International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

Run-Time Monitoring and Hash-Based Verification for Secure Client Machine States in Untrusted Cloud Infrastructures

Deekshitha Doosa
PG Scholar, Department of Computer Science and Engineering, Sree Dattha Institute of Engineering and Science, Sheriguda, Ibrahimpatnam, Hyderabad, Telangana, India – 501510.
Email: deekshithadoosa@gmail.com

Dr. SV Achuta Rao
Professor, Department of Computer Science and Engineering, Sree Dattha Institute of Engineering and Science, Sheriguda, Ibrahimpatnam, Hyderabad, Telangana, India – 501510.
Email: sreedatthaachyuth@gmail.com

***Abstract*— Cloud computing is increasingly adopted across critical sectors such as healthcare, banking, and social media due to its scalability, cost-effectiveness, and high-performance storage and computation capabilities. However, the delegation of data and computation to third-party cloud infrastructures introduces substantial security and privacy risks, including data manipulation and unauthorized access by internal or external entities. A Zero-Trust run-time framework is introduced to address these threats by continuously monitoring and verifying the client machine's state. This framework, termed Trusted Public Cloud (TPC), ensures integrity by integrating a kernel-level monitoring module, trust agent, Software Trusted Platform Module (SWTPM), and a verification cluster. It validates client-side changes using cryptographic hashing and real-time verification logic to detect anomalies and unauthorized modifications. The system supports dynamic verification policies using 'Include' and 'Exclude' strategies to optimize computation. Additionally, a log-based activity monitoring extension is incorporated to identify and block malicious users based on login behavior patterns. The framework is locally implemented to emulate cloud behavior and tested with simulated virtual machines, offering effective machine state tracking and real-time alerts against security breaches.**

Keywords*— *Zero-Trust, Cloud Security, Machine State Monitoring, TPM, Hash Verification, Intrusion Detection, User Activity Logging, Data Integrity.

I. INTRODUCTION

Cloud computing has revolutionized the digital landscape by enabling scalable, on-demand access to computational resources and storage without the need for significant upfront investments [1]. It has become a cornerstone for modern digital infrastructures, playing a crucial role across industries such as healthcare, finance, education, and enterprise IT management [2]. Major cloud service providers like Amazon Web Services and Google Cloud Platform offer geographically distributed data centers that support global operations efficiently [3]. However, the offloading of sensitive data and services to third-party-managed infrastructure introduces significant concerns around security, privacy, and trust. Users often relinquish direct control over their data, making it susceptible to unauthorized access, data breaches, and integrity violations [4].

Cloud environments are frequently targeted by a variety of cyber threats, including malware infections, data manipulation attacks, and malicious insider activity [5]. These attacks often go undetected for long durations, resulting in devastating financial losses and reputational harm to organizations. Traditional security paradigms based on perimeter defenses—where systems inside the network are inherently trusted—are no longer effective in the face of sophisticated, dynamic threats [6]. This "trust-but-verify" model fails when attackers gain access to internal systems, exploiting the absence of deeper runtime validations and internal threat detection.

In response, the Zero Trust Architecture (ZTA) has emerged as a modern security framework that operates on the principle of "never trust, always verify" [7]. It mandates strict user authentication, continuous monitoring, and real-time validation of both user behavior and system integrity. While existing ZTA models emphasize network-layer controls, they often lack visibility into runtime behaviors at the client endpoint or fail to detect insider compromises, including administrator misuse [8]. Addressing these gaps is essential for building a resilient, secure, and trustworthy cloud ecosystem.

II. RELATED WORK

R. Khandelwal et al. [9] proposed PriSec, a privacy settings enforcement controller that enhances user control over personal data. This system monitors and enforces compliance with user-defined privacy settings across applications, emphasizing usability and accountability. The proposed model dynamically adapts to privacy preferences and reveals gaps in existing privacy control systems.

C. Stergiou et al. [10] addressed the triple challenge of security, privacy, and efficiency in sustainable cloud computing, particularly focusing on IoT and big data. Their

work integrates lightweight encryption with efficient data management and sustainable energy utilization strategies. However, the paper noted that combining all three parameters often leads to trade-offs in performance and cost.

