Load Balancing Approaches In Cloud Computing

Ruba Awawdeh
Dept. of software engineering, Birzeit University
Ramallah, Palestine
rubaawawdeh4@gmail.com

***Abstract*—This paper highlights recent innovations in blockchain-based authentication protocols for distributed systems, including IoT networks, health care environments, and edge computing. It brings forward these fully decentralized authentication alternatives to traditional, centralized ones, proving that they are easily more secure, scalable, and resistant to attacks. Describe key technologies such as Proof-of-Authentication (PoAh), smart contracts, elliptic curve cryptography, and federated identity mechanisms. It emphasizes real-world testbeds and simulators for the performance evaluation of specific studies. Some advantages include speedy authentication, reduced computational load, and better resistance to cyber threats, making these new ideas promising for future secure systems.**

***Index Terms*—distributed system, Authentication, Reinforcement Learning.**

I. Introduction

As computing systems continue to fragment across Internet-of-Things devices, edge nodes, and cloud datacenter, establishing secure, scalable, and low-latency authentication has become a critical challenge. Traditional, centralized methods such as Kerberos and single sign-on introduce single points of failure, incur significant communication and computation overhead, and struggle to adapt to highly resource-constrained environments. To address these limitations, a growing body of research has turned to decentralized and blockchain-inspired techniques. A lightweight proof of authenticity scheme that replaces the energy-intensive Proof of Work with block signing and validator nodes per device, achieving block authentication in approximately 3.4 s in simulation and 3.8 s in Raspberry Pis [1]. This paradigm has also been extended to healthcare, demonstrating in NS 2.35 simulations of 20 hospitals and 400 patients how a public blockchain architecture can enable 'authenticate once, access everywhere', thus eliminating redundant reauthorization as patients move between facilities [3]. A three-layer design advancing the state-of-the-art was introduced (Device / Fog / Cloud) that uses hash chains and Ethereum smart contracts to register over 29000 device keys in 120 s and enforce mutual authentication without any trusted third party [5].

Building on these foundations, other works tackle complementary angles of decentralized authentication. One study integrates AES-based mutual authentication with a BFS-driven workload redistribution protocol in both MATLAB simulations and Raspberry Pi testbeds, showcasing secure task offloading with minimal latency [4]. Another work present a rigorous three-phase mutual authentication protocol: registration, client-AS-server exchange, and renewal, combining SHA256, RSA / DES, dynamic salts, and formal BAN logic proofs to defend against replay, dictionary, and man-in-the-middle attacks [6]. More recently,a smart contract-based framework was introduced to cleanly separate identity data from authentication logic, enabling interoperable, updatable modules across federated systems while preserving user anonymity and revocation capabilities [7]. Taken together, these studies illustrate the promise of decentralized blockchain-inspired methods to overcome performance bottlenecks, trust assumptions, and single-point-of-failure risks of classical approaches, although most remain validated only in simulation or small testbeds. In the sections that follow, we analyze the strengths and deployment challenges of each protocol and outline our contributions to the realization of real-world,

practical authentication on a scale.

II. Literature Review

In this section, we survey the leading decentralized and blockchain-inspired authentication schemes, highlighting how each addresses the trade-offs between security, performance, and scalability in resource constrained and multi-tenant environments. We then group these works by their core mechanisms, consensus-based, smart contract-enabled, and hybrid mutual-authentication protocols, to highlight their key innovations and remaining challenges.

A. Blockchain-Based Authentication in Distributed Systems

There are a set of papers that make experiments for using Blockchain to provide authentication in distributed systems:

- In [1], the authors proposed a novel scheme called Proof-of-Authentication (PoAh), designed to enable secure and scalable blockchain integration in resource-constrained distributed systems such as IoT and edge networks. Traditional blockchain mechanisms, like Proof-of-Work (PoW), require significant computational resources and energy, making them impractical for low-power devices such as sensors and smart home appliances. To address this, PoAh allows each device to construct a block of transactions (e.g., sensor readings or commands) and sign it using a private key before broadcasting it to peers. This digital signature ensures the block's authenticity and integrity without incurring the heavy computational costs of PoW [1]. Once the block has been signed, it is shared (broadcast) over the network. Trustable devices-in this situation, normal devices denoted as "validators' are to receive and check the block for valid signature. The process is just like using the sender's public key which works with the private key like a matching pair in order to confirm identity. Following a successful signature, the trusted device carries out another exercise on something referred to as Message Authentication Code (MAC). Meaning all information held within the block was not tampered with over-the-air. A block is secured hence the signature and MAC are consistent, and then that trusted device absolves a "Proof-of-Authentication" label on the same block and issues it to all other devices on the network. Testbed based validation was carried on PoAh, both in simulation and in real time. In the simulation stage, a prototype was developed in Python with the task of modeling a network of five participants, two of which are trusted nodes for block authentication. Each node created transactions, formed blocks, and signed them using asymmetric encryption. Blocks were then broadcast to the network, where the trusted nodes would verify their authenticity with the sender's public key and check the message authentication code (MAC). If authenticated, the block was then broadcasted again and added to the blockchain. The simulation showed that PoAh is able to validate blocks in an average time of about 3.34 seconds [1]. In an experimental testbed, PoAh was deployed in a real-world setup with five Raspberry Pi units simulating a typical setting for IoT. Three of these devices were collecting data from different sensors and the remaining two were used as trusted nodes. Each Raspberry Pi device was configured to have a 1.2 GHz ARM processor and to connect through the internet. The devices operated under the same PoAh procedure, creating blocks and authenticating them based on the described algorithm. Testbed results indicated that PoA has an average block authentication time of 3.8 seconds and an overall end-to-end processing time of 4.4 seconds, from the time the block was created up to the time the block became part of the blockchain. From these results, it can be concluded that PoAh is very much faster and suitable compared with conventional consensus methods like that of the PoW and hence has excellent potential for secure and efficient integration of blockchain into low-power distributed systems [1].
  The approved block is now received by all devices in the network, generating a unique digital fingerprint (hash) for the block and linking it to the previous block as the blockchain. This linking maintains the order and security of all transactions made in the past. Should a block ever falter on either of the two security checks

(signature or MAC), it simply gets deleted and overridden, while the process starts afresh with new data. This approach is efficient, as it skips the computationally intensive tasks involved in traditional blockchain while maintaining the highest order of security. It is also much faster, confirming a block in just about 3 to 4 seconds-this makes it ideal for systems requiring real-time processing in health monitoring, smart homes, or industrial sensors [1].

In [2], the authors introduce a decentralized solution to address key challenges in IoT environments, particularly secure authentication and efficient key management. Traditional IoT systems often depend on centralized authorities like key generation centers, which present vulnerabilities such as single points of failure and poor scalability. To counter these limitations, the authors propose a blockchain-based architecture that eliminates the need for trusted third parties by leveraging one-way hash chains and mutual authentication mechanisms. This approach enhances both scalability and security, making it well suited for the dynamic and distributed nature of modern IoT systems.

The proposed architecture comprises three layers: the Device layer, the Fog layer, and the Cloud layer. At the device layer, resource-constrained IoT devices are grouped into domains managed by Access Managing Nodes (AMNs) in the fog layer. These AMNs handle authentication and key management for the devices in their respective domains. The Cloud layer hosts Manager Nodes (MNs), which oversee the overall network and maintain multiple blockchains containing authentication and key management transactions. The system initializes by registering each device's public/private key pair onto Ethereum smart contracts, after which each device receives a signed license. Secure communication between devices is achieved through mutual authentication using values derived from one-way hash chains, with key values stored and validated via the blockchain [2].

To evaluate the proposed solution, the authors implemented the framework using a combination of cloud servers, laptops, and Raspberry Pi devices, deploying the system on Ethereum's Ropsten testnet. The results demonstrated strong scalability, with the ability to manage over 29,000 key pairs in 120 seconds and support for more than 40,000 devices. The framework also exhibited robust security, protecting against replay, man-in-the-middle, Sybil, and denial of service attacks. Despite these strengths, the current system only supports intra-network communication and assumes synchronized clocks across all devices and AMNs—an assumption that may not hold in real-world deployments. The authors suggest future improvements such as enabling inter-network communication, optimizing AMN load, and exploring energy-efficient blockchain alternatives. Overall, their work presents a promising foundation for secure, decentralized IoT infrastructures [2].

