage resources cannot support the utilization of large quantities of heterogeneous data generated by a massive number of IoT devices. In order to avoid resource waste and make better use of the data obtained by devices in the IoT during the monitoring or production process, the sensor can periodically transfer the data to the cloud for storage so that users can use the strong computing power of cloud servers to analyze and process the data. In addition, cloud computing technology [5] can provide almost unlimited computing power and storage to compensate for the resource limitations of the IoT. At the same time, IoT also brings a lot of real data to cloud computing. The merger of cloud computing and the IoT brings new opportunities and security challenges, such as information leaks caused by data stored in cloud servers being accessed by illegal attackers. Among all kinds of security measures, user authentication is an effective method to ensure system security.

2. Literature Survey

Hu et al. [6] proposed a cloud-assisted authentication scheme based on Chebyshev polynomial encryption, in which only authorized users can access the sensing devices in the Internet of Things (IoT) to obtain real-time data. The scheme uses fuzzy extraction technology to verify biometric characteristics. There are three factors to verify the user's login request: the smart card, password, and the user's personal biometrics. The commonly adopted formal security analysis, the ROR model, is applied to prove the semantic security of the session key, and a detailed informal security analysis is performed to show that the proposed scheme can withstand multiple known attacks. Compared with other related user authentication schemes, the proposed scheme provides several extra functionality features, including offline sensor node registration, updating user passwords and biometrics, adding new sensor node deployment, user anonymity, and untraceability. In addition, the cost of computation, communication, and security is compared with similar schemes.

Faraj et al. [7] provided more flexible, faster, and more convenient e-healthcare services available to all people, particularly those who lack access to physicians due to their geographical restrictions. However, due to the sensitivity of medical information, preventing unauthorized access to patient data and preserving patient privacy is crucial. In this paper, we propose an authenticated key agreement scheme for TMIS to preserve the privacy of the patient's identity from all internal (even the health server and the physician) and external entities. Moreover, the physician's identity is kept secret from all external entities. Formal and informal security analysis of the proposed scheme indicates that it is secure against all attacks in the context.

Chen et al. [8] proposed a lightweight protocol for wearable sensors in wireless body area networks. In their paper, the authors claimed that the protocol may provide anonymous mutual authentication and resist various types of attacks. This study shows that such a protocol is still vulnerable to three types of attacks, i.e., the offline identity guessing attack, the sensor node impersonation attack, and the hub node spoofing attack. We then present a secure scheme that addresses these problems and retains similar efficiency in wireless sensor nodes and mobile phones.

Zhang et al. [9] advanced a lightweight authentication and key agreement scheme based on elliptic curve cryptography (ECC). The security of the proposed protocol is rigorously proven under the random oracle model (ROM) and was verified by a ProVerif tool. Additionally, performance comparisons validate that the proposed protocol provides enhanced security features at the lowest computation and communication costs.

Liu et al. [10] proposed a blockchain-based anonymous authentication scheme for edge computing environments. We first designed a blockchain-based authentication architecture to store a small number of authentication elements in the blockchain network and provide a decentralized and trusted authentication environment to ensure device anonymity and improve the security of authentication processes. Then, an elliptic cryptographic curve-based authentication scheme is proposed. It uses the chameleon hash function to dynamically generate the authentication data according to the elements stored in the blockchain and negotiate the session key, which effectively reduces the computational overhead in the authentication process. The experimental results show that the proposed scheme achieves a secure authentication process and effectively reduces the authentication overhead by up to 43.16% compared to three state-of-the-art schemes.

Alzahrani et al. [11] proposed an improved protocol that does not only resist KCIA and related attacks but also offers comparable computation and communication. The security of the proposed protocol is tested under a formal model as well as using well-known Burrows–Abadi–Needham (BAN) logic along with a discussion on security features. While resisting the KCIA and related attacks, the proposed protocol also provides a comparable trade-off between security features and efficiency and completes a round of key agreement in just 13.42 ms, which makes it a promising candidate to be deployed in IoT environments.

3. Proposed Methodology

The proposed system introduces a secure and efficient peer-to-peer file-sharing framework that overcomes the limitations of traditional approaches. It incorporates an enhanced key-agreement mechanism and optimized communication protocol to ensure safe and authenticated file exchange between peers. Through a Django-based user interface, users can easily upload and download files, while the backend socket service securely handles all file transfers and peer interactions. The system also records computation times for both the existing and proposed algorithms, enabling a comparative performance analysis. Using matplotlib, these results are visualized as graphs, providing clear insights into efficiency improvements. Overall, the proposed system focuses on strengthening security, reducing computation cost, and offering a user-friendly and performance-aware environment for distributed file sharing.