L. Zhao and D. Lie [11] questioned the traditional assumption that hardware-based security is superior to software. Their study critically evaluated various trusted hardware mechanisms and revealed vulnerabilities in technologies like Intel SGX. They concluded that hardware alone is insufficient and advocated for layered, collaborative approaches for critical infrastructure.

S. Ankergard et al. [12] examined software-based remote attestation for IoT environments. Their survey classified attestation mechanisms and identified challenges in trust anchor protection, runtime efficiency, and scalability. The study emphasized the need for robust software-based models, especially for resource-constrained IoT devices where hardware attestation is infeasible.

S. Ameer et al. [13] proposed ZTA-IoT, a novel Zero Trust Architecture tailored for IoT systems. Their model combines dynamic policy enforcement with usage control and continuous authentication. Unlike conventional models, ZTA-IoT accounts for device heterogeneity and volatile connectivity in IoT, ensuring higher resilience against internal threats and unauthorized access.

Y. Liu et al. [14] introduced a sharding blockchain-based framework for secure and scalable cross-domain data sharing in Zero Trust environments. This architecture distributes blockchain storage across cloud, edge, and endpoint layers. Their system supports decentralized verification, minimizing latency and overhead. The study demonstrates improved scalability while maintaining data privacy.

B. Ali et al. [15] implemented a dual fuzzy logic system for Zero Trust security in Multi-Access Edge Computing (MEC). Their model dynamically evaluates trust and allocates computational tasks accordingly. It combines behavioral analysis and contextual data to make access decisions, effectively preventing unauthorized access in high-traffic edge environments.

Lastly, Q. Shen and Y. Shen [16] proposed a collaborative endpoint reinforcement mechanism within a Zero Trust framework. Their approach integrates identity verification, behavior monitoring, and system integrity validation into one architecture. The model ensures that each endpoint complies with security policies continuously, not just during login sessions, addressing a key gap in existing ZTA implementations.

III. Materials And Methods

The proposed system introduces a Zero-Trust run-time framework [18] designed to ensure the continuous verification of client machine states in a cloud environment. It aims to detect unauthorized modifications and potential security breaches by monitoring file-level changes using cryptographic hash verification. The architecture includes four core components: a Kernel Module that captures file activity and generates hashes, a Trust Agent that manages hash queues and communicates with verification services [17], a Trusted Platform Module (TPM) or software-based equivalent for secure cryptographic operations, and a Verification Cluster that validates machine integrity using include/exclude verification policies. Additionally, a user activity logging mechanism is implemented to track suspicious login attempts and detect brute-force attacks. This log can be analyzed by an admin, who can block malicious users to prevent further damage. The system is designed to function on a local cloud server setup, mimicking real cloud infrastructure for effective emulation, monitoring, and enforcement of Zero-Trust principles.

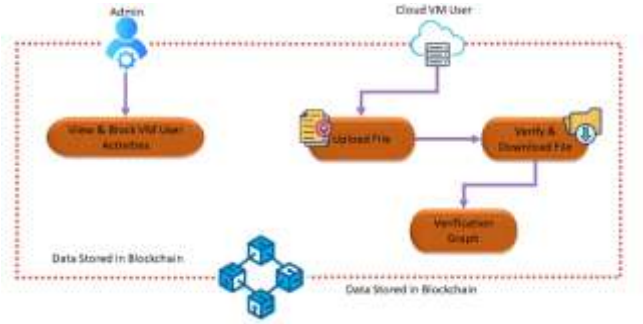


Fig. 1. System Architecture

The System Architecture (fig.1) shows a secure cloud setup where the Admin monitors user activity, and the Cloud VM User uploads, verifies, and downloads files. A verification graph visualizes the process. All actions are stored on the blockchain, ensuring transparency, data integrity, and secure file handling in the cloud.

A) Modules:

1) Admin Module

Admin Login: Allows administrator to securely access the system using credentials to manage and monitor user activities and overall system behaviour framework [19, 20].