- In [3], the authors proposed a decentralized blockchain-based authentication scheme tailored for patients and medical staff within a network of affiliated hospitals. This approach aims to ensure secure and seamless identity verification across distributed healthcare environments, enabling features such as cross-institutional access control and minimizing the need for repeated authentications. The authors emphasize that the main issue is that traditional authenticated healthcare systems do rely on centralized authentication, which although effective, can be a slow and insecure means of transferring patients from one hospital to another. Therefore, a decentralized authentication architecture is proposed in this paper based on a public blockchain in which patients, doctors, and nurses are authenticated once and are then able to access services in any affiliated hospital without re-authentication. Each hospital in this proposed system would have its Nursing Station (NS), responsible for taking care of authentication and transactions onto the blockchain which might include entries like medical records or access logs. In order to substantiate the decentralized authentication architecture in a blockchain-integrated distributed hospital network, the authors conducted an experiment to prove the applicability of their architecture. Simulation-based experiments were conducted on the NS-2.35 network simulator with the extension of an open-source blockchain plug-in: two models were simulated

for comparison, one involving blockchain technology and the other a base model with conventional centralized authentication methods. The simulation environment was set up with given parameters in a 10 km × 10 km area with 20 hospitals, 400 patients, 100 doctors, 250 nurses, and 50 malicious users. The performance was evaluated for 50 simulation runs, with each run lasting 10 minutes and processing around 3000 health transactions. The key performance metrics include throughput, time overhead, response time, and energy consumption. Furthermore, the simulations were also run against common cyberattacks (i.e., spoofing and DDoS) in order to assess the security robustness of their system. Also, the authors performed a threat assessment of their system using the STRIDE threat modeling framework. Their results indicate that the proposed architecture indeed improved both performance and security over the baseline model; in particular, the architecture cut off the time-consuming process of re-authorization whenever a user moved from one associated hospital to another [3]. The system works like this: (1) a patient joins a hospital and gets authenticated by the NS which issues encryption keys depending on their unique device information; (2) the NS authenticates the transaction and creates a new block in the public blockchain; (3) if the patient goes to another hospital, that hospital checks the blockchain for the patient's authentication-no need for redundancy; (4) peer-to-peer communication with encryption can be used to exchange health data between these hospitals more securely. The methodology revolves around going for lightweight encryption (ARX/SPECK) to make the system apt for IoT devices and running simulations via NS-2.35 to test performance [3]. The findings from the simulations show that the proposed system significantly improves throughput, reduces time delay, saves energy, and is more resilient against cyberattacks compared to traditional models. It avoids repeated authentication and supports real-time, secure communication. However, a key limitation is that the solution has only been tested in simulation, not in real hospitals. Also, it depends on the NS unit in each hospital to be reliable and powerful enough to handle blockchain operations, which may present challenges in practical deployment. The authors suggest future work should involve building a real-world prototype to evaluate the system further [3].

- In [4], a decentralized framework is proposed to address critical issues of authentication and key management in the Internet of Things (IoT). This framework aims to secure communications and streamline key distribution among distributed IoT devices. Traditional IoT models rely on centralized authorities for key distribution, which introduces single point of failure and scalability limitations. The authors overcome these limitations by leveraging blockchain technology and one-way hash chains, eliminating the need for trusted third parties. Their approach supports secure mutual authentication and scalable key management, which is especially suitable for the diverse and dynamic nature of modern IoT systems [4].
  The proposed architecture comprises three layers: the Device layer, Fog layer, and Cloud layer. The Device layer hosts IoT devices grouped into domains managed by Access Managing Nodes (AMNs) in the Fog layer. These AMNs are responsible for authenticating devices and generating one-way hash chains for secure key exchange. At the Cloud layer, Manager Nodes (MNs) track blockchain activity and coordinate the network. The system uses smart contracts deployed on Ethereum's blockchain to store identities and keys. When devices communicate, they mutually authenticate using stored credentials and dynamically generate session keys through the hash chain mechanism, ensuring lightweight and secure communication [4].
  To test their system, the authors deployed it on a testbed using Raspberry Pi devices, laptops, and cloud servers, simulating IoT nodes, AMNs, and MNs, respectively. Their evaluations showed strong performance, handling tens of thousands of key pairs in seconds, while maintaining low overhead and robust security. The framework was successfully defended against common attacks such as replay, man-in-the-middle, and Sybil attacks. However, the current model only supports intranetwork communication and assumes synchronized clocks across nodes, which may not hold in real-

