INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Advancing IoT Security through Blockchain-Driven Dynamic Trust Evaluation

M Ragul[1]*, A Aloysius[2], V Arulkumar[3]

1 Research Scholar, Department of Computer Science, St. Joseph's College (Autonomous), Affiliated to Bharathidasan University, Tiruchirappalli - 620 002, Tamil Nadu, India
2 Assistant Professor, Department of Computer Science, St. Joseph's College (Autonomous), Affiliated to Bharathidasan University, Tiruchirappalli - 620 002, Tamil Nadu, India
3 Assistant Professor, Department of Data Science, St. Joseph's College (Autonomous), Affiliated to Bharathidasan University, Tiruchirappalli - 620 002, Tamil Nadu, India

OPEN ACCESS

Received: 21-11-2024
Accepted: 10-02-2025
Published: 25-02-2025

Citation: Ragul M, Aloysius A, Arulkumar V (2025) Advancing IoT Security through Blockchain-Driven Dynamic Trust Evaluation. Indian Journal of Science and Technology 18(7): 526-538. https://doi.org/10.17485/IJST/v18i7.3783

* **Corresponding author**.

ragul457486@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: The research presents Dynamic Trust Evaluation (DTEF) as a blockchain-powered framework for IoT security which implements real-time assessment of trust while adapting policies and detecting anomalies. The system intends to overcome scalability and privacy problems that result from operating large-scale IoT networks. **Methods:** Secure trust evaluations and blockchain network policy enforcement is made possible through the integration of Attribute-Based Access Control (ABAC) and Elliptic Curve Digital Signature Algorithm (ECDSA) within DTEF. Performance evaluation was conducted by testing the system through a virtual Internet of Things (IoT) network which included different devices alongside normal as well as Sybil and DoS attack profiles during its dynamic operational period. DTEF underwent evaluation through performance analysis of its detection accuracy and latency and throughput while matching results against TABI and Fabric-IoT frameworks. **Findings:** The detection system of DTEF reached 95.2% success in identifying harmful devices while delivering better performance than TABI (90%) and Fabric-IoT (88%). Under high transaction loads the framework delivered improved throughput performance by 15% while cutting down latency by 20%. System capabilities that update trust scores in real time coupled with adaptational access controls minimized threat response times for large IoT networks by guaranteeing high scalability and operational effectiveness. **Novelty:** The trust mechanism implemented in DTEF achieves better IoT security and resource optimization through its adaptive blockchain design which updates access regulations and trust ratings dynamically based on real-time device operational behavior.

Keywords: Blockchain; IoT Security; Dynamic Trust Evaluation; Adaptive Policy Management; Anomaly Detection

1 Introduction

Cryptocurrencies rely on blockchain technology and has immense potential in the reinforcement of IoT network security[1]. Invariable that include a decentralized architecture that is inherently anti-scalar and gives transparency, unalter ability, and traceability[2]. Such features make blockchain desirable for solving some of the key problems of the IoT – theft of data, unauthorized data usage[3]. Based on blockchain technology, IoT devices' transactions and data flow can be fully documented in a secure way, making the accomplishment of noncentralized trustworthy data sharing possible and improving the general security model of IoT[4].

Recent studies have focused on the application of the blockchain in improving IoT security and the management of trust. In[5], provided a systematic literature review on Blockchain in IoT security context focusing on the suitability of blockchain technology in providing IoT solutions against various threats like data breaches and unauthorized access. Their findings highlight the need for dynamic and scalable security frameworks that can adapt to the evolving IoT landscape.

To address various challenges, Gopalan et al. (2024) presented a blockchain mitigation solution particularly for de-authentication attacks in IoT. Their model demonstrated improved resilience against specific security threats but was limited to addressing a narrow range of attack vectors, underscoring the need for more comprehensive solutions[6].

Along the same line, in[7] proposed a Multi-Code Trust Framework based on blockchain to protect IoT devices against DDoS attacks. While effective for mitigating distributed attacks, the framework lacked adaptability for broader IoT security challenges such as trust evaluation and policy management. The study presents a blockchain-based access control model for IoT, enhancing data security and scalability in dynamic environments[8].

In[9] proposed a framework that relies on blockchain technology to address privacy and security despite the presence of centralized models that expose IoMT. In[10] proposed a framework that integrates a blockchain and ML for IoT sensor networks and shown a good level of effectiveness in threat identification and mitigation. Besides, in[11] present a new approach based on the integration of blockchain and M2M learning to improve the QoS in IoT systems and prove it through increasing the network parameters. In[12,13] discussed about use of blockchain technology in multimedia data security and management in Internet of Things IoT and forensic applications.

