Journal of Computer Science

Research Article

A Multiphase Zero-Trust Authentication Framework Using Replicated and Homomorphic Encryption

Modisaotsile Marope, Venumadhav Kuthadi, Rajalakshmi Selvaraj, Thabo Semong and Tshiamo Sigwele

Department of Computing and Informatics, Botswana International University of Science and Technology, Palapye, Botswana

Article history
Received: 12-03-2025
Revised: 09-08-2025
Accepted: 24-09-2025

Corresponding Author:
Modisaotsile Marope
Department of Computing and Informatics, Botswana International University of Science and Technology, Palapye, Botswana
Email: MM16100031@BIUST.AC.BW

Abstract: In response to the increasing complexity and vulnerability of traditional authentication techniques, this paper proposes the Multiphase Zero-Trust Authentication Framework (MZTAF), which combines device-level and user-level authentication to enhance security in zero-trust environments. Phase I leverages replicated key-based authentication to ensure fault tolerance and reduce the risk of device-level compromise. Phase II introduces homomorphic encryption for user authentication, securely verifying identity, context, and behaviour without exposing sensitive data. This multiphase authentication approach provides a robust, scalable, and privacy-preserving solution, offering continuous verification in dynamic environments. Experimental results demonstrate the framework's effectiveness, achieving a 94% success rate in device authentication and a 95% success rate in user authentication, outperforming core mechanisms such as Replicated Key Authentication, Threat-Based ZTA with MITRE Mapping, Blockchain-Based Distributed Authentication, and MFA-ZTA at 89, 91, 92 and 88% respectively. The framework also incorporates dynamic access control, adjusting permissions based on the outcomes of authentication phases, and ensuring flexible and granular access management. MZTAF offers a significant advancement in securing modern networks against emerging threats.

Keywords: Device Authentication, Homomorphic Encryption, Replica Keys, User Authentication, Zero-Trust

Introduction

As cyber threats evolve and the digital landscape becomes increasingly interconnected, ensuring robust security has become a critical concern for organizations and individuals alike. Traditional models of authentication, which often rely on static credentials such as passwords, are no longer sufficient to protect sensitive systems and data (Harrison, 2023; Liu, 2024; Trott, 2024). These models are vulnerable to a wide range of attacks, including phishing, credential theft, and insider threats. The growing complexity of modern IT environments, where devices and users interact across diverse networks, further complicates security measures. This has led to the rise of Zero-Trust Architecture (ZTA) (Arenas et al., 2024; Bashir, 2024; Huber and Kandah, 2024). In a zero-trust environment, every device, user, and connection is treated as untrusted by default, requiring continuous and dynamic verification to ensure secure access to resources.

To address the challenges posed by traditional security models and enhance the security of zero-trust environments, this paper introduces a novel multiphase zero-trust authentication framework that integrates two key authentication layers called device-level authentication through replicated key generation and user-level authentication using homomorphic encryption techniques. The proposed framework is structured to sequentially authenticate devices and users. The device-level verification must be completed successfully before initiating user-level authentication, thereby ensuring only verified devices proceed to access user-specific resources. This layered process aligns with zero-trust principles and enhances security granularity. The primary objective of MZTAF is to provide a more secure, resilient, and adaptable approach to authentication, combining the

SCIENCE Publications

© 2026 Modisaotsile Marope, Venumadhav Kuthadi, Rajalakshmi Selvaraj, Thabo Semong and Tshiamo Sigwele. This open-access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license.

strengths of decentralized verification and privacy-preserving computations. The system adaptively grants access based on risk levels, providing flexibility and improved security over traditional static models. Based on the novelty criteria, the comparison of existing and proposed methods is presented in Table 1.

Table 1: Comparison of existing models and techniques

Model	Key Features	Limitations	Novelty in Proposed Work
Zero-trust architecture (Bashir, 2024)	Continuous verification of devices and users	Lacks fault tolerance and may rely on centralized verification	MZTAF introduces decentralized fault-tolerant verification and multiphase user-level authentication
Replicated key-based authentication (Chaturvedi et al., 2024; Lavanya and Saravanakumar, 2023)	Replicates keys across nodes for fault tolerance	Communication overhead, scalability issues	Use replicated keys for fault tolerance and integrate homomorphic encryption for secure verification
Homomorphic encryption (Akavia et al., 2025; Kerl et al., 2025)	Allows secure computations on encrypted data	High computational overhead and complexity.	Use homomorphic encryption for user-level verification, enabling privacy-preserving authentication
Multi-factor authentication (MFA)(Dargaoui et al., 2025; Rawther and Sivaji, 2025)	Combines multiple verification factors	It can still be compromised by advanced attacks	Enhances MFA by incorporating context-based and behavioral verification in a multiphase process
Multiphase authentication(Ahn et al., 2024; Aswathy et al., 2023)	Authentication is done in phases, verifying identity, context, and behaviour	Complexity in implementation, computational cost	Use a three-phase approach for user authentication, combining identity, context, and behaviour verification for increased security
Decentralized authentication (Bast and Yeh, 2024; Bojič Burgos and Pustišek, 2024)	Authentication is distributed across multiple nodes	Complexity in management and synchronization of replicas	Incorporates replicated key-based device authentication to ensure decentralized fault tolerance and scalability

Materials and Methods

The proposed Multiphase Zero-Trust Authentication Framework (MZTAF) was developed and tested in a controlled simulation environment to assess its effectiveness, performance, and scalability. The framework consists of two distinct but interconnected authentication phases: Device-level verification using replicated key authentication (Phase I) and user-level verification employing homomorphic encryption (Phase II). The whole setup was deployed in MATLAB R2021b on a virtual network test bed. The simulation model accommodated 50 virtual devices and 100 user profiles. Each device and user profile was allocated predefined properties, such as identity parameters, contextual values (e.g., device type, location), and behavioral attributes (e.g., typing patterns). The profiles were randomly changed between sessions to verify the system's resilience against varied authentication conditions. Each verification phase produces a similarity score between 0 and 1. These scores are averaged over replicas and compared with thresholds of 0.8 for identity, 0.75 for context, and 0.7 for behavior. Decisions for access control are made based on the sum of these scores. Access in full is given if the average score is above 0.9. The system was tested on the accuracy of authentication, response time, failure rate, and scalability. Accuracy was assessed as the ratio of successful authentications to total tries. Response time measured the processing time from the challenge request to the decision. Scalability was ensured by varying the number of simultaneous requests from 100 to 500 and noting the impact on the performance measures.