View & Block VM User Activities: Enables admin to view user login logs, detect suspicious behavior, and block unauthorized users from accessing cloud services.

Logout (Admin): Safely terminates the admin session, preventing unauthorized access and ensuring secure exit from the administrative control panel.

2) User Module

User Login: Allows registered users to log into the system securely and access file-related services like upload, verification, and download.

Upload File: Enables users to outsource encrypted files with hash generation and select verification type—Include or Exclude—for integrity checking.

Verify & Download File: Performs file hash verification to detect tampering and allows users to decrypt and download verified files securely.

Verification Graph: Displays comparison of verification time between trusted and untrusted machines, helping users visualize security performance metrics.

Logout (User): Ends the user session securely, ensuring that no further actions can be performed without re-authentication.

b) Methods/Technologies:

1. File Hashing (Integrity Check): File hashing generates a unique fingerprint for each file using cryptographic algorithms. During verification, current hashes are compared with original ones to detect unauthorized modifications or tampering.

2. File Encryption (Confidentiality): Before uploading, files are encrypted using secure encryption methods. This protects file content from being accessed or read by unauthorized users, ensuring privacy even on untrusted cloud infrastructure.

3. Verification Types (Include/Exclude): Users assign each file a verification type—Include or Exclude. Only Include files undergo hash verification, helping reduce computational overhead while allowing selective integrity checks based on user preference.

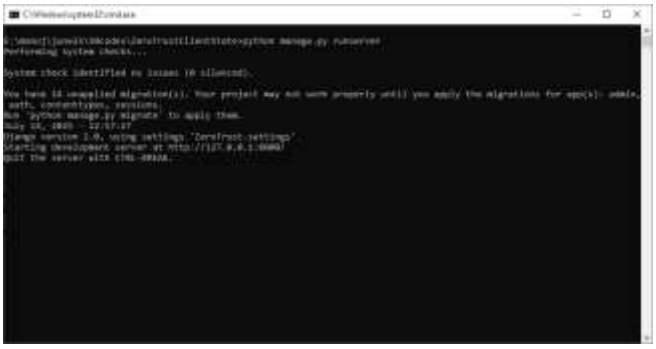
4. Kernel-Level Monitoring: The kernel module constantly monitors file access events. It captures and hashes files at runtime when accessed or modified, enabling real-time detection of unauthorized changes in system state.

5. TPM (Trusted Platform Module): Integration TPM securely stores encryption keys and measurement data. It ensures that only authenticated, tamper-free client systems can interact with the cloud, enhancing overall data and system security.

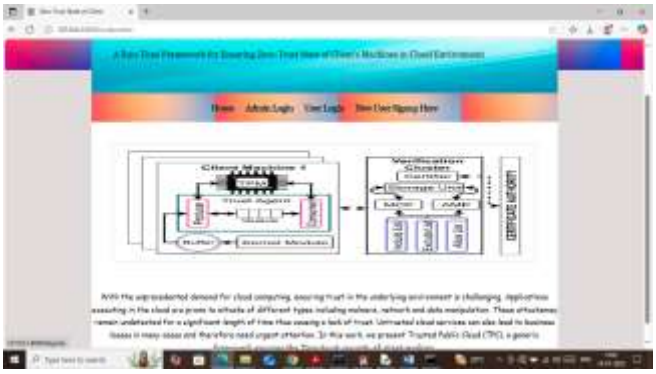
6. Activity Logging and User Behavior Monitoring: All user actions are recorded in logs, including login attempts and uploads. Admin reviews logs to detect suspicious patterns and block users involved in unauthorized or repeated failed logins.