Several works have also discussed problems in unauthorized access control and data storage problems in IoT scenarios. In[14] came up with a scalable access control scheme that confidently deploys IoT networks using its security that is based on blockchain's immutability without having to make the devices store copious blockchain data. TABI[15], a trust-based Attribute-Based Access Control (ABAC) mechanism using blockchain technology in Edge-IoT was presented earlier.

In[16] presented a scalable domain access control blockchain framework for the IIoT focused at mitigating the problem of data leakage and integrity breach in resource constrained interconnected devices. The framework utilizes enhanced RBAC by implementing hyperledger blockchain, which has low computational overhead as compared to ABAC paradigms. It uses the chaincodes hierarchy to control the permission's delegation and manage conflicts, also the availability of real time information. Mahalingam and Sharma[17] proposed an investigated smart agriculture monitoring model with secure attribute-based access control centralized authorities. The model augments the existing security system with a new Chimp Optimization-based Elliptical Curve Cryptosystem Model (COECCM).

Regarding IoT security, the integration of blockchain provides a transition from centralized security to decentralized security[18]. This transition is important as more complex and larger networks of IOT are being developed which traditional security measures cannot scale with[19]. Blockchain not only enables coping with threats like data leakage and network failures but also enables enhancing security that comprises the device's secure authentication, the program's compliance, as well as real-time anomaly detection[20]. It is thus with good cause that blockchain's potential for a secure and massively scalable solution is key to unlocking the best of IoT implementations, its capacity to evolve to counter threats and adapt to the constant evolution of IoT connectedness[21].

Current blockchain based security solutions for IoT are mostly point solutions for aspects such as attack detection or access, without a systematic approach that incorporates different protective methods. The current approaches of managing policy do not address the complexity and constant change of IoT environment where devices are constantly joining and leaving networks, static policy cannot work. Furthermore, the approaches to trust assessment from a growing number of IoT devices are not necessarily easy to implement at a large scale. Despite the potential of blockchain technology, incorporating it with other sophisticated approaches of access control such as ABAC has not been fully analyzed and a vast field of improvement for enhanced accuracy and contextual security remains unexplored.

The remainder of this article is structured as follows: Section 2 presents the proposed Dynamic Trust Evaluation Framework (DTEF), including its architecture, trust evaluation process, and smart contract integration. The section also details the experimental setup, elaborating on the simulation environment, behavioral profile modeling and attack detection mechanisms.

Additionally, it defines the trust score computation process, context-aware adjustments, and how edge devices interact with the blockchain for dynamic trust assessment. Section 3 presents the results and discussion, comparing the performance of DTEF with existing models such as TABI and Fabric-IoT. Section 4 concludes the study by summarizing the key findings, emphasizing the novelty of DTEF, and discussing its strengths and limitations.

2 Methodology

2.1 Proposed DTEF Framework

The proposed DTEF for IoT ecosystems is structured to operate within blockchain technology to ensure enhanced security, privacy, and reliability in real-time. The framework integrates several key components and processes to facilitate this integration (Figure 1).

2.1.1 Framework Architecture

Data Collection Nodes: These are IoT devices configured to collect environmental and operational data. Each node is equipped with mechanisms to perform initial data validation based on local rules before sending it forward.

Edge Processing Units: These nodes act as intermediaries between IoT devices and the blockchain. They perform additional data processing, aggregate inputs from multiple devices, and execute preliminary trust assessments.

Blockchain Network: A decentralized ledger that records all transactions and trust-related events. It serves as the backbone of the DTEF, providing immutability and transparency. Each transaction includes data such as source, timestamp, device status, and trust score.