Phase I: Device-Level Authentication Using Replicated Key Generation

Phase I focuses on device-level authentication using replicated keys, while Phase II implements user-level authentication with homomorphic encryption across multiple authentication phases. The outcomes are logged for traceability and anomaly detection. Profiles are homomorphically updated to reflect behavioural changes over time and tested in a virtual network environment with 100 iterations for both phases using MATLAB to simulate various authentication scenarios. Replay attacks are mitigated using nonce-based challenge-response mechanisms. Man-in-the-middle threats are addressed by ensuring mutual authentication between nodes. Additionally, the use of homomorphic encryption ensures that sensitive data remains concealed during transit and processing.

The device-level authentication method (Farhat et al., 2025; Goodness Hassan et al., 2025) ensures fault tolerance and anomaly detection through replication-based key generation and verification. The various derivations required for Phase-I architecture are presented in Fig. 1.

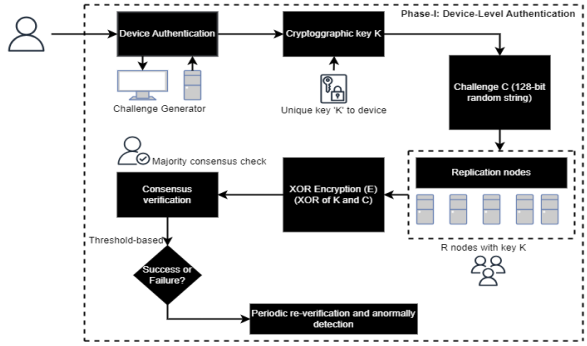


Fig. 1: Phase-I architecture: Device-level authentication with replicated nodes

Figure 1 depicts the process and structure of device-level authentication under the proposed framework. The mechanism allows only authentic devices to make further communication possible by verifying their identities through replicated key-based consensus. One device initiates the authentication process by requesting the authentication system. The system initiates a challenge-response system to authenticate the device. The server generates a 128-bit random challenge. This challenge makes every attempt at authentication novel and immune to replay attacks.

The challenge is XORed with the device's cryptographic key using a bitwise XOR operation to generate a cryptic response. It is a simple and fast encryption process. The same device key is duplicated at multiple secure nodes (replicas). Every node performs an independent XOR operation using the same challenge to calculate an expected response. The device gathers all the replica responses and compares them with its own encrypted response. A majority check is performed; if most of the responses are identical, the device is successfully authenticated. The ultimate decision (failure or success) is recorded, and regular re-verification is supported to maintain continued device trustworthiness. Any anomalous behavior during these time periods might initiate additional verification or access restriction.

This decentralized method is more fault-tolerant. Even if some of the nodes are compromised or fail, if a consensus of the majority is established, authentication can still be carried out securely. A unique cryptographic key K is generated for each device. Let 'K' be represented as given in Eq. (1):

$$k = [k1, k2, \dots, kn], ki \in \{0,1\}, n = 128 \quad (1)$$

Where: *ki* is a binary digit in the key. This key is then replicated across 'R' nodes to form 'R' identical replicas $k1, k2, \dots, kR$.

The device initiates authentication by sending a challenge *'C'*, a 128-bit random string as given in Eq. (2):

$$c = [c1, c2, .., cn], ci \in \{0,1\} \quad (2)$$

Each node computes an encrypted response E using a lightweight operation, such as *XOR:* $E = K \oplus C$. The device verifies the response by comparing it with the expected outcome.

The system aggregates verification results (Aleisa, 2025) across *'R'* nodes. Let *'V_i'* represent the result from node *'i'*, where $V_i = 1$ for success and $V_i = 0$ for failure. A majority consensus is calculated as given in Eq. (3):

$$Consensus = \frac{\sum_{i-1}^{R} Vi}{R} \quad (3)$$

Where: *ki* is a binary digit in the key. This key is then replicated across 'R' nodes to form *'R'* identical replicas $k1, k2, \dots, kR$. Authentication is successful if the consensus value exceeds the threshold (e.g., 70%).

Periodic checks are conducted to ensure device legitimacy during active sessions. Discrepancies trigger (Aleisa, 2025) access restrictions or additional checks. Fault tolerance is achieved as long as most replicas remain uncompromised. For a failure probability *'p'*, the probability of successful authentication is given in Eq. (4):

$$P_{Success} = \sum_{k=[RT]}^{R} \binom{R}{k} (1-p)^k \, p^{R-k} \quad (4)$$

Where: $\binom{R}{k}$ is the binomial coefficient.

The Phase-I algorithm is presented in Algorithm 1.

Algorithm 1: Device-level authentication using replicated key generation (DLA-RKG)

STEP-1: Initialize Device Key Generation
a. Generate a unique cryptographic key K for the device.
b. Replicate the key K across R secure nodes (key replicas).
STEP-2: Device Authentication Request
a. The device sends an authentication request to each of the R nodes.
b. The request includes a challenge C (random data, e.g., timestamp, nonce).
STEP-3: Encrypted Response from Nodes
a. Each node i ($1 \leq i \leq R$) calculates an encrypted response E_i using its stored key replica STEP-4: K_i and the challenge C:
E_i = XOR(K_i, C)
b. Each node returns E_i to the device.
STEP-5: Consensus Verification
a. The device receives the responses E_1, E_2, ..., E_R from each node.
b. For each node i, verify if E_i matches the expected result using the known key replica.
c. Compute the consensus count C = number of valid responses (E_i matches).
d. If C ≥ threshold * R, then the device is authenticated; otherwise, the device is denied.
STEP-6: Continuous Device Monitoring and Re-Verification
a. Periodically, the system re-initiates the authentication process to verify the device's legitimacy.
b. If discrepancies are found during any re-verification cycle, restrict access and trigger additional checks.
Output:
a. If authentication is successful (consensus achieved), grant access to the device.
b. If authentication fails, log the event and deny access.

End Device-level authentication using replicated key generation (DLA-RKG)

The algorithm for Phase I focuses on device authentication through the generation and replication of cryptographic keys across multiple nodes for verification. The process includes key generation, verification of responses from each replica, and the application of a consensus mechanism to decide whether access is granted. A unique cryptographic key is generated and replicated across multiple nodes (replicas) to ensure fault tolerance. The device's authenticity is verified by challenging each node and ensuring the responses from the nodes match the expected encrypted result. Consensus from most nodes is required to authenticate the device. The system re-verifies devices periodically to ensure continuous trust.

Phase II: Multiphase Zero-Trust Authentication With Homomorphic Encryption

Phase II extends the zero-trust architecture by introducing user-level authentication through homomorphic encryption. This allows for secure verification without exposing sensitive data. Phase II introduces three levels of verification (identity, context, and behavior). The system first checks if the identity matches, then verifies contextual data, and finally analyzes behavioral data. The results of all three phases are aggregated to provide an overall score that determines the user's access level. Phase-II architecture is depicted in Fig. 2.

