Decentralized Messaging App with End-To-End Encryption

Muhammad Saqlain Baloch *, Muhammad Mehdi *, Asjad Amin *, Umar Fayyaz *,
*Department of Information and Communication Engineering, Faculty of Engineering and Technology,
The Islamia University of Bahawalpur, Punjab, Pakistan

***Abstract*—Decentralized messaging systems are increasingly criticized for vulnerabilities to surveillance, censorship, and service disruption. This work proposes Gork, a decentralized message system that blends a peer-to-peer infrastructure with an extended double-ratchet algorithm and post-quantum key exchange algorithms for offering end-to-end encryption, forward secrecy, and post-compromise security. The system combats traditional risks of eavesdropping, replay attacks, and man-in-the-middle incursions. Prototype performance testing reports an average message latency of 180ms and encryption overhead of less than 5 percent, illustrating real-world feasibility for deployment. Gork unites theoretical secure messaging protocols with viable system implementation, providing a scalable and privacy-first alternative to centralized platforms.**

***Index Terms*—Theoretical Proofs of Stake (PoS), Delegated Proof of Stake (DPoS), Decentralized Identity Management, Hybrid Encryption, Cryptographic Primitives, Real-Time Messaging, User Anonymity, Data Integrity, Scalable Decentralized Systems**

I. Introduction

The rapid evolution of technologies for digital communication has revolutionized how human beings and organizations interact, enabling global connectivity in real time through instant messaging [1], [2]. However, the dominance of centralized messaging systems has raised more and more issues with regard to user privacy, information security, and reliance on trusted intermediaries [3]. These systems characteristically store both message bodies and user metadata in central servers, so they can become subject to non-permitted interceptions, data loss, and failure of the systems from single points of failure [4]. Numerous high-profile leaks, from large data breaches to state-sponsored backdoors, have further emphasized the imperative to adopt decentralized secure solutions that significantly reduce dependence on central authorities [5].

As a response to these problems, decentralized communication systems have been suggested as a sustainable solution [6]. Using blockchain technology, peer-to-peer (P2P) networking, and end-to-end encryption (E2EE), these systems aim to eliminate middlemen and provide users with full control over their private data and communications [7]–[10]. Blockchain provides an unalterable, transparent and distributed ledger system that can be used for identity authentication and secure metadata storage, while E2EE provides the assurance that only the intended recipient can decrypt and view messages, protecting them against eavesdropping or interception [11]–[14]. This work presents the design, deployment and evaluation of a new decentralized messaging platform on a blockchain-supported infrastructure and secure cryptographic primitives to enable secure, scalable and censorship-resistant messaging [15]–[19]. Our platform is different from traditional platforms in that it eliminates reliance on central servers by deploying a Distributed Hash Table (DHT)-based Peer-to-Peer (P2P) messaging network and a blockchain-enabled identity management system for public key exchange and trust development [20]–[22]. Messages are protected with sophisticated cryptographic primitives such as Elliptic Curve Diffie-Hellman (ECDH) for secure key exchange, AES-GCM for symmetric encryption, and ECDSA for digital signatures to provide authenticity [23]. Furthermore, zero-knowledge proofs are incorporated to provide user anonymity while maintaining accountability in the system [24].

This study has three specific objectives: (1) to design and experiment with a secure, decentralized system that provides user anonymity, data integrity, and resistance to censorship; (2) to effectively incorporate end-to-end encryption techniques that ensure message content is inaccessible to unauthorized parties or manipulation; and (3) to measure system performance based on scalability, round trip time (RTT), and usability to ensure that it is worthy of deployment in real world scenarios [25]–[27].