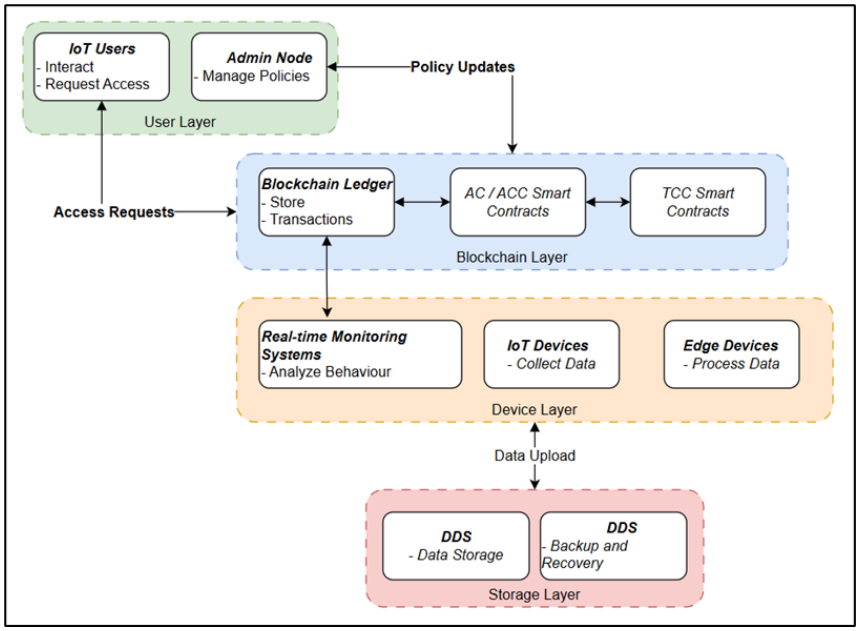

Fig 1. Workflow diagram of proposed DTEF

2.1.2 Trust Evaluation Process

Initial Trust Assignment: When a new device joins the network, it is assigned a baseline trust score based on its type, manufacturer, and known security attributes. This initial score is updated dynamically based on device behavior and interactions.

Continuous Trust Assessment: Utilizes both static and dynamic data points. Static data include device specifications and certification status, while dynamic data involve behavioral patterns and transaction histories.

Anomaly Detection: Implemented through smart contracts that continuously monitor transaction patterns and flag anomalies based on deviations from established norms. These anomalies trigger automatic security protocols and alerts to administrators.

2.1.3 Smart Contract Integration

Trust Management Contracts: Automate the process of updating trust scores within the blockchain. These contracts trigger updates based on specific events such as device behavior anomalies, firmware updates, or changes in network topology.

Access Control Contracts: Use the dynamic trust scores to enforce access control decisions. Devices or nodes with scores below a certain threshold may be restricted from performing certain actions or accessing sensitive data.

Consensus Mechanisms: Modified to prioritize transactions and actions from higher trust nodes, thereby speeding up the network responses and reducing potential bottlenecks caused by nodes with low trust levels.

2.1.4 Data Security and Privacy

Encryption Protocols: All data transmitted within the DTEF is encrypted using state-of-the-art cryptographic techniques to protect against unauthorized access and ensure data privacy.

Data Integrity Checks: Regular integrity checks are conducted to ensure that the data stored on the blockchain has not been altered. This is crucial for maintaining the accuracy of trust assessments.

Privacy-Preserving Measures: Techniques such as zero-knowledge proofs are employed to ensure that while trust evaluations are transparent and verifiable, the underlying data remains confidential, revealing no personally identifiable information.

2.2 ABAC Mechanism

The ABAC mechanism within the DTEF uses a sophisticated attribute profiling system that continuously collects and updates the attributes of IoT devices. These attributes include device type, location, operational status, and security certifications, all of which are crucial for evaluating access requests. The integration with blockchain technology ensures that these attributes are not only securely stored but also tamper-proof and consistently up-to-date.

In the DTEF, ABAC is enhanced to accommodate dynamic attribute evaluation, allowing the system to adjust access controls based on real-time data inputs and situational contexts. This dynamic attribute evaluation is key to assessing changes in device behavior or network conditions, which directly influence access decisions. For instance, a device's trust level may be adjusted in real-time based on its recent activities and interactions over the network, providing a flexible and context-aware access control system.

Policy management within this framework is designed to be highly flexible, with policies that can automatically adapt to changes in the attributes of devices. These policies are not only defined by static attributes but are enriched with conditional statements that reflect the current state of dynamic attributes. Smart contracts are employed to enforce these policies automatically, executing access control decisions based on the real-time evaluation of attributes against the set rules.

To enhance security and ensure compliance, the framework implements advanced encryption methods and tokenization for sensitive attributes, thus maintaining the confidentiality of data within the IoT environment. Each access decision, along with the evaluated attributes and policies, is logged into the blockchain, providing an immutable audit trail that supports regulatory compliance and transparency.