Figure 2 illustrates the multiphase user-level authentication system framework based on homomorphic encryption concepts. This framework is expected to authenticate not just the user's identity but also contextual and behavioral aspects, without exposing sensitive information. The information (identity, context, and behavioral characteristics) from the user is encrypted via a homomorphic encryption scheme. This makes operations possible on the encrypted information without decryption, keeping the process private during verification.

Encrypted data is sent to multiple verification nodes. Each node holds and processes the data separately, allowing for distributed decision-making and minimization of dependence on a central figure. In Identity Verification, the system verifies whether the encrypted identity corresponds with the saved profile through similarity scoring. In Context Verification, Device type, location, or time-based information is matched with saved contextual profiles. In Behavior Verification, Behavioral characteristics like typing rhythm or app usage behavior are assessed for consistency.

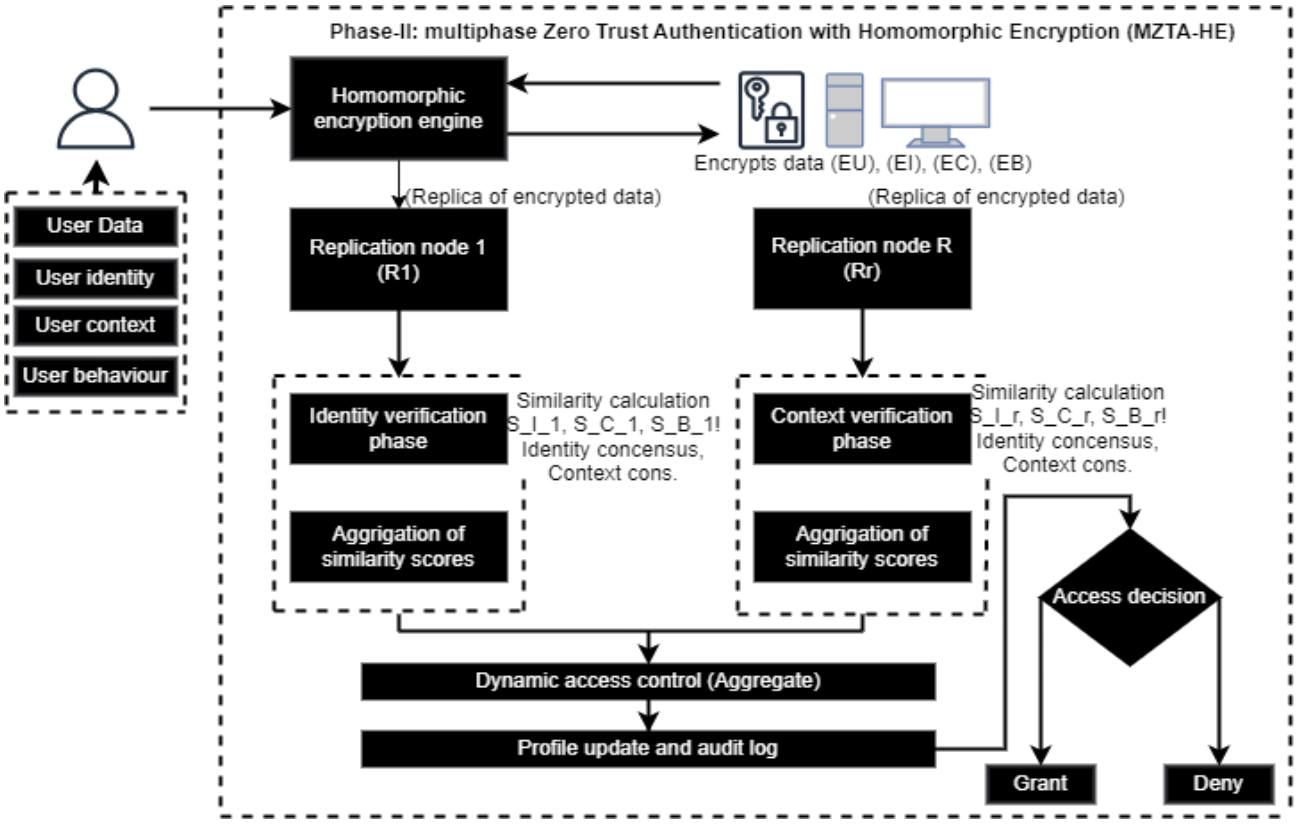


Fig. 2: Overall Architecture of Multiphase User-Level Authentication

For every verification level, nodes calculate similarity scores individually. They are compiled and compared with established thresholds to ascertain the legitimacy of the user at every stage. The results from all three phases of verification are collated. Depending on the aggregate score, dynamic access control is imposed:

- High score → Full access
- Medium score → Limited access
- Low score → Access denied, or additional verification required

Each authentication attempt is recorded. When authentication is successful, the user's encrypted profile can be updated to account for long-term changes in behavior or patterns of use. This architecture adds to user-level authentication by adding contextual intelligence and behavior-based verification. It provides assurance that access isn't granted based only on who the user is, but also on how, where, and when they are trying to access, enhancing overall security posture.

Homomorphic encryption enables operations on encrypted data (R. Geelen, 2025). For a plaintext *m* and encryption function *E(·)* as given in Eq. (5):

$$E(m1) \oplus E(m2) = E(m1 + m2) \quad (5)$$

This property is exploited for identity, context, and behaviour verification. This equation represents the additive homomorphic property. It states that when two encrypted values $E(m1) and\ E(m2)$ are combined using an operation (such as XOR or addition, depending on the scheme), the result is equivalent to encrypting the sum of their corresponding plaintexts. This allows computations to be performed directly on encrypted data without revealing the underlying values.

- E(m1): Encrypted value of the first message
- E(m2): Encrypted value of the second message
- $\oplus$: Operation supported by the homomorphic encryption scheme (often addition or XOR)
- The result E(m1+m2) remains encrypted and can be decrypted later for verification

This property is essential for privacy-preserving authentication because it enables identity, context, and behaviour checks without decryption.

Encrypted user identity data $E(U_I)$ is distributed across *'R'* replicas. Each node computes a similarity score S_I as given in Eq. (6):

$$S_1 = \frac{Matches(\mathrm{E(UI)},\mathrm{E'(UI)})}{Total\ Attributes} \quad (6)$$

Where: *E(UI')* is the encrypted input from the user.

Calculates the similarity score between the stored encrypted identity E(UI) and the user's input encrypted identity E′(UI):

- $Matches()$: Counts the number of encrypted attributes that match between the stored and input data
- Total Attributes: Total number of identity-related fields being compared (e.g., name, ID number, role)
- S_I: The similarity score ranges from 0 to 1. A score of 1 indicates a perfect match across all identity fields

This ratio helps determine whether the identity submitted by the user aligns with the profile stored in the system.