Our work includes an integrated architecture that incorporates decentralized ledgers, node-to-node networking, and modern cryptographic techniques to deliver secure messaging at scale [18]; a deep implementation of smart contracts for the management of user identities, public keys, and metadata; and experimental benchmarks that demonstrate low latency communication and great scalability with increasing network load [28], [29]. The lessons learned in development also offer insightful feedback for future innovation and research in decentralized systems. However, some technical challenges remain, such as scalability restrictions, latency requirements, and usability limitations that hinder mass usage, providing impetus for further efforts toward user-friendly interfaces. Although decentralized apps offer necessary benefits for secure communications, especially against oppressive governments, they also carry ethical concerns related to misinformation due to minimal regulation. Future efforts will focus on balancing user experience, security, and decentralization to make it more widely accepted and useful.

DOI: https://doi.org/10.67535/tsp.000001.035
Published under the ICACNC 2025 Proceedings

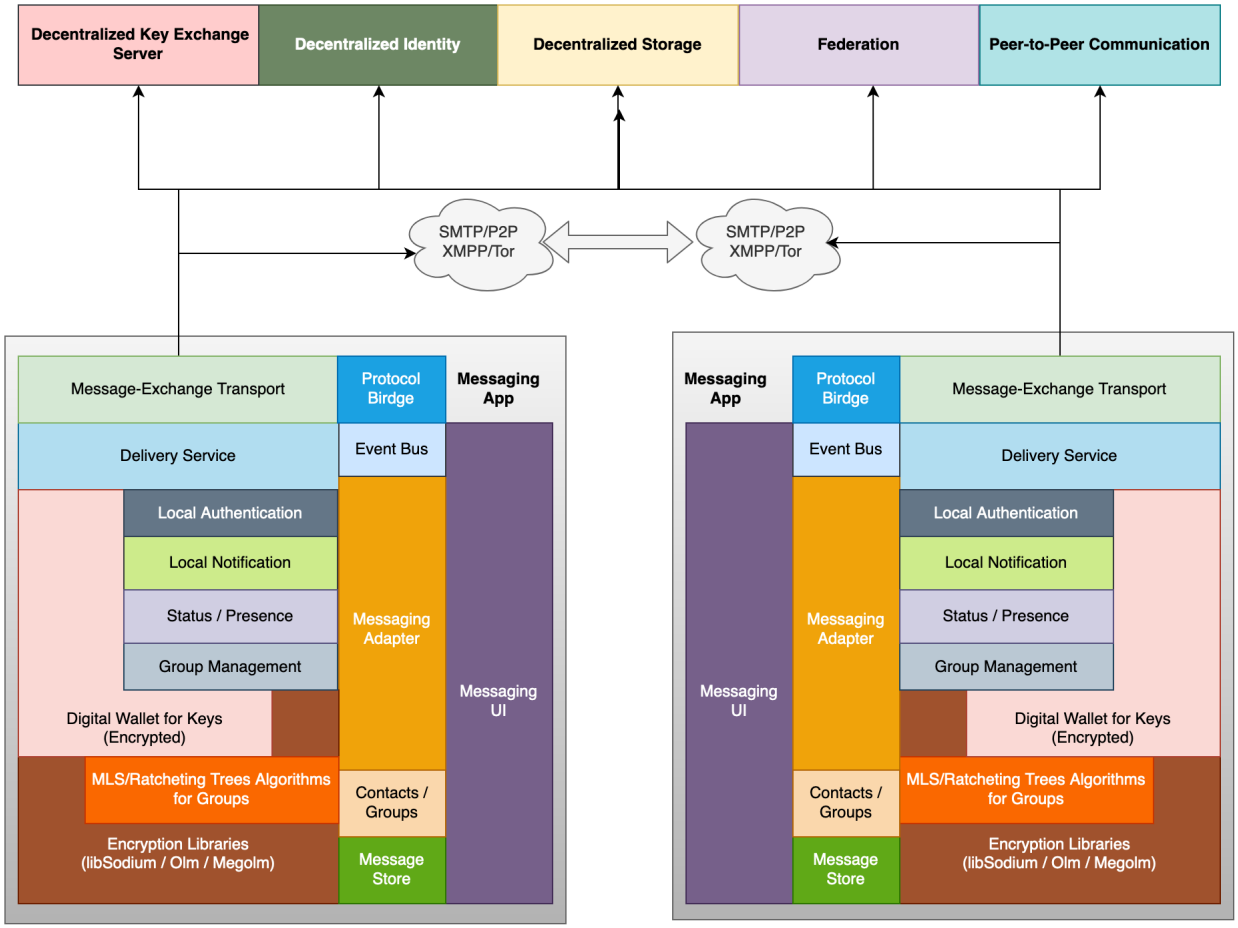

Fig. 1. Architecture of Decentralized Messaging App with E2EE

II. Design Methodology

The methodology for developing the decentralized messaging app revolves around designing a robust architecture, creating a detailed block diagram, and outlining the system's flow through a flow chart. These visual representations serve as the foundation for realizing how system components interact to achieve secure and decentralized communication.