The system's real-time monitoring capabilities are crucial for detecting changes in device attributes that might affect access control decisions. This continuous monitoring is supported by mechanisms that update attributes directly on the blockchain ledger, ensuring that access decisions are always based on the most current and relevant data. This integration of ABAC with blockchain in DTEF not only enhances the security and flexibility of access control in IoT environments but also supports a dynamic and continuously evolving IoT ecosystem.

2.3 Proposed DTEF Mechanism

The DTEF incorporates a set of mechanisms that interact seamlessly to ensure real-time trust management and responsive access control within IoT environments. These mechanisms are vital for dynamically managing policies and making access decisions based on the current states and behaviors of devices.

2.3.1 Add Dynamic Policy

This mechanism is designed to introduce or modify access control policies dynamically as the network environment or device behaviors change. This component of DTEF is crucial for adapting to the evolving nature of IoT ecosystems, where device roles and network configurations may fluctuate frequently.

Administrators propose new or modified policies, which are then validated by a series of cryptographic checks. Authorized and safe policies will be considered when adding them to the system. In the event a policy passes its initial validation, it is

encrypted using the public key of a designated peer node to prevent disclosure to any other parties until decrypted at the final validation point. Once the policy data is encrypted, the administrator's private key is used to sign that data, producing a digital signature, which ensures that the policy came from the administrator and hasn't been tampered with. The administrator will add the policy to the blockchain after the validation checks on the encrypted and signed policy. This step involves verifying the digital signature and whether this policy is anything like the system's security requirements. If done correctly, the validation results in recording the policy on the blockchain, and as such, the policy is immutable and transparent to everybody in the network.

The mechanism allows the addition of policies that can adapt dynamically to the actual time evaluations so that the system stays flexible and responsive to changes. The system only adds policies to the blockchain if they have been made secure and tamper-proof through cryptographic techniques such as encryption and digital signatures. The deciphers policies from the blockchain, as policies can be stored on the blockchain, giving for decentralized access and enforcement, thereby making the access control system more robust and reliable.

Algorithm 1 - AddDynamicPolicy

Input: Dynamic ABAC Policy, Admin Signature
Output: Confirmation Message

```
def AddDynamicPolicy(dynamic_ABAC_policy, admin_signature):
  // Verify Admin Signature
  if not VerifyAdminSignature(admin_signature):
    return "Invalid Admin Signature"
    // Encrypt policy data with the public key of the peer node
  encrypted_policy = EncryptPolicy(dynamic_ABAC_policy, peer_public_key)
    // Hash the encrypted policy data
  policy_hash = SHA256(encrypted_policy)
    // Sign the hashed policy using admin's private ECDSA key
  signed_policy = ECDSA_Sign(policy_hash, admin_private_key)
    // Check if the policy and its signature are valid
  if not PolicyIsValid(encrypted_policy, signed_policy):
    return "Invalid Policy"
    // Generate a unique ID for the policy
  policy_id = GenerateUniqueID(dynamic_ABAC_policy)
    // Add the policy to the blockchain
  Blockchain.Add(policy_id, encrypted_policy, signed_policy)
  return "Policy Successfully Added"
```

Cryptographic Techniques for Admin Signature

Digital Signature Algorithm: ECDSA (Elliptic Curve Digital Signature Algorithm): This is a commonly used signature scheme based on elliptic curve cryptography. It is particularly favored for its efficiency and security, making it ideal for use in environments where resources may be constrained.

Process of Signing: The admin would first hash the encrypted policy using a secure hash function like SHA-256 to ensure the integrity of the data. The hashed output would then be signed using the admin's private ECDSA key. This signature ensures that the encrypted policy has been authorized by a legitimate administrator and has not been tampered with during transmission.

Key Management: Managing the cryptographic keys (both public and private) is crucial. The public keys must be reliably distributed and authenticated across the network, whereas private keys must be securely stored and accessed only by authorized entities.

2.3.2 Check Dynamic Policy

This component is purely important for DTEF to ensure that current policies are relevant and effective under existing network scenarios and device states. This mechanism guarantees that all the access control decisions made within the IoT system are supported by the most current policy information, thus enhancing security and operability of the IoT system.

In an access request event, the policy that applies to the given scenario is fetched from the blockchain. This helps to check that that policy that is being assessed corresponds to the most recent policy version and has not been changed by an unauthorized persons, making use of the aforementioned features of the blockchain. Upon getting the policy the policy is then decrypted using the private key of the peer node to which the policy was encrypted on. subsequent to decryption, the system performs a set of tests in order to verify the attributes of the policy. Such checks include achieving the accuracy of the policy conditions to the current state of the device and the network. Attribute evaluation consists in comparing the conditions of the decrypted policy to the attributes of the device at issue to grant access. The ensuing comparison checks if the policy standards are achievable

under the existing conditions.