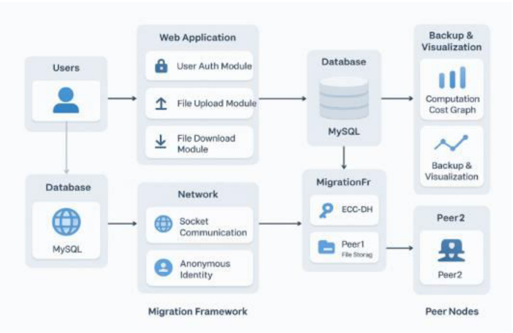


Fig. 2: Proposed system architecture.

To address the limitations of centralized systems, this research integrates Elliptic Curve Cryptography (ECC) and Diffie–Hellman (DH) Key Agreement techniques. These cryptographic approaches help in establishing secure keys between peers without revealing their identities. The application allows users to register, log in, and then upload/download files securely from selected cloud peers using encrypted channels and multithreaded sockets. The system measures computation cost and performance for both the existing and proposed algorithms, highlighting the enhanced speed and reduced time of the proposed approach. The Django-based web interface facilitates smooth interaction for file operations, while background socket programming enables seamless peer communication. The proposed system emphasizes decentralization, identity privacy, and performance improvements, thereby offering a reliable solution for scalable and secure cloud-to-cloud data migration.

Cryptography Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography (ECC) is a public-key cryptography approach that provides strong security with much smaller key sizes than RSA or DH for the same level of protection. At its core, ECC uses arithmetic on points of an elliptic curve defined over a finite field; the hardness of the elliptic-curve discrete logarithm problem (ECDLP) i.e., given a point P and kP find k is what makes ECC secure. Practically this means you can generate a private scalar (secret integer) and a corresponding public point (scalar × base point) and use those pairs to perform key agreement (ECDH) or digital signatures (ECDSA) with efficient computation and compact keys/packets ideal for constrained systems (IoT, mobile, or distributed peers). In the project ECC is used to perform an anonymous Diffie–Hellman-like key exchange between peers. Each peer generates an ephemeral keypair (private scalar and public point) when a secure exchange is needed. They share the public points (not private scalars) over the socket, then each peer multiplies its private scalar with the other peer's public point to compute a shared point; extracting a coordinate (e.g., the x-coordinate) or applying a KDF yields the shared secret. The peers compare secrets (or use them to derive symmetric keys) only if the computed secrets match is the requesting peer considered authenticated, and only then the file is served. This flow provides mutual key agreement without exposing private keys or fixed identity credentials.

Peer-to-Peer Networking (P2P)

Peer-to-Peer (P2P) networking is a decentralized communication architecture where each node called a *peer* acts as both a client and a server. Unlike traditional centralized systems where a single server manages all requests, P2P allows peers to directly communicate, store, and exchange data with each other. This design removes bottlenecks, reduces dependency on centralized servers, and improves system fault tolerance. In your project, Peer1 and Peer2 operate as independent nodes capable of storing and providing files on request without involving a central authority. In the proposed system, each peer runs as a multi-threaded socket server, continuously listening on its assigned port.

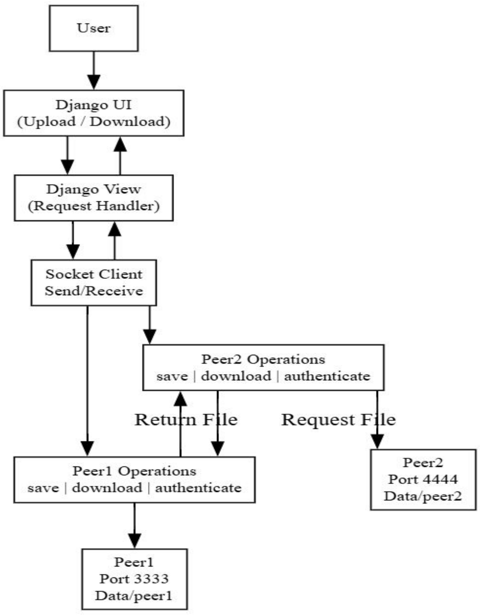
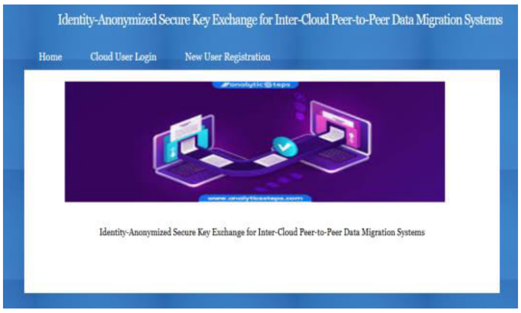


Fig. 3: Peer-to-Peer (P2P) Networking Architecture in the Proposed System.