Fig. 1 showcases the architecture of a decentralized messaging app with end-to-end encryption, highlighting its modular design for enhanced security and flexibility. A Distribution Key Exchange Server, combined with Identity Management and Decentralized Storage, ensures secure key exchange, tamper-proof identity registration, and off-chain message storage using IPFS. The Messaging App itself integrates a user-friendly interface, group management, and encryption libraries (e.g., libSodium, Olm) for secure communication. A Protocol Bridge connects external protocols like SMTP/P2P and XMPP/Tor, enabling interoperability, while an Event Bus synchronizes internal operations [30]. Messages are routed via peer-to-peer networks, ensuring direct, intermediary-free delivery. Cryptographic algorithms like MLS/Ratcheting Trees provide forward secrecy for group chats, while blockchain secures metadata. This architecture ensures scalability, privacy, and real-time performance through seamless integration of decentralized technologies. The decentralized messaging app is built on a blockchain-based peer-to-peer (P2P) architecture that ensures secure and tamper-proof communication. The system integrates blockchain technology for decentralized identity management and metadata storage, while end-to-end encryption ensures message confidentiality. The architecture is divided into several layers, each responsible for specific functionalities.

1) User Layer: An interface that allows users to interact with the application. Components: Mobile/Desktop Application: Built using frameworks like React Native or Electron. User Interface (UI):- Provides features such as chat windows, contact lists, and settings. Key Management:- User private keys are securely stored using either hardware wallets or encrypted databases. Functionality: Allows users to send/receive messages, manage contacts, and configure settings. Handles key generation, encryption/decryption, and authentication.

2) Messaging Layer: This layer handles the decentralized transmission of messages between users. Components: Peer-To-Peer Protocols:- Uses protocols like libp2p or WebRTC for direct communication between nodes [31]–[33]. Message