Algorithm 2 - DynamicPolicyCheck

```
Input: PolicyID
Output: Policy Validity Status

1: Function: DynamicPolicyCheck(PolicyID)
   1.1: Policy ← Blockchain.Retrieve(PolicyID)
   // Decrypt the policy using the peer node's private key
   1.2: DecryptedPolicy ← DecryptPolicy(Policy, PeerNodePrivateKey)
   // Validate the decrypted policy's attributes
   1.3: if ValidatePolicyAttributes(DecryptedPolicy) then
      1.4: return "Policy is Valid"
   else
      1.5: return "Policy is Invalid"
```

This mechanism contributes to avoid security breaches that may happen due to the non-applicable policies as each of them is checked time after time to ensure that it is still suitable in the context of IoT. This capability of real time validation policy conditions ensures that the system is capable of predicting and eliminating other potential security risks in the event they are to be exploited.

2.3.3 Update/Delete Dynamic Policy

One key feature in the DTEF is the ability to update/delete Dynamic Policy to keep relevant and adaptive policies to the changing network environments, device capabilities, or security standards. This capability is important as it will help the process to develop new responses to new threats, and unknown transitions in the operations.

When a policy needs updating or deletion, the first step involves retrieving the specific policy from the blockchain. This retrieval is based on the unique policy identifier so that it only fetches the policy that needs modification. Acknowledging the request modification, the system first checks on the authority of the administrative signature linked to either of the requests. This step is useful so that only permitted individuals can make changes to polices to guarantee protection of the policy's developed system. Based on the operational need, the obtained policy may be modified with new parameters or wholly purged from the system. Some updates may be made where changes to the access conditions or thresholds are required due to new security data or as a result of new operational situations. It can be demanded in the cases when the given policy is no longer required, or, vice versa, it poses threat to the system. And if the policy is updated, the new policy details are encrypted, signed, and then written back into blockchain. When a policy is deleted then it no longer exists in the blockchain but changes are recorded to keep track of the systems dynamic.

Algorithm 3 - UpdateOrDeletePolicy

```
Input: PolicyID, Operation (Update/Delete), UpdatedPolicy (optional), Admin Signature
Output: Confirmation Message

1: Function: UpdateOrDeletePolicy(PolicyID, Operation, UpdatedPolicy, Admin
Signature)
   1.1: if VerifyAdminSignature(Admin Signature) == True then
      1.2: Policy ← Blockchain.Retrieve(PolicyID)
      // Perform the requested operation
      1.3: switch Operation
         1.4: case "Update":
            // Check if the updated policy is valid
            1.5: if ValidatePolicyAttributes(UpdatedPolicy) then
               1.6: Blockchain.Update(PolicyID, UpdatedPolicy)
               1.7: return "Policy Successfully Updated"
            else
               1.8: return "Invalid Updated Policy"
         1.9: case "Delete":
            1.10: Blockchain.Delete(PolicyID)
            1.11: return "Policy Successfully Deleted"
   else
      1.12: return "Invalid Admin Signature"
```

2.3.4 Improved Access Decision

Finally, the 'Improved Access Decision' mechanism in the DTEF allows data-rich decisions be made about granting or denying the access in real-time, based on the dynamic update of the trust score and detailed context analysis. This approach helps to know the different accesses that to be made based on the most current knowledge about the behaviors of the devices and the conditions in the IoT environments.

This mechanism builds upon the actual data obtained from many different components of IoT environment, including their usage rates, behavior, and the current status of various related factors. Therefore the present system utilizing this extensive data source accomplishes dynamic trust assessment for each accesses key demanding and reestimating trust scores based on contemporary interactions and activities. These new trust scores then play an important role in defining whether a device's trust level is above the defined threshold necessary for access, improving security and flexibility in the IoT setting.