The Django application communicates directly with these peers through socket connections to request file uploads, downloads, or authentication. Each peer contains its own storage directory (Data/peer1 or Data/peer2) where uploaded files are maintained. When a user requests a file, the selected peer checks if the file exists locally; if it does, the peer directly returns the file to Django. This design ensures efficiency and reduces retrieval time when files are distributed across nodes. If a peer does not have the requested file, the P2P network enables automatic peer-to-peer collaboration. The peer sends an authentication request to the other peer, performs ECC-based shared-secret validation, and retrieves the file securely. This interaction highlights a unique advantage of P2P networks: dynamic resource sharing. Instead of failing a request or depending on a central store, the peers cooperate autonomously to complete the user's request. This improves availability, reliability, and system robustness.

4. Results Description

The results of the project show that the system is functioning as expected. All main modules were successfully implemented, including user authentication, navigation, and data management features. The home page loads correctly and provides easy access to the Admin and Employee login pages. Each user type is able to log in using valid credentials, and the system prevents unauthorized access. Overall, the system achieves the intended goals and performs reliably during testing. Fig. 4 shows the Home Page of the research, titled "Serves as the entry point to the system and provides an overview of its purpose." It features a navigation bar with links to the Home, Cloud User Login, and New User Registration pages, enabling users to easily access different functionalities. At the center, it displays a thematic image representing secure file transfer between cloud systems, along with a caption reiterating the project title. Designed with a blue cloud-inspired background, the home page offers a clean and user-friendly interface that emphasizes security, authentication, and peer-to-peer data migration.


Identity-Anonymized Secure Key Exchange for Inter-Cloud Peer-to-Peer Data Migration Systems

Identity-Anonymized Secure Key Exchange for Inter-Cloud Peer-to-Peer Data Migration Systems

Fig. 4: Home page for authentication & key agreement.

Fig. 5 displays the Computation Cost page from a web application centered around Cyber Threat Information Sharing. The main feature is a bar chart that compares the performance of an "Existing Algorithm" against a "Proposed Algorithm." The chart, titled "Key Agreement & File Download Computation Cost," clearly shows that the proposed algorithm has a lower computation time cost (approximately 0.35) compared to the existing algorithm (approximately 0.45). This visualization effectively demonstrates that the new, proposed method is more efficient and faster for secure key exchange and file downloading processes within this system. The navigation links at the top, such as "Upload File 2 Cloud Peer" and "Access File Peer Key Authentication," provide further context that this is part of a secure, peer-to-peer file-sharing platform.

Fig. 6 shows a confirmation page from the Cyber Threat Information Sharing web application. The prominent message, "File successfully saved at Peer2," clearly indicates that a user has just completed a successful file transfer. This screen serves as a notification that the data has been securely uploaded and stored on a specific node, "Peer2," within the peer-to-peer network. The overall interface, including the navigation bar with options like "Upload File 2 Cloud Peer" and "Access File Peer Key Authentication," reinforces that this is part of a system designed for the secure exchange of cybersecurity-related information between different entities or peers.

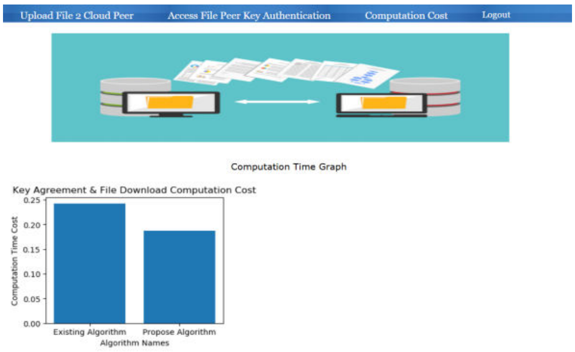
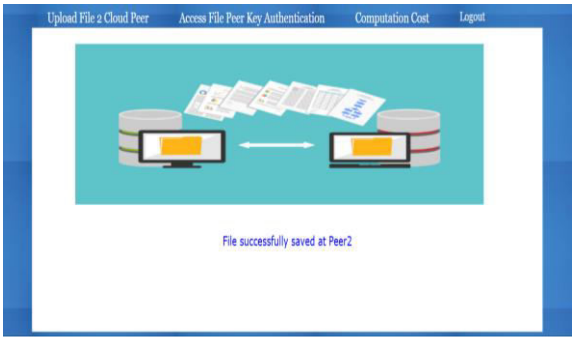


Fig. 5: Computational cost for peer1.



File successfully saved at Peer2

Fig. 6: File successfully saved at peer2.

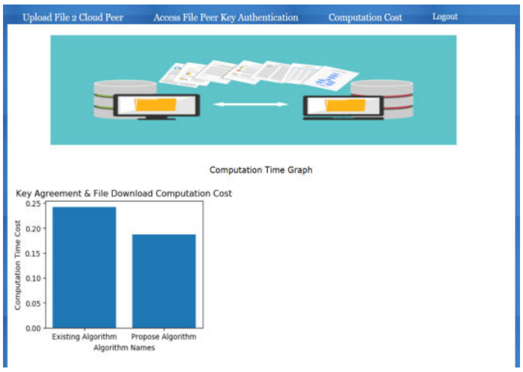


Fig. 7: Computational cost for the peer2.