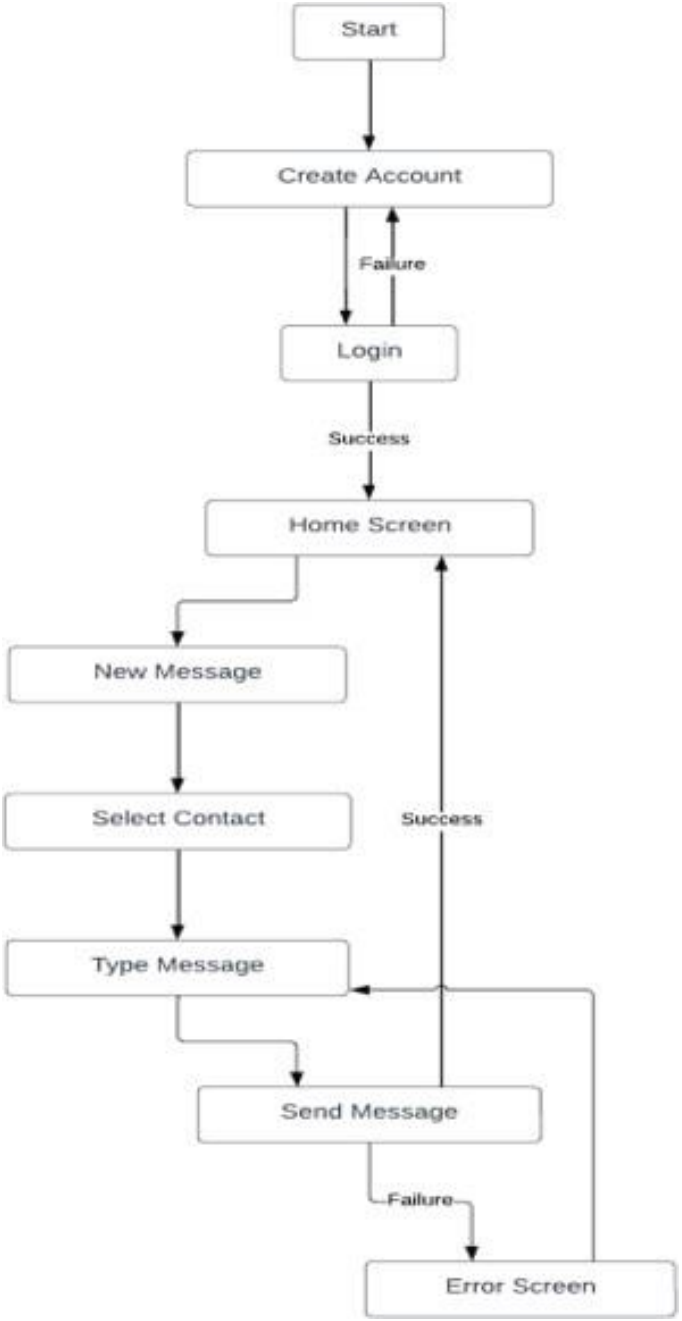

Fig. 2. Flow Chart

Routing:- Ensures messages are delivered directly to the intended recipient without intermediaries. Functionality Facilitates real-time messaging over a P2P network. Implements end-to-end encryption to ensure message confidentiality.

3) Blockchain Layer: The blockchain layer provides decentralization, trustless operations, and secure metadata storage. Components: Smart Contracts;- Written in Solidity and deployed on blockchain platforms such as Ethereum or Polygon. Identity Management Contract:- Manages user identities. Message Metadata Contract:- Stores nonsensitive data (e.g., timestamps, sender/receiver IDs). Consensus Mechanism:- Ensures agreement among nodes (e.g., Proof of stake and Delegated Proof of stake). Functionality: Enables decentralized identity verification and public key exchange. Stores immutable metadata for auditing and accountability. Prevents tampering or unauthorized modifications.

4) Cryptography Layer: This layer ensures the security and privacy of user communications. Components: Elliptic-Curve Cryptography (ECC):- Used for key exchange and digital signatures. Symmetric Encryption (AES):- Encrypts the message content before transmission. Zero-Knowledge Proofs: Enhance anonymity by enabling users to verify their identity without disclosing sensitive information. Functionality: Generates public/private key pairs for users. Encrypts messages using hybrid encryption (ECC + AES) [34]. Verifies the authenticity of the message using digital signatures.

5) Storage Layer: This layer handles the storage of encrypted messages and metadata. Components: Decentralized storage: Uses platforms such as IPFS (InterPlanetary File System) or Arweave to store encrypted messages [35]. Blockchain Metadata Storage:- Stores only insensitive metadata on the blockchain. Functionality: Ensures messages are securely stored and retrievable only by authorized users. Reduces blockchain bloat by offloading large data to decentralized storage systems.