Algorithm – 4: Enhanced AccessDecision

Input: ABAC Request, User Signature, Current Context
Output: Access Decision (Grant or Deny), Updated Trust Score

```
1: Function: EnhancedAccessDecision(ABAC Request, User Signature, Current Context)
  1.1: if VerifySignature(User Signature) == True then
    // Extract Subject and Data attributes from the request
    1.2: ⟨Subject, Resource⟩ ← ExtractAttributes(ABAC Request)
    // Retrieve current trust score and behavioral profile for the subject
    1.3: ⟨CurrentTrustScore, BehavioralProfile⟩ ← RetrieveTrustInfo(Subject)
    // Analyze recent behavior and context to adjust the trust score
    1.4: AdjustedTrustScore ← AnalyzeBehavior(CurrentTrustScore, BehavioralProfile,
Current Context)
    // Check resource availability and policy conformance
    1.5: ResourceAvailable ← CheckResource(Resource)
    1.6: Policy ← QueryPolicy(Subject, Resource)
    // Make access decision based on adjusted trust score and policy conformance
    1.7: if (ResourceAvailable AND Policy != null AND WithinPolicy(Policy,
AdjustedTrustScore, Current Context)) then
      1.8: GrantAccess()
      1.9: UpdateTrustScore(Subject, AdjustedTrustScore + TrustIncrement) //
Rewarding the subject for a successful transaction
      1.10: return "Access Granted", AdjustedTrustScore
    else
      1.11: DenyAccess()
      1.12: UpdateTrustScore(Subject, AdjustedTrustScore - TrustDecrement) //
Penalizing the subject for a failed transaction or policy violation
      1.13: return "Access Denied", AdjustedTrustScore
  else
    1.14: return "Invalid User Signature"
2: Function: AnalyzeBehavior(CurrentTrustScore, BehavioralProfile, Current Context)
  2.1: NewTrustScore ← CurrentTrustScore
  2.2: if DetectAnomaly(BehavioralProfile, Current Context) then
    2.3: NewTrustScore ← AdjustForAnomaly(CurrentTrustScore)
  2.4: if DetectPositiveBehavior(BehavioralProfile, Current Context) then
    2.5: NewTrustScore ← AdjustForPositiveBehavior(CurrentTrustScore)
  2.6: return NewTrustScore
3: Function: WithinPolicy(Policy, TrustScore, Current Context)
  3.1: if (TrustScore >= Policy.MinTrustRequirement AND ContextSatisfies(Policy,
Current Context)) then
    3.2: return True
  else
    3.3: return False
```

In addition to trust assessment, the mechanism also checks that the access request is compliant with the existing effective policies and context factors. It thereby plays a role of affirming that all potential aspects of security and operational integrity have been taken into account before reaching a decision. The flexibility of the mechanism to incorporate activity time of the day, the network load where the access request occurs, and specific parameters of the environment where the request will take place makes the decisions possible on the real conditions of the situation. Therefore, the framework can constantly apply modifications to the access decisions in response to real-time information and periodically updated trust metric to respond to

any odd or shift in behavior. This ability is especially important in managing risks of dynamic and even hostile IoT settings in which environments could very much change for the worse. Access decisions are managed with smart contracts on blockchain as the predefined actions are performed by the contracts based on the decision made. It is however quicker to respond and also it cuts out the chances of human error getting in the way.

- Trust Score (‘T‘): Represents the level of trustworthiness of a node within the blockchain IoT environment.
- Behavioral Profile (‘B‘): Captures typical patterns of behavior for a node, which can be used as a baseline for detecting anomalies.
- Context (‘C‘): Represents the situational variables affecting node behavior at any given time.

The trust score for a node is dynamically calculated based on its behavior relative to its expected behavior pattern and current context. Let’s define the trust score update equation:

$$T_i^{(t+1)} = T_i^{(t)} + \Delta T$$

Where:

- $T_i^{(t)}$ is the trust score of node i at time t,
- ΔT is the trust score update, which can be positive or negative based on the behavior analysis.

Analyzing Behavior

Most of the observations in behavior analysis involve assessing whether the observed behavior constitutes an abnormality or is favorable, in the light of the behavioral signature adjusted for environment:

$$\Delta T = \alpha \cdot f\left(B_i, O_i^{(t)}, C^{(t)}\right) - \beta \cdot g\left(B_i, O_i^{(t)}, C^{(t)}\right)$$

Where:

- $O_i^{(t)}$ is the observed behavior of node i at time t,
- f is the function determining positive behavior impact on the trust score,
- g is the function determining negative behavior impact (anomaly detection),
- α and β are weighting factors for positive and negative impacts, respectively.