world scenarios. The authors recommend that future work include support for internetwork communication, load balancing for AMNs, and the exploration of energy-efficient blockchain platforms to enhance the performance and feasibility of the system at scale [4].

- A decentralized alternative to conventional federated identity systems such as Google ID—which, despite their convenience, carry drawbacks like centralized control, susceptibility to impersonation, and loss of user anonymity—was proposed in [7]. The solution introduces a blockchain-based authentication framework using smart contracts, with a particular focus on updatable contracts. This approach enables the system to modify authentication logic without altering the underlying data or requiring users to re-register, effectively addressing the rigidity and immutability challenges typical of blockchain-based systems [5]. The proposed system architecture consists of two primary smart contracts: the Main Contract (MC), which securely stores user data, and the Authentication Contract (AC), which processes authentication logic. By employing the delegatecall function in Ethereum, the MC can call the latest version of the AC without changing itself, allowing for modular updates to logic while preserving stored data. The authors also designed the framework to enable mutual authentication between users and service providers via an Identity Provider (IdP), enhancing both security and privacy. Formal verification through the ProVerif tool demonstrated the model's resistance to impersonation attacks and its preservation of user anonymity, all while removing the dependency on a central authority [5].
  Evaluation results showed that the proposed framework significantly improves performance, achieving a 71 percent increase in authentication speed over earlier models like those proposed by Xue et al. Furthermore, by avoiding centralized points of failure and enabling logic updates without service disruption, the solution enhances system scalability and maintainability. The modular smart contract design makes it an ideal candidate for modern distributed environments that demand secure, flexible, and high-performance authentication mechanisms. Ultimately, this approach provides a robust foundation for privacy-preserving federated login systems that can evolve over time without compromising user experience or system integrity [5].

B. Secure Authentication and Load Balancing in Edge Datacenters

The paper titled "Secure Authentication and Load Balancing of Distributed Edge Datacenters" focuses on solving two major problems in the growing field of edge and fog computing: (1) securely verifying edge datacenters (EDCs) before allowing them to share tasks, and (2) dynamically balancing workloads among EDCs to prevent some from becoming overloaded while others remain underused. In edge computing, data is processed closer to the source (e.g., IoT devices) rather than being sent all the way to a centralized cloud [6]. This helps reduce latency, but also introduces security and performance risks, especially when EDCs are deployed in open and distributed environments. Existing methods often address load balancing and authentication separately, or rely on static methods that don't adapt to real-time system load. This paper proposes an integrated and efficient solution that does both securely and in real time. The authors of this method define two critical phases of operation. The first phase is the authentication phase where each EDC gets an ID, a private key, and a shared key issued by the cloud at initialization. When one EDC wishes to connect with another, it sends a message encrypted for authentication along with its cloud credentials by using AES (symmetric encryption) [6]. The destination EDC authenticates that message with a contact to the cloud server for validation, hence allowing legitimate EDCs to participate. When both EDCs trust each other, they switch to an asymmetric encryption (where public/private keys are used) for exchanging load information. The second phase is resource balancing, which occurs when an EDC, detecting overload, broadcasts its current load state and listens for responses from EDCs that neighbor it. Each responder has its beyond-current-load and availability, also encrypted by the originator's public key. The originator finds the best EDC (least loaded) using the Breadth-First Search (BFS) method and then forwards the tasks there. This process minimizes processing delays and prevents the commissioning