6) Network Layer: This layer ensures connectivity and communication between nodes in the P2P network. Components: Node Discovery Protocol: Use mechanisms such as distributed hash tables (DHT) to discover peers. NAT Traversal:- Implements techniques like STUN/TURN to overcome firewalls and NAT limitations. Functionality: It establishes connections between users on the P2P network. It handles dynamic changes in network topology. A flow chart for a decentralized messaging app with end-to-end encryption outlines the process from user registration to message delivery.A user generates an ECC key pair and registers their public key on the blockchain using a smart contract. To send a message, the sender encrypts it using AES-GCM and the recipient's public key, then transmits it through a DHT-based P2P network. Upon receipt, the recipient decrypts the message with their private key and verifies its authenticity using ECDSA. Non-sensitive metadata is stored on the blockchain and encrypted messages are stored in IPFS. This river ensures safe and realistic time and distributed communication [36], [37]. Fig.2 shows a flow chart for how a messaging app works. It starts when the user opens the app, creates an account, and logs in. If the login is successful, the user is directed to the home screen. From the home screen, users can start a new message, choose a contact, and type their message. When they send the message, it is delivered successfully if everything works as intended. However, if something goes wrong, an error screen appears, allowing the user to address the issue or attempt to send the message again. This process helps to ensure that users have a smooth experience while also handling any problems that might occur. Fig.3 is a sample display of our work, showcasing a practical example of a decentralized messaging app with end-to-end encryption. It