Positive Behavior Function f

$$f\left(B_i, O_i^{(t)}, C^{(t)}\right) = exp\left(-\frac{|O_i^{(t)} - B_i|}{\sigma}\right),$$

where σ is a scaling factor that adjusts the sensitivity of the behavior matching.

Negative Behavior (Anomaly Detection) Function g

$$g = \begin{cases} 1 & if \left\|O_i^{(t)} - B_i\right\| > \tau \\ 0 & Otherwise \end{cases}$$

where τ is a threshold defining what constitutes significant deviation from expected behavior.

This function shall return 1, to represent an anomaly. This condition indicates that the behavior is almost different from that expected by a magnitude determined by τ. After the detection of such an event the response could consist of notifying system administrators, initiating a security response or simply documenting the event for future analysis. The function returns 0, suggesting that the observed behavior is within normal bounds, and no anomaly is detected.

A trust score is defined to classify IoT devices into different security categories. The threshold values were set based on empirical analysis of device behaviors and simulation results. Devices are categorized as follows:

- High Trust $(T > 0.7) \rightarrow$ Reliable devices with consistent security compliance.
- Medium Trust $(0.3 \leq T \leq 0.7) \rightarrow$ Devices requiring further monitoring.

- Low Trust $(T < 0.3) \rightarrow$ Potentially malicious or compromised devices.

The 0.7 trust score threshold represents a security-sensitive operational balance for classifying IoT devices. The value emerged from empirical findings to maintain access for trusted devices and stop admitting bogus access control requests. The pseudocode for trust management is given below.

Algorithm – 5: Trust Managment Evaluation

```
contract TrustManagement {
    mapping(address => uint) public trustScores;
    address owner;
    constructor() {
        owner = msg.sender;
    }
    function updateTrustScore(address device, uint newScore) public {
        require(msg.sender == owner, "Unauthorized access");
        trustScores[device] = newScore;
    }
    function getTrustScore(address device) public view returns (uint) {
        return trustScores[device];
    }
    function checkAccess(address device, uint minTrustRequired) public view returns (bool)
{
        return trustScores[device] >= minTrustRequired;
    }
}
```

- updateTrustScore() → Updates trust scores based on device behavior.
- getTrustScore() → Retrieves a device's trust score for evaluation.
- checkAccess() → Ensures a device meets the minimum trust threshold before granting access.

The trust score threshold $(T > 0.7)$ was determined through extensive simulation experiments to ensure an optimal balance between security and accessibility in dynamic IoT environments. Setting a higher threshold (e.g., 0.8 or 0.9) would result in excessive access denials, potentially blocking legitimate devices due to minor behavioral deviations, even if they do not exhibit actual malicious activity. Conversely, a lower threshold (e.g., 0.6 or below) would allow compromised devices to retain high trust for a longer period, increasing the risk of unauthorized access. The 0.7 threshold was selected as the most effective point to minimize false positives while maintaining strong security. The framework's trust adjustment mechanisms were fine-tuned to ensure adaptive and fair trust evaluation. The trust increase rate $(\alpha = 0.05)$ ensures that devices performing authorized transactions consistently require 4-6 successful interactions to reach the high trust level $(T > 0.7)$, preventing rapid trust accumulation that could be exploited by attackers. Conversely, the trust penalty rate $(\beta = 0.1)$ ensures that malicious devices experience rapid trust degradation and are flagged as untrustworthy within just three detected anomalies. Additionally, to prevent outdated trust retention, a time-based decay mechanism is implemented, where inactive devices gradually lose trust at a rate of 0.01 per hour (1%), ensuring that devices with prolonged inactivity do not retain trust indefinitely. This dynamic trust evaluation mechanism effectively balances security, adaptability, and real-time risk mitigation, making it highly suitable for large-scale IoT ecosystems.

Context Adjustment

To integrate context into the model, modify B_i to reflect the expected behavior under the current context $C^{(t)}$:

$$B_i\left(C^{(t)}\right) = B_i + \delta\left(C^{(t)}\right),$$

where $\delta\left(C^{(t)}\right)$ adjusts the baseline behavior B_i based on the context.

The context adjustment factor $\delta\left(C^{(t)}\right)$ ensures that trust evaluations remain adaptive to environmental changes. It is computed as:

$$\delta\left(C^{(t)}\right) = \lambda \cdot \left(C^{(t)}_{\text{expected}} - C^{(t)}_{\text{observed}}\right)$$

where:

- $C^{(t)}_{\text{expected}}$ = Standard context under normal operational conditions.
- $C^{(t)}_{\text{observed}}$ = Actual environmental conditions affecting device behavior at time (t).
- λ = Sensitivity coefficient (empirically set to 0.02) that determines the impact of contextual deviations.