IV. Experimental Results

To run project double click on 'run.bat' file to start python cloud web server and then will get below page

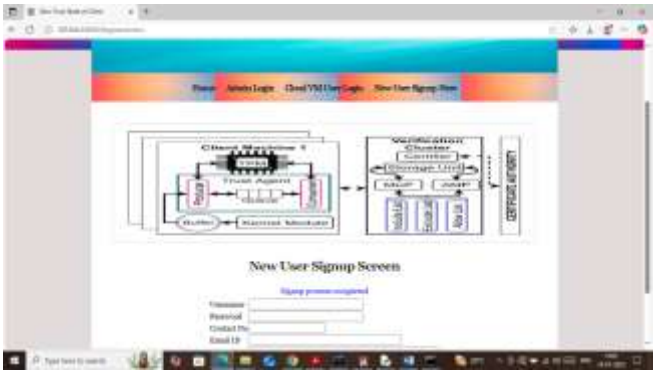


In above screen python server started and now open browser and enter URL as http://127.0.0.1:8000/index.html and then press enter key to get below page

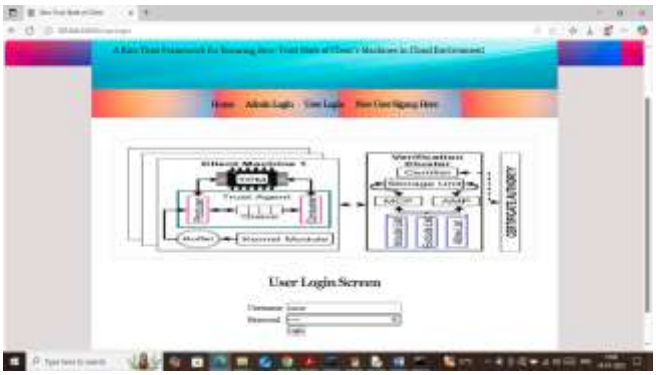


In above screen click on 'New User Sign up' link to get below page

In above screen user is entering sign up details and then press button to get below page

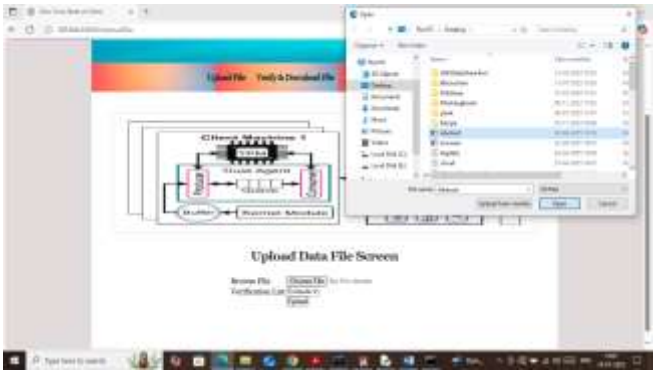


In above screen user sign up completed and similarly you can sign up as many users as you want. Now click on 'Cloud VM User Login' link to get below page



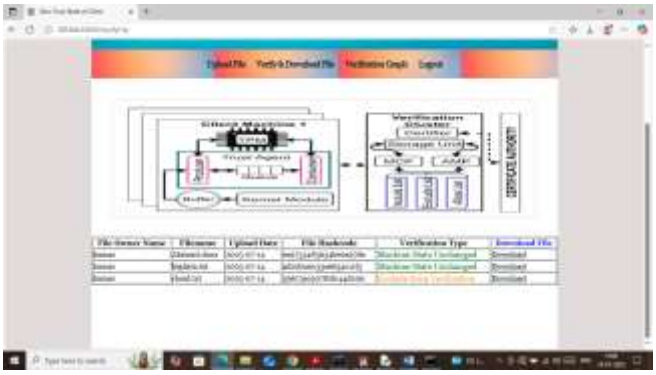
In above screen user is login and after login will get below page

In above screen user can click on 'Upload File' link to outsource file to client current state