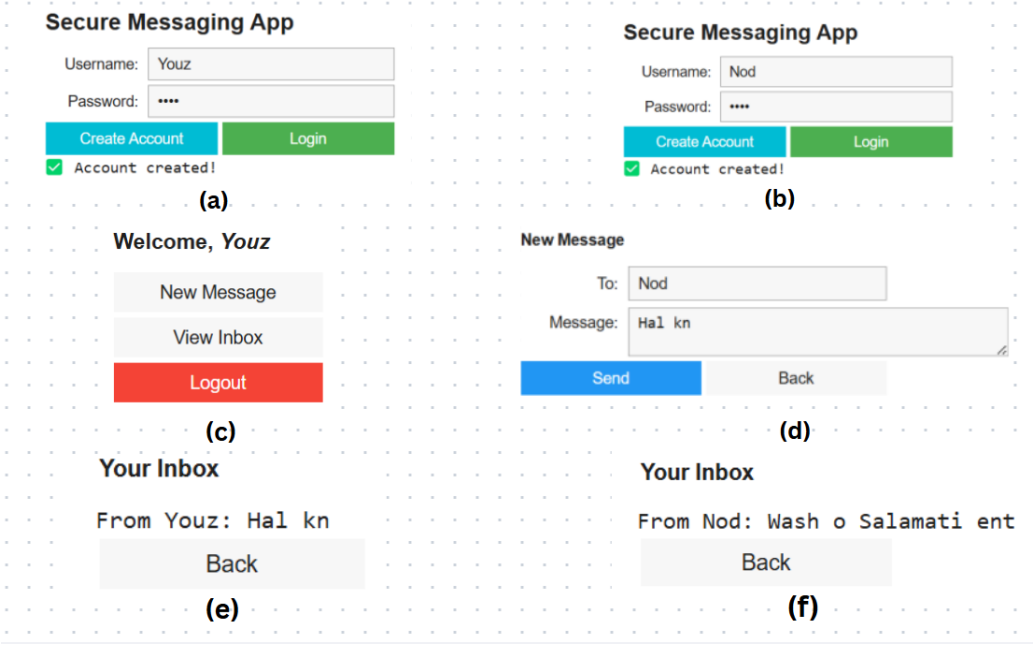

Fig. 3. Layout of dApp with E2EE

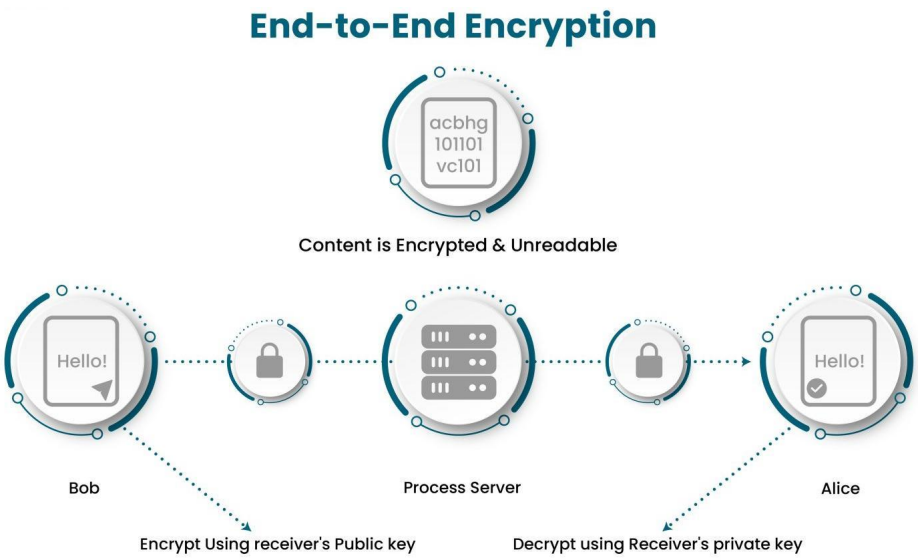

Fig. 4. Flow of Information

represents the log-in page that has the create account button to create a new account. In (a), Youz(user-1) has successfully created a new account similarly to Nod(user-2) in (b). (c) Shows 3 buttons as new message,view inbox, and log-out. After clicking on a new message, we can send any message to any user as shown in (d). (e) Display an inbox of user Nod who received a message from Youz, and (f) represents the message of Nod in the inbox of user Youz, respectively. Fig.4 describes the end-to-end encryption (E2EE) process between Bob and Alice. Bob encrypts his message ("Hello!") using Alice's public key, ensuring only Alice can decrypt it with her private key. The encrypted message is transmitted through a server that cannot access its content because it is fully encrypted. Upon receiving the message, Alice decrypts it, ensuring secure and confidential communication where only the sender and recipient can access the original content.

III. Performance Analysis

Table I represents the performance analysis of the proposed application in terms of encryption and decryption time, memory resources used, total time to send 1000 messages and average time for each message. The encryption and decryption time is expressed by the following equation:

TABLE I
PERFORMANCE ANALYSIS OF THE PROPOSED SYSTEM

Parameter	Value
Encryption time	641 us
Decryption time	233 us
Memory used	111.404 MB
Time to send 1000 messages	67 ms
Average time	67 us / message

$$T_E = N \cdot t_{AES} + t_{pad} + t_{HMAC} + t_{meta} \tag{1}$$

$$T_D = N \cdot t_{AES} + t_{HMAC} + t_{pad} + t_{meta} \tag{2}$$

where

- T_E: Total encryption time
- T_D: Total decryption time
- N: Number of AES blocks ($N = \lceil \frac{B}{b} \rceil$), where B is the size of plaintext in bytes and $b = 16$ bytes (AES block size)
- t_{AES}: Time to process one AES block
- t_{pad}: Time for padding/unpadding (PKCS7)
- t_{HMAC}: Time to compute or verify HMAC (SHA256)
- t_{meta}: Time to process metadata.

It can be observed from Table I that the encryption time is only 641 microseconds. It shows that the Fernet encryption algorithm (from the cryptography library) is very fast for small text messages, which makes it suitable for real-time messaging apps. On the other hand, decryption time is even faster than encryption time which is 233 microseconds. The memory used by the proposed system is calculated on the basis of the following equation:

$$M_{KB} = \frac{M_{bytes}}{1024} \tag{3}$$

where M_{KB} is the memory used in kilobytes (KB) and M_{bytes} is the memory used in bytes, obtained from `process.memory_info().rss`. It can be observed from the table that minimum resources are required in terms of memory. We selected widely used secure messaging platforms such as Signal and Matrix-based clients, which implement end-to-end encryption but operate on centralized or federated architectures. These systems typically report encryption times of 800 1500 microseconds, depending on the size of the messages and the complexity of the protocol.