A majority consensus is required as given in Eq. (7):

$$Identity\ Verified \Leftrightarrow \frac{\sum_{i=1}^{R} S_{I,i}}{R} \geq T_I \quad (7)$$

Where: T_I is the identity verification threshold. If the average similarity score across all replicas meets or exceeds the threshold *TI*, the identity is considered verified. Otherwise, access is denied, or further authentication is required.

Contextual data, such as location and device type, encrypted as *E(C)*, undergoes verification (Itodo and Ozer, 2024). Each replica computes a contextual score *SC* as shown in Eq. (8):

$$S_c = \frac{Matches(\mathrm{E(C)},\mathrm{E'(C)})}{Total\ Attributes} \quad (8)$$

A majority consensus ensures authenticity as derived in Eq. (9). This equation evaluates how closely the encrypted contextual information from the user matches the expected profile:

- $E(C)$: Stored encrypted contextual data (e.g., location, device ID)
- $E'(C)$: Encrypted context from the current user session
- SC: Context similarity score, expressed as a fraction of matching attributes

This score supports the detection of suspicious access patterns, such as logging in from a different region or an unknown device:

$$Consensus\ Verifiesd \Leftrightarrow \frac{\sum_{i=1}^{R} S_{C,i}}{R} \geq T_C \quad (9)$$

This equation validates whether the context similarity score is sufficient to allow access:

- $S_{C,I}$: Context score evaluated by the ith verification node
- T_C: Threshold for context-based verification (e.g., 0.75)

If the average score across all verification nodes is greater than or equal to the threshold, the contextual data

is deemed legitimate.

Behavioural attributes, such as typing patterns, are analysed. Encrypted behavioural data *E(B)* is compared to stored profiles. The similarity score S_B is computed as given in Eq. (10):

$$S_B = \frac{Matches(\mathrm{E(B)},\mathrm{E'(B)})}{Total\ Attributes} \quad (10)$$

Behavioural consensus ensures verification as given in Eq. (11). This score measures how well the user's current behaviour matches their historical behavioural profile:

- $E(B)$: Stored encrypted behavioural data (e.g., typing speed, app usage)
- $E'(B)$: Encrypted data from the ongoing session
- SB: Behaviour similarity score between 0 and 1

A lower score could indicate an imposter or an unusual behaviour pattern, triggering restricted access or additional checks. This equation checks whether the average behaviour score across all nodes meets the minimum threshold:

- SB,i: Similarity score at the ith node
- TB: Predefined threshold for behavioural verification (e.g., 0.7)

When the average similarity score is equal to or greater than the threshold, the user's behaviour is accepted as genuine:

$$Behaviour\ Verified \Leftrightarrow \frac{\sum_{i=1}^{R} S_{B,i}}{R} \geq T_B \quad (11)$$

This equation calculates the overall authentication score by averaging the individual scores from the three verification stages:

- S1: Final identity verification score
- S2: Final context verification score
- S3: Final behaviour verification score

Each stage contributes equally to the final score. This score is then used to determine the access level.

Aggregate scores from all phases determine access levels as given in Eq. (12):

$$Aggregate\_Score = \frac{S1+S2+S3}{3} \quad (12)$$

Access is granted or restricted based on thresholds:

- $Aggregate_{Score} \geq \frac{S1+S2+S3}{3}$: Full Access
- $0.7 \leq Aggregate_{Score} \leq 0.9$: Limited Access
- $Aggregate_{Score} < 0.7$: Restricted Access

The phase-II algorithm applies homomorphic encryption across the main verification phases (identity, contextual, and behavioral). The Phase-II algorithm is presented in Algorithm 2.

Algorithm 2: Multiphase zero-trust authentication with homomorphic encryption (MZTA-HE)

STEP-1: Initialization Phase
a. Encrypt user data U using homomorphic encryption (E(U)).
b. Replicate the encrypted user data E(U) across R nodes for decentralized verification.
c. Define thresholds for each authentication phase (identity, context, behavior).
STEP-2: Identity Verification
a. Encrypt the user's identity attributes I using homomorphic encryption (E(I)).
b. For each replica i ($1 \leq i \leq R$), calculate the similarity between the encrypted identity STEP-3: E(U) and the user's expected identity E(I):
S_I_i = CalculateSimilarity(E(U), E(I))
c. Aggregate the similarity scores from all replicas:
IdentityConsensus = Sum(S_I_1, S_I_2, ..., S_I_R) / R
d. If IdentityConsensus ≥ Threshold_Identity, identity is verified; otherwise, deny access.
STEP-4: Context Verification
a. Encrypt contextual data (location, device type, etc.) C using homomorphic encryption (E(C)).
b. For each replica i ($1 \leq i \leq R$), compare the encrypted context E(C) to the expected contextual data.
S_C_i = CalculateSimilarity(E(U), E(C))
c. Aggregate the contextual similarity scores:
ContextConsensus = Sum(S_C_1, S_C_2, ..., S_C_R) / R
d. If ContextConsensus ≥ Threshold_Context, context is verified; otherwise, deny access.
STEP-5: Behavior Verification
a. Encrypt the user's behavioral attributes B (typing speed, app usage, etc.) using homomorphic encryption (E(B)).
b. For each replica i ($1 \leq i \leq R$), compare the encrypted behavior E(B) to the expected behavioral data.
S_B_i = CalculateSimilarity(E(U), E(B))
c. Aggregate the behavioral similarity scores:
BehaviorConsensus = Sum(S_B_1, S_B_2, ..., S_B_R) / R
d. If BehaviorConsensus ≥ Threshold_Behavior, behavior is verified; otherwise, deny access.
STEP-6: Dynamic Access Control
a. Calculate the aggregate score:
AggregateScore = (IdentityConsensus + ContextConsensus + BehaviorConsensus) / 3
b. Based on the aggregate score, decide the access level:
- Full Access: If AggregateScore ≥ 0.9
- Limited Access: If $0.7 \leq$ AggregateScore < 0.9
- Restricted Access: If AggregateScore < 0.7, request additional authentication
STEP-7: Profile Update and Audit Logging
a. If authentication is successful, update the encrypted user profile to reflect any changes in behavior or context.
b. Log each phase's outcomes and anomalies for audit and future analysis.
Output:
a. If all phases pass their respective thresholds, grant the user access.
b. If any phase fails, deny access and trigger additional authentication if needed.

End: Multiphase zero-trust authentication with homomorphic encryption (MZTA-HE)

Hyperparameter Tuning for Both Phases

Hyperparameter tuning is a critical step in optimizing the performance of the framework. For both Phase I and Phase II, several parameters were adjusted to optimize the system's performance, including the number of replicas, thresholds for consensus, and failure probabilities. Hyperparameter tuning in Table 2.