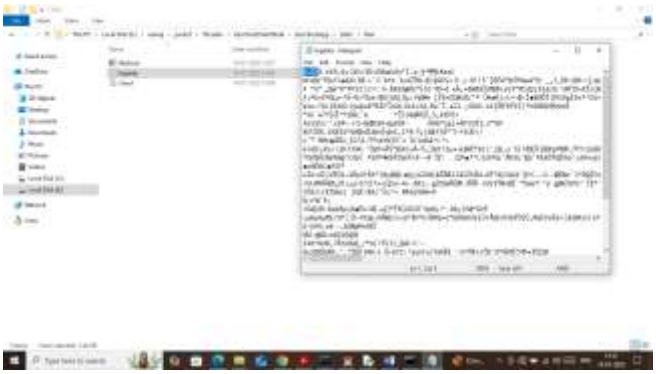


In above screen user can select desired file and then choose verification type as 'Exclude or Include' and then press button to get below page

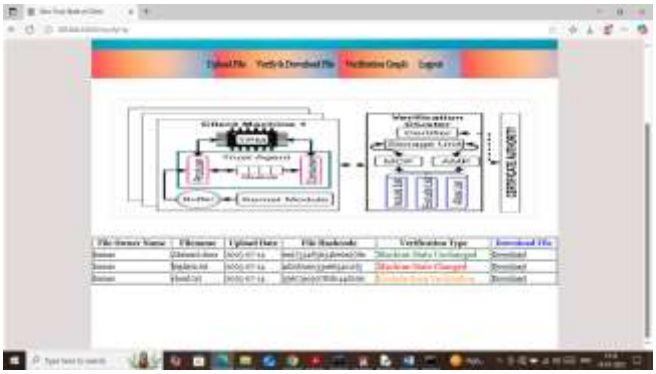
In above screen file encrypted and outsource to cloud and can see generated file hash for future file chain verification. Similarly by using above module you can upload any type and any number of files. Now click on 'Verify & Download File' link to get below page



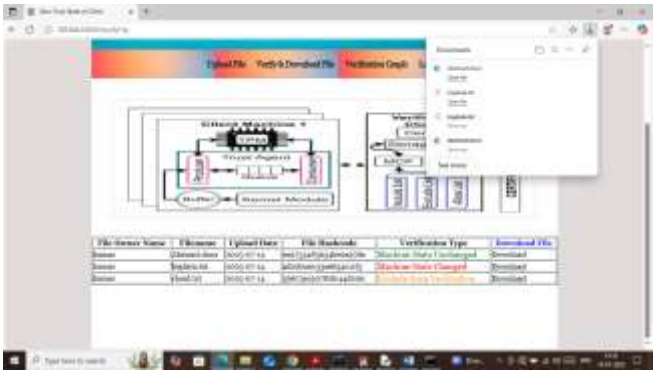
In above screen displaying list of uploaded files and can see all files are in unchanged state. Files with Include Verification type and with unchanged state will show in Green colour and File with Exclude will show in Orange colour. Files which tamper will show in Red colour. In above screen all files are unchanged and now manually will change file content from server memory by using below screen



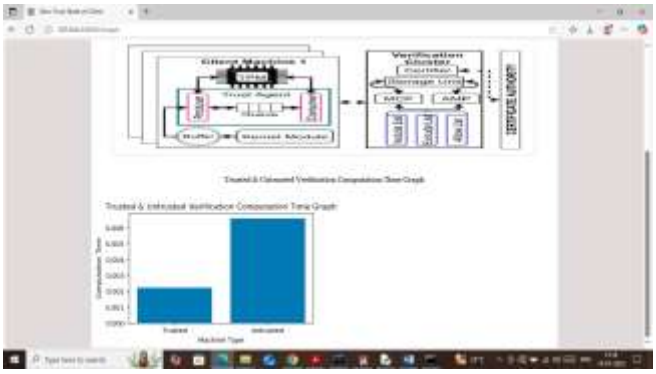
In above screen under 'ZeroTrustApp/static/files' folder you can see all uploaded files in Encrypted format and now change above file data by removing or adding new content and then saved the file. After saving just go back to application and perform verification