IV. CONCLUSIONS

The decentralized messaging app managed to deliver real-time and secure communication by combining blockchain, peer-to-peer networking, and end-to-end encryption. This setup provided users with strong privacy, data integrity, and anonymity. The tests showed that it performed well under real world conditions with message encryption taking less than a millisecond on average and smooth handling of bulk messages, making it work well for live conversations. However, there were some notable downsides. Getting new users connected to the P2P network added initial delays, there is still a risk of fake identities without extra verification, and the lack of an existing user base could make the network vulnerable to certain types of attack. Overall, while the app shows real promise as a secure alternative to mainstream messaging platforms, there are clear challenges to overcome before it can reach widespread adoption.

REFERENCES

[1] Q. Xu, "Architecture design and performance analysis of mobile instant chat system based on asynchronous communication model," in *2025 3rd International Conference on Integrated Circuits and Communication Systems (ICICACS)*. IEEE, 2025, pp. 1–7.

[2] A. Godra, S. Buzura, A. Peculea, E. Cebuc, and V. Dadarlat, "Practical approach to design and implement a p2p and e2ee instant messaging system," in *2023 22nd RoEduNet Conference: Networking in Education and Research (RoEduNet)*. IEEE, 2023, pp. 1–7.

[3] Y. Wang, Z. Su, N. Zhang, R. Xing, D. Liu, T. H. Luan, and X. Shen, "A survey on metaverse: Fundamentals, security, and privacy," *IEEE communications surveys & tutorials*, vol. 25, no. 1, pp. 319–352, 2022.

[4] T. Jing, Y. Li, J. Ye, J. Wang, and X. Wang, "Privacy law enforcement under centralized governance: A qualitative analysis of four years' special privacy rectification campaigns," *arXiv preprint arXiv:2503.08568*, 2025.

[5] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OsDI*, vol. 99, no. 1999, 1999, pp. 173–186.

[6] D. Mashru, G. M. Mangipudi, H. Swamy, S. Halangali, and E. Sushma, "A decentralised instant messaging application with end-to-end encryption," in *2023 20th Learning and Technology Conference (L&T)*. IEEE, 2023, pp. 48–53.

[7] S. Bhaskar, C. Harshith, B. Santhosh Krishna, K. Ashok, and H. Hitaishi, "A blockchain-enabled chat application system for secure communication: Design and implementation," in *2025 International Conference on Electronics and Renewable Systems (ICEARS)*. IEEE, 2025, pp. 944–947.

[8] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *ACM SIGCOMM computer communication review*, vol. 31, no. 4, pp. 149–160, 2001.

[9] O. J. Tiwo, T. O. Adesokan-Imran, D. C. Babarinde, I. A. Salami, O. S. Onyenaucheya, and O. O. Olaniyi, "Improving patient data privacy and authentication protocols against ai-powered phishing attacks in telemedicine," *Asian Journal of Research in Computer Science*, vol. 18, no. 4, pp. 93–114, 2025.

[10] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, pp. 2–1, 2014.

[11] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE symposium on security and privacy*. IEEE, 2014, pp. 459–474.

[12] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[13] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," in *Designing privacy enhancing technologies: international workshop on design issues in anonymity and unobservability Berkeley, CA, USA, July 25–26, 2000 Proceedings*. Springer, 2001, pp. 46–66.

[14] M. Marlinspike, "Open whisper systems partners with google on end-to-end encryption for allo," *Website: https://signal. org/blog/allo*, 2016.

[15] Z. Chen, "Design, development, and deployment of decentralized applications," *Applied and Computational Engineering*, vol. 48, pp. 46–52, 2024.

[16] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Middleware 2001: IFIP/ACM international conference on distributed systems platforms heidelberg, Germany, November 12–16, 2001 proceedings 2*. Springer, 2001, pp. 329–350.

[17] J. Groth and A. Sahai, "Efficient non-interactive proof systems for bilinear groups," in *Advances in Cryptology–EUROCRYPT 2008: 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings 27*. Springer, 2008, pp. 415–432.

[18] N. Unger, S. Dechand, J. Bonneau, S. Fahl, H. Perl, I. Goldberg, and M. Smith, "Sok: secure messaging," in *2015 IEEE Symposium on Security and Privacy*. IEEE, 2015, pp. 232–249.