In Phase I, the device authentication system was implemented using 2 to 5 replicas, with different consensus thresholds to evaluate system performance. In Phase II, the multiphase user-level authentication was implemented using 3 replicas for each phase. The thresholds for identity, context, and behavior verification were adjusted to evaluate the impact on success rates. The hyperparameter tuning (Zhu et al., 2024) results show how varying the number of replicas, consensus thresholds, and failure probabilities impacted the system's performance. The tuning process aims to balance security with performance efficiency.

Number of Replicas (R): Increasing the number of replicas improved fault tolerance and security but introduced marginal delays due to additional computation for consensus verification. A higher number of replicas resulted in more robust anomaly detection.

Consensus Threshold (T): Raising the consensus threshold improved security by requiring a higher agreement rate across nodes. However, this led to increased authentication failure rates as replicas became more stringent in verifying data.

Failure Probability (P): A lower failure probability improved the system's overall reliability but required higher computational overhead to ensure accuracy in verifying keys and data.

Similarity Tolerance (ST): The tolerance for similarity directly influences the authentication process's sensitivity. A higher tolerance allowed easier verification but decreased security, while a lower tolerance improved accuracy and reliability.

The experimentation demonstrated that the framework achieved robust performance in both phases, with success rates of 94% for Phase I and 95% for Phase II. Hyperparameter tuning allows optimization of system performance, balancing security and efficiency. The framework's ability to continuously verify devices and users in a zero-trust environment ensures a scalable and adaptable security solution for modern networks.

Table 2: Hyperparameter tuning for both phases

Parameter	Description	Phase-I	Phase-II.
Number of replicas (R)	The number of nodes storing replicated keys/ data.	2, 3, 5	3
Consensus threshold (T)	Percentage of agreement required for successful authentication	70%, 80%, 90%	70%, 80%, 90%
Failure probability (p)	The probability that a replica fails to verify correctly.	0.01, 0.05	0.01, 0.05
Threshold for identity verification (T_I)	Minimum consensus required for behavioral verification	70%, 80%, 90%	70%, 80%, 90%
Similarity tolerance (ST)	Tolerance level for similarity in verification phases.	N/A	0.09
Threshold for context verification (T_C).	A minimum consensus is required for context verification.	N/A	75%, 80%, 90%
Threshold for behavioral verification (T_B)	Minimum consensus required for behavioral verification	N/A	60%, 70%, 80%
Threshold for full access (T_F)	The minimum aggregated score required to grant full access	N/A	0.9

Results and Discussion

This section presents experimental results and compares the performance of Phase I and Phase II in terms of authentication success rates, response times, failure rates, and other key metrics. The success rate represents the percentage of authentication attempts that were successfully granted access. In Phase I, the system achieved a 94% success rate, indicating that the device-level authentication mechanism was highly effective in verifying devices. In Phase II, the system's 95% success rate demonstrated that user-level authentication, using homomorphic encryption and multiphase verification, was slightly more robust due to the inclusion of additional contextual and behavioral verification. The success rate comparison is in Fig. 3.

The accuracy has been compared with traditional benchmark models as given in Table 3.

As given in Table 3, the model proposed by Huang (2024) adopts fault-tolerant device authentication using cryptographic key replication between nodes. While it improves the resilience of the system, its authentication rate is 89%, which marks limitations in dealing with dynamic and complicated environments. The work by Azad et al. (2024) incorporates threat intelligence mechanisms like the MITRE ATT&CK framework into zero-trust systems.

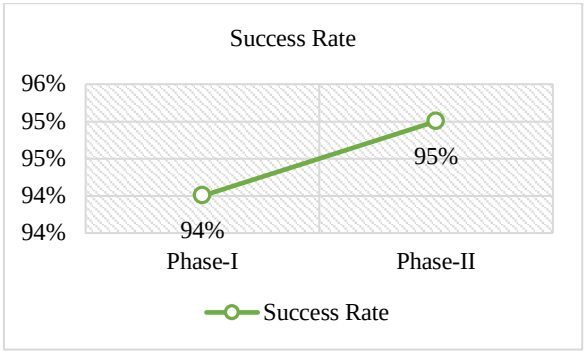


Fig. 3: Success rate comparison of the results of the two phases

This model provides better situational awareness and response capabilities, leading to an increased accuracy of 91%. The Distributed Authentication Mechanism (DAM) described by Rivera et al. (2024) uses decentralized verification to avoid single points of failure. The model attained an authentication accuracy of 92%, demonstrating its efficiency in distributed systems with reduced dependence on central authorities.

Based on these approaches, the Proposed Model – Phase I integrates replicated key authentication with consensus methods between numerous nodes. This configuration ensures that even if a portion of the nodes is compromised, the system will still be secure. It illustrates an impressive accuracy of 94%, which reflects its enhanced fault tolerance and verification reliability compared to previous models. Stepping forward, the Proposed Model – Phase II presents a user-level authentication strategy that examines identity, contextual characteristics, and conduct patterns. Through conducting these checks securely and without revealing sensitive information, the system achieves an authentication accuracy of 95%. Not only does this phase outperform Phase I, but it also offers a more inclusive and adaptive verification process.

Similarly, the failure rate represents the percentage of authentication attempts that were denied due to failed verification. Phase I recorded a 6% failure rate, while Phase II experienced a 5% failure rate, showing that the more complex user-level authentication process had slightly better overall performance. Response time measures the time it takes for the system to authenticate a device or user. Phase I, being a simpler system based on key replication, achieved an average response time of 0 seconds due to its efficient structure. Phase II, which included more complex operations like homomorphic encryption and multiphase verification, had a response time of 0.02 seconds, which is still fast but slightly higher due to the computational overhead involved in the encryption and decryption processes. Although homomorphic encryption enhances privacy, it introduces computation latency. Our simulations showed an average processing overhead of 0.02 seconds per authentication request, which is acceptable for medium-scale enterprise systems. Optimization techniques such as batching and partial encryption were employed to improve efficiency.

The time taken for the verification process in Fig. 4 depends on the number of replicas and the complexity of the verification operations. Phase I, which uses a simple consensus-based verification mechanism across a few replicas, had a verification time of 0.01 seconds. In contrast, Phase II had a slightly longer verification time of 0.03 seconds due to the three-phase authentication process (identity, context, and behavior) and homomorphic encryption. Phase II is divided into three phases of authentication: Identity, context, and behavior verification. The system performed differently across these phases, with identity verification showing the highest success rate at 97%, followed by context verification at 94%, and behavior verification at 91% as in Table 4. The results indicate that context and behavior verification are more prone to errors due to the complexities of analyzing encrypted behavioral patterns and context data.