of tasks to already-busy or malicious nodes [6]. It justifies their work by doing simulated experiments using MATLAB and doing actual tests using Raspberry Pi devices. It compares their suggested method against the classical approaches such as random, greedy, and proportional allocation, plus one of the known methods called CTOM. In terms of response time, success rate for task accomplishment, and the resource a task consumes, even with strong security mechanisms, the said method did better than the others. In reality, however, Raspberry Pi devices at their location served as simulated EDCs, communicating via Wi-Fi, using actual sensors, and a cloud backend. 100 percent successful and secure authentications were achieved, and load balancing was initially with low latency. The authors acknowledge that there are limitations in the assumption made about the total trust of the cloud, as not all scenarios might fit this, and the fact that the implementation was conducted in a relatively small network. Future works may focus on scaling this system, improving the trust models in decentralized environments. However, this paper provides a very robust practical foundation for secure, efficient edge computing systems [6].

C. Mutual Authentication Protocol for Distributed Systems

A novel architecture for a strong mutual authentication protocol in distributed systems was introduced in [7]. These environments, such as cloud computing, require strong mechanisms to ensure that no modifications have been made to data and to ascertain that the correct source of communication has been identified [7]. The proposed protocol lays stress on mutual authentication of clients, authentication servers, and multiple service servers, as well as being resilient to some common attacks such as brute force, dictionary, and man-in-the-middle attacks [7]. The authors designed a three-phase mutual authentication protocol. In the Registration Phase, clients register with an authentication server (AS) over HTTPS. During this process, the server applies cryptographic techniques, such as dynamic salt generation and hashing, to securely store authentication information. A unique key is created based on the client's ID and the generated salt. The Communication Phase consists of three subphases. The first component is called the Client-AS Mutual Authentication, whereby both the client and the AS authenticate each other based on shared keys, salt values, and encrypted authenticators. The second part of the Communication phase is called AS-Service Server Mutual Authentication. Here the AS verifies the identity of the service server using public/private key encryption for secure communication. The last phase is called Client-Service Server Mutual Authentication. This is whereby the client and service server authenticate each other using keys provided by the AS, creating a complete chain of trust [7]. In the final Renewal Phase, clients can refresh their credentials and cryptographic keys. This is done through a secure process that involves dynamic salt usage and ticket regeneration, ensuring continued session security and updated credentials. In order to evaluate the effectiveness of the protocol, the authors combined a behavioral study and a formal verification. In the Behavioral Study, they implemented key generation and distribution processes based on cryptographic primitives such as SHA-256, RSA, and DES. Indeed, they concluded that keys produced during such sessions were unique and unrelated owing to dynamic salts and the RotDy function. The mutual authentication between all entities has been formally verified through BAN Logic Analysis and proved both secure sharing and belief in centralized keys and authenticators. Further evidence of the protocol's strength to mitigate many threats comes from the Security Analysis: mutual authentication and session tickets make it difficult for a man in the middle to get between the parties; dictionary attacks are prevented by using dynamic salts and irreversible hash functions; and brute-force attacks are minimized using hashing, dynamic rotations, and time-limited tickets [7]. The paper clearly illustrates how mutual authentication is achieved in a distributed system. A ticket-based mechanism, along with cryptographic key exchanges and dynamic functions like S2KexS and RotDy, ensures that each participant in a session can verify the others' identity, thereby maintaining a secure and trusted environment. The suggested protocol has been able to effectively solve the two major security shortcomings in existing authentication systems for distributed environments. Mutual authentication along with dynamic parameters that are session-based with the cryptographic primitives makes it a much more secure method than existing ones including Kerberos and Single Sign-On (SSO). The authors conclude that modern distributed systems

require mutual authentication and that this protocol does not compromise performance for that goal [7].

III. Discussion

The reviewed studies explore a variety of blockchain-based authentication systems tailored to the growing needs of distributed environments such as IoT networks, healthcare systems, and edge computing. Despite their diverse application areas, these systems share several common goals: improving authentication speed, ensuring secure communication between devices, and eliminating the reliance on centralized authorities. One major strength apparent from among the studies is that there exist lightweight cryptographic methods, such as ECC, one-way hash chains, and smart contracts to support resource-constrained devices, including sensors and Raspberry Pi units. Such schemes not only reduce computation overhead, but also ensure highly reliable security standards that make them suitable for real-time applications. The systems also exhibit resilience to many forms of attacks, including, but not limited to, spoofing, replay, and man-in-the-middle types of attacks. Another great recurring advantage of blockchain decentralized identity verification with tamper-proof records and transparent audit logs. This is particularly important in situations like the healthcare sector, where secure and seamless access to patient records across multiple institutions is very critical.