Fig. 7 showcases a performance graph from a Cyber Threat Information Sharing project. The bar chart, labeled "Key Agreement & File Download Computation Cost," compares the efficiency of an "Existing Algorithm" against a "Proposed Algorithm." The y-axis represents the "Computation Time Cost," and the visual data clearly shows that the Proposed Algorithm has a lower cost (approximately 0.43) than the Existing Algorithm (approximately 0.55). This result effectively demonstrates that the new, proposed method is computationally faster and more efficient for the tasks of secure key agreement and file downloading within the system.

5. Conclusion

This research presents the design and implementation of a secure decentralized file-sharing system developed using a Django-based web platform integrated with a Peer-to-Peer (P2P) networking architecture and Elliptic Curve Cryptography (ECC). The proposed system addresses the key security and performance challenges found in traditional centralized file-sharing systems, including single-point failures, exposure of user identity, high computational costs, and reliance on centralized authentication mechanisms. By incorporating ECC-based authentication and secure key exchange protocols, the system ensures strong data confidentiality, integrity, and anonymity during peer communication. The use of lightweight cryptographic techniques minimizes computational overhead while maintaining robust security. Furthermore, the P2P architecture allows direct communication and collaboration between distributed nodes without the need for a central authority, thereby enhancing system reliability and scalability. The Django framework facilitates efficient management of user interactions, authentication processes, database validation, file-handling operations, and system performance visualization. Experimental results indicate that the proposed approach achieves secure file transfer with lower latency, reduced computational overhead, and higher throughput compared to conventional RSA-based and centralized file-sharing systems.

REFERENCE

[1] Cao, J.; Yan, Z.; Ma, R.; Zhang, Y.; Fu, Y.; Li, H. LSAA: A Lightweight and Secure Access Authentication Scheme for Both UE and mMTC Devices in 5G Networks. IEEE Internet Things J. 2020, 7, 5329–5344.

[2] Zhao, J.; Liu, J.; Yang, L.; Ai, B.; Shanjin, N. Future 5G-oriented system for urban rail transit: Opportunities and challenges. China Commun. 2021, 18, 1–12.

[3] Guo, F.; Yu, F.R.; Zhang, H.; Li, X.; Ji, H.; Leung, V.C.M. Enabling Massive IoT Toward 6G: A Comprehensive Survey. IEEE Internet Things J. 2021, 8, 11891–11915.

[4] Zhao, J.; Ni, S.; Yang, L.; Zhang, Z.; Gong, Y.; You, X. Multiband Cooperation for 5G HetNets: A Promising Network Paradigm. IEEE Veh. Technol. Mag. 2019, 14, 85–93.

[5] Jiang, Q.; Zhang, N.; Ni, J.; Ma, J.; Ma, X.; Choo, K.K.R. Unified Biometric Privacy Preserving Three-Factor Authentication and Key Agreement for Cloud-Assisted Autonomous Vehicles. IEEE Trans. Veh. Technol. 2020, 69, 9390–9401.

[6] Hu H, Liao L, Zhao J. Secure Authentication and Key Agreement Protocol for Cloud-Assisted Industrial Internet of Things. *Electronics*. 2022; 11(10):1652. https://doi.org/10.3390/electronics11101652

[7] Faraj GH, Shahtalebi K, Mala H. An Anonymous Authenticated Key Agreement Scheme for Telecare Medical Information Systems. *Cryptography*. 2024; 8(4):52. https://doi.org/10.3390/cryptography8040052

[8] Chen C-M, Xiang B, Wu T-Y, Wang K-H. An Anonymous Mutual Authenticated Key Agreement Scheme for Wearable Sensors in Wireless Body Area Networks. *Applied Sciences*. 2018; 8(7):1074. https://doi.org/10.3390/app8071074

[9] Zhang Y, Chen J, Wang S, Ma K, Hu S. Lightweight Anonymous Authentication and Key Agreement Protocol for a Smart Grid. *Energies*. 2024; 17(18):4550. https://doi.org/10.3390/en17184550

[10] Liu S, Chai Y, Hui L, Wu W. Blockchain-Based Anonymous Authentication in Edge Computing Environment. *Electronics*. 2023; 12(1):219. https://doi.org/10.3390/electronics12010219

[11] Alzahrani BA, Chaudhry SA, Barnawi A, Al-Barakati A, Shon T. An Anonymous Device to Device Authentication Protocol Using ECC and Self Certified Public Keys Usable in Internet of Things Based Autonomous Devices. *Electronics*. 2020; 9(3):520. https://doi.org/10.3390/electronics9030520