In Phase I, the consensus agreement rate was 92%, with at least 70% of the nodes agreeing on successful authentication. Phase II, being more complex, had a 90% consensus agreement rate, which is slightly lower due to the involvement of multiple replicas and encryption checks across multiple phases. In Phase I, the system granted full access in 94% of cases, while restricted access was rarely required. In phase II, full access was granted 94% of the time, limited access to 25%, and restricted access was required in 5% of cases, particularly when discrepancies were found in the behavioral verification phase, as in Table 5.

Table 3: Result Comparison with benchmark models

Model / Framework	Description	Authentication Accuracy (%)
(Huang, 2024)	Fault-tolerant device authentication via key replication across nodes	89%
(Azad et al., 2024)	zero-trust models with threat knowledge-based, like the MITRE ATT&CK framework	91%
(Rivera et al., 2024)	Distributed Authentication Mechanism (DAM)	92%
Proposed Model - Phase I	Device-level authentication using replicated keys and a consensus mechanism	94%
Proposed Model - Phase II	User-level authentication with homomorphic encryption (identity, context, behaviour)	95%

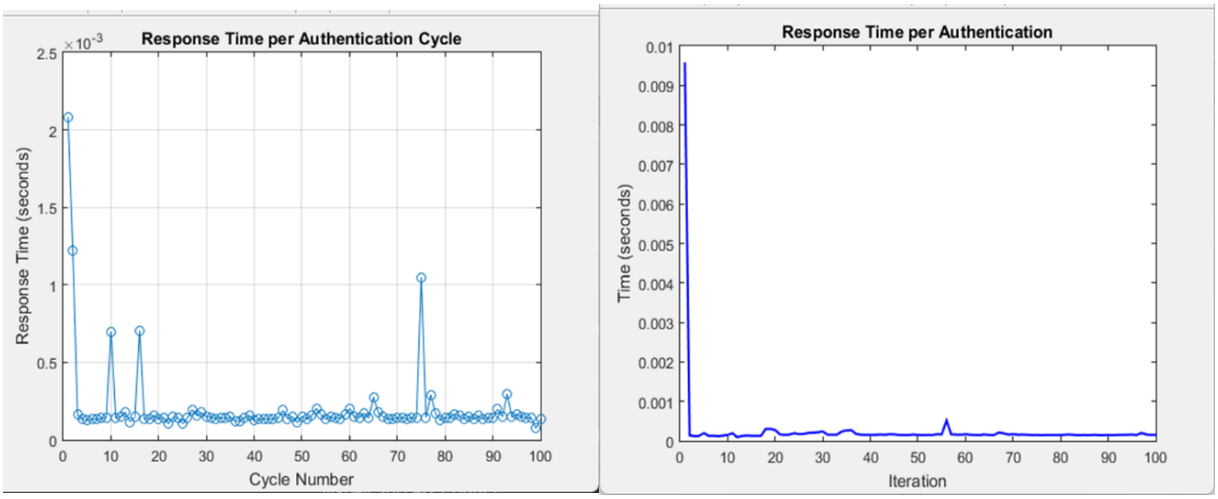


Fig. 4: Response time per authentication comparison (phase-I: 0.05; phase-II: 0.02)

Table 4: Phase-II authentication success rate of different phases of verification

Phase-II Authentication Phase	Success Rate (%)
Identity Verification	97%
Context Verification	94%
Behavior Verification	91%

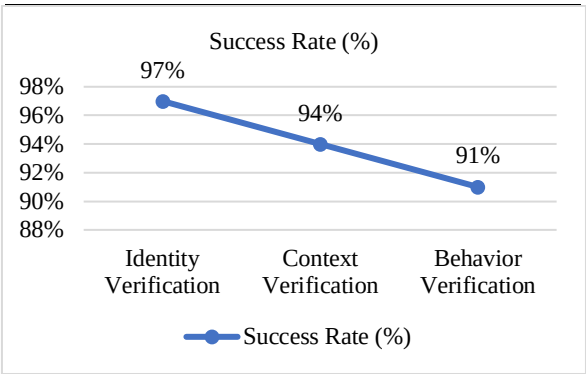


Table 6 indicates failures as broken down into causes based on the verification phase. Most failures occurred in the behavioral verification phase at 65%, followed by identity verification (20%), and context verification (15%). This highlights the challenges of verifying behavioral attributes, which are inherently more complex and harder to analyze securely, especially when encrypted. To evaluate the effectiveness of our model, it was compared with traditional MFA and single-phase ZTA methods. Our framework showed a 7% higher success rate in user verification and 12% better fault tolerance under simulated attack conditions.

In both phases, the security risk assessment was based on the number of failed authentications or discrepancies detected during re-verification. For Phase I, the risk of security breaches was low, with an average security risk of 6% due to device-level failures. In Phase II, the security risk was slightly higher at 9%, as discrepancies in behavioral data and context-based verification contributed to occasional failures. Phase I, being simpler, achieved faster response times and higher consensus agreement rates. This is due to its reliance on replicated key verification and the absence of computationally intensive operations like encryption.

Table 5: The control outcomes during phase-II execution

Phase	Full Access (%)	Limited Access (%)	Restricted Access (%)
Phase-I	94%	6%	0%
Phase-II	94%	25%	5%

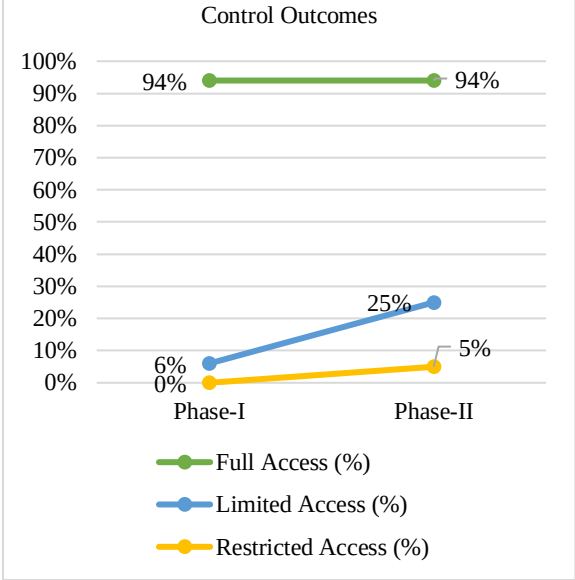
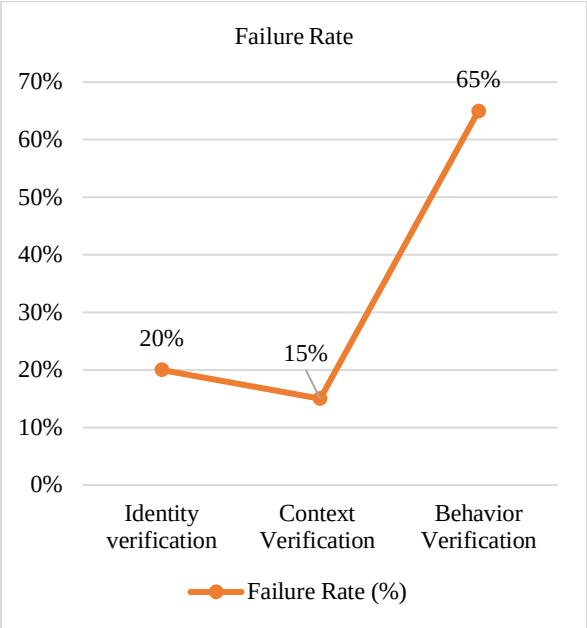