However, we can also find most of the common constraints. Some of the solutions assume total privacy for some components like edge nodes, cloud servers, or hospital stations, which cannot be the case in a real-world setup for all those entities. Furthermore, while performance was evaluated via simulation or small-scale testbed, most of the systems lack proof of broad-scale, real-world deployment. Areas that need more research include scalability and interoperability and energy-efficient architecture.

In addition, many papers share common themes on an important issue. Users have not been given a focus on privacy and dynamic trust management. Authentication and performance are quite well addressed, but examples such as anonymous or pseudonymous identities, revocation mechanisms, and decentralized trust updates are quite poorly covered. Inclusion of such practices would strengthen the reliability and adaptability of blockchain-based distributed system authentication schemes. In summary, the research reviewed highlights significant progress in integrating blockchain with distributed authentication, offering practical and secure alternatives to traditional methods. However, to ensure real-world effectiveness, future efforts should focus on deployment scalability, trust decentralization, and comprehensive privacy preservation.

IV. Conclusion

The reviewed works show that blockchain can be of significant help in enhancing the authentication processes for distributed and IoT systems. They provide secure alternatives or faster processing times, as well as mutual authentication and tamper-proof logs instead of centralization. However, some have depended on trusted nodes or expected ideal cases, which hardly reflect real-life complexities, while many are still only in simulation or testbed settings. Further work in the direction of the feasibility study of practical implementations of future blockchain authentication would include tasks such as cross-platform interoperability and energy efficiency and scalability optimization.

References

[1] D. Puthal, S. P. Mohanty, P. Nanda, E. Kougianos, and G. Das, "Proof-of-Authentication for Scalable Blockchain in Resource-Constrained Distributed Systems," in Proceedings of the IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, Jan. 2019, pp. 1–5. doi: 10.1109/ICCE.2019.8662009.

[2] S. Guo, X. Hu, S. Guo, X. Qiu, and F. Qi, "Blockchain Meets Edge Computing: A Distributed and Trusted Authentication System," IEEE Transactions on Industrial Informatics, vol. 17, no. 11, pp. 7768–7778, Nov. 2021. doi: 10.1109/TII.2019.2938001. scholar.google.com.

[3] A. Yazdinejad, G. Srivastava, R. M. Parizi, A. Dehghantanha, K.-K. R. Choo, and M. Aledhari, "Decentralized Authentication of Distributed Patients in Hospital Networks Using Blockchain," IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 8, pp. 2146–2156, Aug. 2020. doi: 10.1109/JBHI.2020.2969648.

[4] S. S. Panda, D. Jena, B. K. Mohanta, S. Ramasubbareddy, M. Daneshmand, and A. H. Gandomi, "Authentication and Key Management in Distributed IoT Using Blockchain Technology,"IEEE Internet of Things Journal, vol. 8, no. 16, pp. 12947–12954, Aug. 2021. doi: 10.1109/JIOT.2021.3063806.

[5] K. Kim, J. Ryu, H. Lee, Y. Lee, and D. Won, "Distributed and Federated Authentication Schemes Based on Updatable Smart Contracts," Electronics, vol. 12, no. 5, pp. 1217, Mar. 2023. doi: 10.3390/electronics12051217.

[6] D. Puthal, R. Ranjan, A. Nanda, P. Nanda, P. P. Jayaraman, and A. Y. Zomaya, "Secure authentication and load balancing of distributed edge datacenters," Journal of Parallel and Distributed Computing, vol. 124, pp. 60–69, Feb. 2019. doi: 10.1016/j.jpdc.2018.10.007.

[7] Z. Tbatou, A. Asimi, C. El Balmany, Y. Asimi, and A. Guezzaz, "A Novel Architecture of a Strong and Mutual Authentication Protocol for Distributed Systems," Engineering Letters, vol. 28, no. 2, pp. 268–279, 2020.