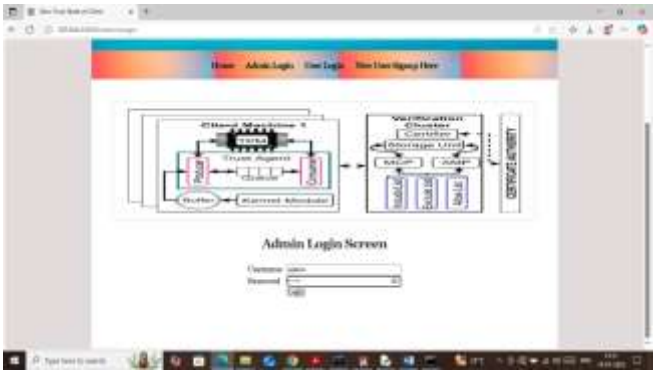
In above screen after manipulating file content we got 'Machine State Changed' verification status in red text. So by using above module we can detect weather files saved in cloud are changed or not. Now click on 'Download' link to download file and then will get below page



In above screen in browser status bar can see file is downloaded and decrypted and now click on 'Verification Graph' link to get below page

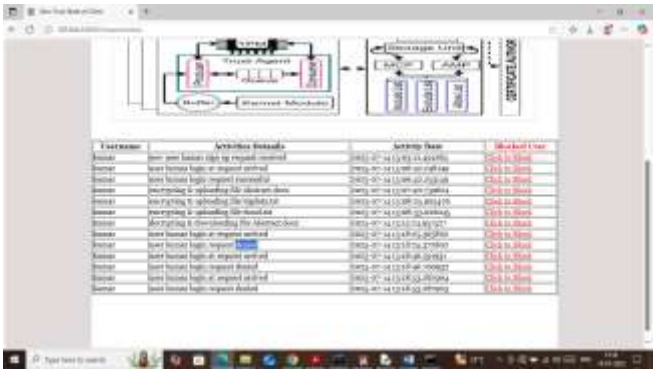


In above graph x-axis represents Machine Type and y-axis represents Verification time and can see untrusted machine taking more time to verify. Now logout and login as admin to analyse log



In above screen admin is login and after login will get below page

In above screen admin can click on 'View & Block VM User Activities' link to get below list of user activities



In above screen admin can view all activities from user and can see in above table many number of times application denied user login so admin can understand this user trying to get illegal access to cloud. Admin can carefully analyse and take decision of Blocking user by clicking on 'red colour link'. After block will get below page and user will not allow to login or access cloud in future

In Above Screen Can See Selected User Access Successfully Blocked.

V. Conclusion

This work establishes a functional Zero-Trust security framework capable of monitoring and validating the run-time state of client machines in a cloud environment. Through kernel-level monitoring, cryptographic hashing, and trusted platform modules, it ensures continuous verification of file integrity and rapid detection of unauthorized changes. The verification cluster efficiently compares file states, while activity logging strengthens internal security by tracking and analyzing user behavior. Admins can identify and block suspicious users, minimizing insider threats. Implemented on a local cloud simulation, the system effectively showcases real-time anomaly detection, secure file outsourcing, and strong defense mechanisms against potential internal and external cloud attacks.

Future improvements may include integration with real public cloud platforms, implementation of machine learning for advanced threat detection, and development of automated response mechanisms. Enhanced scalability, multi-cloud compatibility, and AI-driven user behavior analysis can further strengthen security and extend Zero-Trust adoption across large-scale enterprise environments.

References

[1] Rosén, A. (2025). Achieving Zero Trust: A Strategic Roadmap for Phased Implementation of Zero Trust Architecture.

[2] Scientific, L. L. (2025). SECURE AND RESILIENT VIDEO BLOCK MANAGEMENT AND PRINCIPLES FOR ZERO TRUST CLOUD NETWORK. Journal of Theoretical and Applied Information Technology, 103(7).

[3] Jena, B., Mishra, D., Mishra, S., & Mishra, S. (2025). Zero Trust Architecture-Based Enhanced Security Framework for OTT in Hybrid Cloud Environment. In Ethical Impacts of Using AI for Sustainable Development (pp. 89-112). IGI Global Scientific Publishing.

[4] Avirneni, S. T. (2025). Intent-Aware Authorization for Zero Trust CI/CD. arXiv preprint arXiv:2504.14777.