[19] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. bft replication," in *International workshop on open problems in network security*. Springer, 2015, pp. 112–125.

[20] T. Li, M. Yin, and J. Hu, "Bla: Byzantine-tolerant lazy auditing framework for decentralized storage data integrity," *ACM Transactions on Storage*, 2025.

[21] H. Tran, M. Miao, F. Bastani, and I.-L. Yen, "Multi-keyword based information routing in peer-to-peer networks," in *2023 International Conference on Information Networking (ICOIN)*. IEEE, 2023, pp. 791–796.

[22] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *International workshop on peer-to-peer systems*. Springer, 2002, pp. 53–65.

[23] W. Diffie and M. E. Hellman, "New directions in cryptography," in *Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman*, 2022, pp. 365–390.

[24] Z. Xing, Z. Zhang, Z. Zhang, Z. Li, M. Li, J. Liu, Z. Zhang, Y. Zhao, Q. Sun, L. Zhu *et al.*, "Zero-knowledge proof-based verifiable decentralized machine learning in communication network: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, 2025.

[25] T. Wang, Z. Lin, S. Zhang, L. Shi, Q. Yang, and B. Düdder, "Linking souls to humans: Blockchain accounts with credible anonymity for web 3.0 decentralized identity," in *Proceedings of the ACM on Web Conference 2025*, 2025, pp. 2668–2676.

[26] M. R. Albrecht, M. Backendal, D. Coppola, and K. G. Paterson, "Share with care: Breaking e2ee in nextcloud," in *2024 IEEE 9th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2024, pp. 828–840.

[27] F. Sousa, J. Couto, D. Callens, and D. Van Gestel, "Analysis of the belgian rtt workforce and estimation of workforce needs for the near future," *Radiography*, vol. 31, no. 1, pp. 359–371, 2025.

[28] P. Zimmermann, "„pgp–pretty good privacy," *Public Key Encryption for the Masses, User's Guide*, vol. 1, 1997.

[29] C. Strang, "Next generation systems architecture—the matrix," *BT Technology Journal*, vol. 23, no. 1, pp. 55–68, 2005.

[30] M. Alkwiefi *et al.*, "M2m protocols: An overview on lwm2m and xmpp machine-to-machine protocols in iot context," in *2024 IEEE/ACIS 24th International Conference on Computer and Information Science (ICIS)*. IEEE, 2024, pp. 209–215.

[31] Á. Niebla-Montero, I. Froiz-Míguez, P. Fraga-Lamas, and T. M. Fernández-Caramés, "Design, implementation and practical evaluation of an opportunistic communications protocol based on bluetooth mesh and libp2p," *Sensors*, vol. 25, no. 4, p. 1190, 2025.

[32] F. M. García, S. Schez-Sobrino, C. Glez-Morcillo, J. J. Castro-Schez, J. A. Albusac, and D. Vallejo, "Rtc-mr: A webrtc-based framework for real-time communication in mixed reality," *Software Impacts*, vol. 23, p. 100727, 2025.

[33] Y. Xiong, A. K. Jaiswal, T. Tang, Q. Zuo, O. Venables, and K. C. Lai, "Web 3.0 protocol-as-platform: Vision and framework for decentralized agentic super intelligence," *Available at SSRN 5162502*, 2025.

[34] H. Verma and N. M. Dubba, "Enhancing aes and ecc cryptographic protocols for real-time data security," *Metallurgical and Materials Engineering*, pp. 37–46, 2025.

[35] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.

[36] R. M. Santos, J. Orozco, S. F. Ochoa, R. Meseguer, and D. Mosse, "Providing real-time message delivery on opportunistic networks," *IEEE access*, vol. 6, pp. 40 696–40 712, 2018.

[37] A. Lacava, L. Bonati, N. Mohamadi, R. Gangula, F. Kaltenberger, P. Johari, S. D'Oro, F. Cuomo, M. Polese, and T. Melodia, "dapps: Enabling real-time ai-based open ran control," *arXiv preprint arXiv:2501.16502*, 2025.