Table 6: Failure rate assessment in the phase-II authentication phase

Phase-II Authentication Phase	Failure Rate (%)
Identity verification	20%
Context Verification	15%
Behavior Verification	65%



Phase II, with its homomorphic encryption and multiphase authentication process, introduced slightly higher response times and verification delays but achieved a higher overall success rate due to its multifaceted approach. Both phases demonstrated high levels of security, with Phase II showing a marginally higher failure rate due to the complexity of behavioral and contextual verification. However, Phase II also demonstrated stronger security against sophisticated attacks because of its multiphase, adaptive approach. Phase I was highly scalable, with a minimal increase in time delays even as the number of replicas increased. Phase II showed moderate scalability, with a slight increase in response times as the complexity of verification operations grew. To validate the effectiveness of the proposed Multiphase Zero-Trust Authentication Framework (MZTAF), an extensive comparison was carried out with leading zero-trust authentication models recently proposed in the literature, as shown in Table 7. The comparison covered multiple evaluation criteria, including authentication accuracy, average response time, scalability under network load, and tolerance to node failures. The proposed MZTAF framework consistently outperforms existing models across key performance indicators.

Authentication Accuracy: MZTAF achieves 94% accuracy in device-level verification and 95% in user-level authentication. This is notably higher than other models, which range between 88% and 92%. The enhanced accuracy is due to the layered verification process combining identity, contextual, and behavioural checks.

Response Time: Despite involving multiphase verification and encrypted computations, MZTAF maintains a low average response time of 0.02 seconds. This is due to the lightweight XOR-based operations in Phase I and the use of optimized homomorphic functions in Phase II.

Fault Tolerance: Using key replication and consensus mechanisms, the framework remains resilient even when some nodes are compromised.

Table 7: Comparative Analysis of Zero-Trust Authentication Frameworks

Framework	Core Mechanism	Authentication Accuracy	Average Response Time (s)	Fault Tolerance	Scalability
(Huang, 2024)	Replicated Key Authentication	89%	0.05	Moderate	Moderate
(Azad et al., 2024)	Threat-Based ZTA with MITRE Mapping	91%	0.06	Low	Limited
(Rivera et al., 2024)	Blockchain-Based Distributed Authentication	92%	0.08	High	Moderate
MFA-ZTA (Dargaoui et al., 2025)	Multi-Factor + Location + Token	88%	0.04	Low	Limited
MZTAF (Proposed)	Replicated Key + Homomorphic User Verification	94% (Phase I) / 95% (Phase II)	0.02	High	High

Compared to blockchain-based models, which provide high fault tolerance but incur latency, MZTAF balances both speed and resilience.

Scalability: The architecture was tested with increased node counts and authentication requests. Results show stable performance, indicating the framework's ability to scale efficiently in dynamic environments such as enterprise networks or cloud-based infrastructures.

The comparative analysis confirms that MZTAF offers superior performance in terms of security, adaptability, and efficiency. It bridges the limitations of single-factor and centralized models by adopting a layered, decentralized verification strategy that is both robust and scalable.

Conclusion

The development of secure and scalable authentication systems is essential for safeguarding digital resources in an increasingly complex and distributed world. Traditional authentication models, which rely on passwords, tokens, or biometric data, have proven to be vulnerable to various forms of attack. These systems often struggle to provide robust security in dynamic environments where both devices and users may change their access patterns or may be subject to advanced cyber threats. The Multiphase Zero-Trust Authentication Framework (MZTAF), as presented in this research, offers a comprehensive solution to these challenges by integrating decentralized-level authentication and privacy-preserving user-level authentication through homomorphic encryption. While this study focused on a simulated environment, future work will involve pilot deployment in enterprise networks to validate real-world applicability and performance under varying load and threat conditions. MZTAF introduces significant innovations in the way authentication is conducted. First, its use of replicated key-based device authentication ensures that devices are continuously verified in a fault-tolerant manner, without relying on a central authority. This reduces the risk of single points of failure and enhances the security of device access in zero-trust networks. Second, by utilizing homomorphic encryption, MZTAF preserves the privacy of sensitive user data during authentication, making it suitable for privacy-sensitive applications while still allowing for detailed behavioral analysis and contextual checks. The three-phase user authentication process, which involves identity, context, and behavior verification, adds an extra layer of granularity, thereby reducing the risk of unauthorized access.

The experimental results highlight the effectiveness of MZTAF, achieving 94% success in device authentication (Phase I) and 95% success in user authentication (Phase II). The results also demonstrate the scalability of the framework, with minimal performance overhead even as the number of replicas increases. The system's ability to adaptively grant access based on the aggregate scores from all authentication phases ensures that security is maintained without sacrificing user convenience. A key strength of MZTAF lies in its dynamic access control mechanism, which adjusts access permissions based on the outcomes of each authentication phase. This flexibility allows the system to handle a wide range of access scenarios, from granting full access to highly trusted users to imposing restrictions on users whose authentication results are borderline. This approach represents a significant departure from traditional models, which typically rely on static authentication results. While this work demonstrates high authentication accuracy, formal security proofs will be developed in future iterations using standard frameworks such as IND-CPA and zero-knowledge proof models. A comparative study against cryptographic benchmarks will also be explored.

Despite its promising results, there are areas for future improvement. The behavioral verification phase in Phase II, while effective, can be further optimized. Future versions of MZTAF could incorporate machine learning techniques to improve behavioral analysis, reducing the failure rate in this phase. Additionally, real-time deployment and testing in more complex, large-scale environments will help assess the framework's performance under more dynamic conditions. Overall, the Multiphase Zero-Trust Authentication Framework (MZTAF) represents a significant advancement in the field of authentication systems. By combining replicated key-based device authentication with homomorphic encryption for user authentication, it offers a robust, scalable, and secure solution for modern zero-trust environments. Its multiphase approach, dynamic access control, and privacy-preserving techniques provide a higher level of security than traditional authentication models, making it an ideal solution for securing distributed, privacy-sensitive systems in today’s threat landscape.

Acknowledgment

We thank the Journal of Computer Science for helping us share the findings of this research with a wide range of audiences.