[5] Asaithambi, S., Nallusamy, S., Yang, J., Prajapat, S., Kumar, G., & Rathore, P. S. (2025). A secure and trustworthy blockchain-assisted edge computing architecture for industrial internet of things. Scientific Reports, 15(1), 15410.

[6] D. N. Jha, G. Lenton, J. Asker, D. Blundell, and D. Wallom, "Trusted platform module based privacy in public cloud: Challenges and future perspective," IEEE IT Professional, pp. 1–5, 2022.

[7] M. Ma, Z. Yin, S. Zhang, S. Wang, C. Zheng, X. Jiang, H. Hu, C. Luo, Y. Li, N. Qiu et al., "Diagnosing root causes of intermittent slow queries in cloud databases," Proceedings of the VLDB Endowment, vol. 13, no. 8, pp. 1176–1189, 2020.

[8] H. Tabrizchi and M. K. Rafsanjani, "A survey on security challenges in cloud computing: issues, threats, and solutions," The journal of supercomputing, vol. 76, no. 12, pp. 9493–9532, 2020.

[9] R. Khandelwal, T. Linden, H. Harkous, and K. Fawaz, "Prisec: A privacy settings enforcement controller," in 30th {USENIX} Security Symposium ({USENIX} Security 21), 2021.

[10] C. Stergiou, K. E. Psannis, B. B. Gupta, and Y. Ishibashi, "Security, privacy & efficiency of sustainable cloud computing for big data & iot," Sustainable Computing: Informatics and Systems, vol. 19, pp. 174– 184, 2018.

[11] L. Zhao and D. Lie, "Is hardware more secure than software?" IEEE Security & Privacy, vol. 18, no. 5, pp. 8–17, 2020.

[12] S. F. J. J. Ankergard, E. Dushku, and N. Dragoni, "State-of-the-art ˚ software-based remote attestation: Opportunities and open issues for internet of things," Sensors, vol. 21, no. 5, p. 1598, 2021.

[13] S. Ameer, L. Praharaj, R. Sandhu, S. Bhatt, and M. Gupta, "Zta-iot: A novel architecture for zero-trust in iot systems and an ensuing usage control model," ACM Transactions on Privacy and Security, 2024.

[14] Y. Liu, X. Xing, Z. Tong, X. Lin, J. Chen, Z. Guan, Q. Wu, and W. Susilo, "Secure and scalable cross-domain data sharing in zerotrust cloud-edge-end environment based on sharding blockchain," IEEE Transactions on Dependable and Secure Computing, 2023.

[15] B. Ali, M. A. Gregory, S. Li, and O. A. Dib, "Implementing zero trust security with dual fuzzy methodology for trust-aware authentication and task offloading in multi-access edge computing," Computer Networks, vol. 241, p. 110197, 2024.

[16] Q. Shen and Y. Shen, "Endpoint security reinforcement via integrated zero-trust systems: A collaborative approach," Computers & Security, vol. 136, p. 103537, 2024.

[17] R. Geng, C. Fang, S. Guo, D. Kang, B. Lyu, S. Zhu, and P. Cheng, "Flowpinpoint: Localizing anomalies in cloud-client services for cloud providers," IEEE Transactions on Cloud Computing, 2023.

[18] B. Shen, H. Yu, P. Hu, H. Cai, J. Guo, B. Xu, and L. Jiang, "A cloudedge collaboration framework for generating process digital twin," IEEE Transactions on Cloud Computing, 2024.

[19] R. Xin, P. Chen, P. Grosso, and Z. Zhao, "A fine-grained robust performance diagnosis framework for run-time cloud applications," Future Generation Computer Systems, 2024.

[20] A. Noor, D. N. Jha, K. Mitra, P. P. Jayaraman, A. Souza, R. Ranjan, and S. Dustdar, "A framework for monitoring microservice-oriented cloud applications in heterogeneous virtualization environments," in 2019 IEEE 12th international conference on cloud computing (CLOUD). IEEE, 2019, pp. 156–163.