Funding Information

This research work was not funded by any funding agency, either public, commercial, or non-profit making organizations.

Author’s Contributions

Modisaotsile Marope: A PhD scholar who designed the study and carried out experiments.

Venumadhav Kuthadi: Assisted with supervision and refining research core ideas.

Rajalakshmi Selvaraj: Assisted with supervision and review of methods and materials used in the manuscript.

Thabo Semong: Supported with coding in MATLAB and proofreading the manuscript.

Tshiamo Sigwele: Assisted with problem and solution implementation. All authors approved the final manuscript.

Ethics

This work adheres to all research ethical guidelines stated in the Journal of Computer Science.

References

Ahn, G., Jang, J., Choi, S., & Shin, D. (2024). Research on Improving Cyber Resilience by Integrating the Zero Trust Security Model With the MITRE ATT&CK Matrix. *IEEE Access*, *12*, 89291–89309. https://doi.org/10.1109/access.2024.3417182

Akavia, A., Gentry, C., Halevi, S., & Vald, M. (2025). Achievable CCA2 Relaxation for Homomorphic Encryption. *Journal of Cryptology*, *38*(1), 5. https://doi.org/10.1007/s00145-024-09526-1

Aleisa, M. A. (2025). Blockchain-Enabled Zero Trust Architecture for Privacy-Preserving Cybersecurity in IoT Environments. *IEEE Access*, *13*, 18660–18676. https://doi.org/10.1109/access.2025.3529309

Arenas, Á., Ray, G., Hidalgo, A., & Urueña, A. (2024). How to keep your information secure? Toward a better understanding of users security behavior. *Technological Forecasting and Social Change*, *198*, 123028. https://doi.org/10.1016/j.techfore.2023.123028

Aswathy, R., H., Srithar, S., Dayana, K. R., Padmavathi, A., & Suresh, P. (2023). MIAS: An IoT based Multiphase Identity Authentication Server for Enabling Secure Communication. *Journal of Internet Services and Information Security*, *13*(4), 114–126. https://doi.org/10.58346/jisis.2023.i4.008

Azad, M. A., Abdullah, S., Arshad, J., Lallie, H., & Ahmed, Y. H. (2024). Verify and trust: A multidimensional survey of zero-trust security in the age of IoT. *Internet of Things*, *27*, 101227. https://doi.org/10.1016/j.iot.2024.101227

Bashir, T. (2024). Zero Trust Architecture: Enhancing Cybersecurity in Enterprise Networks. *Journal of Computer Science and Technology Studies*, *6*(4), 54–59. https://doi.org/10.32996/jcsts.2024.6.4.8

Bast, C., & Yeh, K.-H. (2024). Emerging Authentication Technologies for Zero Trust on the Internet of Things. *Symmetry*, *16*(8), 993. https://doi.org/10.3390/sym16080993

Bojič Burgos, J., & Pustišek, M. (2024). Decentralized IoT Data Authentication with Signature Aggregation. *Sensors*, *24*(3), 1037. https://doi.org/10.3390/s24031037

Chaturvedi, I., Pawar, P. M., Muthalagu, R., & Tamizharasan, P. S. (2024). Zero Trust Security Architecture for Digital Privacy in Healthcare. *Information Technology Security*, 1–23. https://doi.org/10.1007/978-981-97-0407-1_1

Dargaoui, S., Azrour, M., El Allaoui, A., Guezzaz, A., Alabdulatif, A., & Ahmad, S. (2025). ECC-Based Anonymous and Multi-factor Authentication Scheme for IoT Environment. *International Journal of Online and Biomedical Engineering (IJOE)*, *21*(01), 56–75. https://doi.org/10.3991/ijoe.v21i01.51633

Farhat, A., Eldosouky, A., Ibnkahla, M., & Matrawy, A. (2025). Interaction-Aware Trust Management Scheme for IoT Systems With Machine-Learning-Based Attack Detection. *IEEE Internet of Things Journal*, *12*(11), 17169–17182. https://doi.org/10.1109/jiot.2025.3539646

Goodness Hassan, Y., Collins, A., Babatunde, G. O., Alabi, A. A., & Mustapha, S. D. (2025). Holistic software solutions for securing Iot ecosystems against data theft and network-based cyber threats. *Gulf Journal of Advance Business Research*, *3*(1), 252–261. https://doi.org/10.51594/gjabr.v3i1.77

Harrison, H. (2023). Why zero trust should start at the endpoint. *Network Security*, *2023*(7). https://doi.org/10.12968/s1353-4858(23)70032-9

Huang, W. (2024). ECC-based three-factor authentication and key agreement scheme for wireless sensor networks. *Scientific Reports*, *14*(1), 1787. https://doi.org/10.1038/s41598-024-52134-z

Huber, B., & Kandah, F. (2024). Zero Trust+: A Trusted-based Zero Trust architecture for IoT at Scale. *Proceeding in the IEEE International Conference on Consumer Electronics (ICCE)*, 1–6. https://doi.org/10.1109/icce59016.2024.10444321

Itodo, C., & Ozer, M. (2024). Multivocal literature review on zero-trust security implementation. *Computers & Security*, *141*, 103827. https://doi.org/10.1016/j.cose.2024.103827

Rivera, J. J. D., Muhammad, A., & Song, W. C. (2024). Securing Digital Identity in the Zero Trust Architecture: A Blockchain Approach to Privacy-Focused Multi-Factor Authentication. *IEEE Open Journal of the Communications Society*, *5*, 2792–2814. https://doi.org/10.1109/ojcoms.2024.3391728

Kerl, M., Bodin, U., & Schelén, O. (2025). Privacy-preserving attribute-based access control using homomorphic encryption. *Cybersecurity*, *8*(1), 5. https://doi.org/10.1186/s42400-024-00323-8

Lavanya, S., & Saravanakumar, N. M. (2023). Graph-Based Replication and Two Factor Authentication in Cloud Computing. *Computer Systems Science and Engineering*, *45*(3), 2869–2883. https://doi.org/10.32604/csse.2023.029040

Liu, Z. (2024). A Review of Advancements and Applications of Pre-Trained Language Models in Cybersecurity. Proceeding in the International Symposium on Digital Forensics and Security (ISDFS), 1-10. https://doi.org/10.1109/isdfs60797.2024.10527236

Rawther, S., & Sivaji, S. (2025). Protecting Cloud Computing Environments from Malicious Attacks using Multi-factor Authentication and Modified DNA Cryptography. *Recent Patents on Engineering*, *19*(1), 1–21. https://doi.org/10.2174/1872212118666230905141926

Trott, D. (2024). A zero-trust journey through the threat landscape. *Network Security*, *2024*(2). https://doi.org/10.12968/s1353-4858(24)70008-7