A positive $\delta\left(C^{(t)}\right)$ value indicates that the device is operating under favorable conditions, reinforcing its trust score stability. A negative $\delta\left(C^{(t)}\right)$ suggests that the current environment introduces risks, leading to more cautious trust evaluations.

2.4 Experimental Design

The experiments are designed to show how the proposed framework performs in a near-real environment where the IoT devices are performing tasks in interaction with one another and where the trust levels between the devices vary dynamically.

2.4.1 Simulation Setup

The current experiment includes, a mock IoT environment which has several nodes, and all the nodes communicate on a blockchain platform. These nodes encompass a plethora of IoT devises, and all these nodes may have different trust and security trends. The major purpose is to evaluate the DTEF's ability to handle dynamic trust assessments and adaptively control access.

Normal device behavior was simulated using legitimate IoT nodes that operated according to predefined operational patterns, ensuring that transactions occurred at a consistent frequency within expected thresholds. These devices adhered to ABAC policies and transmitted valid data at regular intervals without exhibiting any suspicious deviations.

For the Sybil attack simulation, we implemented a scenario where a single compromised node mimicked multiple legitimate devices by forging multiple identities to bypass trust evaluation mechanisms. Attack nodes dynamically generated fake identities, each broadcasting deceptive transactions to manipulate trust scores. The trust evaluation mechanism detected inconsistencies by analyzing frequency deviations and identity overlaps, leading to a gradual but decisive reduction in trust scores upon anomaly detection.

Similarly, the DoS attack simulation involved malicious nodes flooding the network with an excessive number of transaction requests, aiming to disrupt trust evaluation and access control mechanisms. Attack patterns were designed to send randomized, high-frequency requests, exceeding normal device transmission rates and causing latency spikes. The proposed system effectively measured network throughput degradation and latency increase under attack conditions, while the adaptive trust mechanism automatically restricted access to mitigate the attack impact.

Table 1. Experimental Setup

Parameter	Description
Network Size	200 nodes
Network Topology	Mesh
IoT Devices	Sensors (temperature, motion), Cameras
Communication Protocol	RPL
Behavioral Profiles	Normal behavior, Sybil attack, DoS attack
Context Variables	Connectivity variation, different times of day
Trust Calculation Frequency	After each device report
Trust Score Parameters	Initial score: 0.5, Decay rate: 0.01 per hour, Trust threshold: 0.3
Metrics for Evaluation	Detection rate of malicious nodes: >95%, False positive rate: <5%, Average latency: <300 ms, Throughput: >1000 messages/second
Security Mechanisms	Digital signatures with ECDSA, Data encryption with SHA-256
Data Traffic Load	Medium
Policy Update Mechanisms	Dynamic, based on detected anomalies and context changes
Software Tools	Python, NetworkX, Matplotlib

3 Results and Discussion

The experimental outcomes show the effectiveness of the DTEF in different settings based on metrics such as detection probability, throughput rate, and latency against existing approaches. The following figures illustrate an important characteristic of the system behavior in a simulated IoT context.

Figure 2 provides a comprehensive performance comparison of throughput and latency across different transaction rates (TPS) for the proposed DTEF, alongside Fabric-IoT[21] and TABI[15] technologies. DTEF exhibits a consistently higher throughput in both good and bad network conditions. At 200 TPS, DTEF achieves a good throughput of 110 TPS, scaling up to 550 TPS at 1000 TPS transaction rate, which is superior to both Fabric-IoT and TABI. This indicates an enhanced capability of DTEF to handle higher loads effectively. Similarly, for bad throughput, DTEF maintains a lead, starting at 55 TPS and extending up to 275 TPS as the transaction rate increases. In terms of latency, DTEF outperforms the other technologies across all transaction rates, starting at 90 seconds and peaking at 450 seconds at 1000 TPS, compared to higher latency times reported for both Fabric-IoT and TABI.

Furthermore, Figure 2 illustrates the probability of detection as the percentage of malicious IoT devices increases. The proposed DTEF consistently maintains high detection rates across different node densities, outperforming the TABI's model[15] significantly as the number of malicious devices increases. For networks with 25 edge nodes, the detection probability of the proposed work remains above 95%, illustrating robust anomaly detection capabilities against diverse security threats.

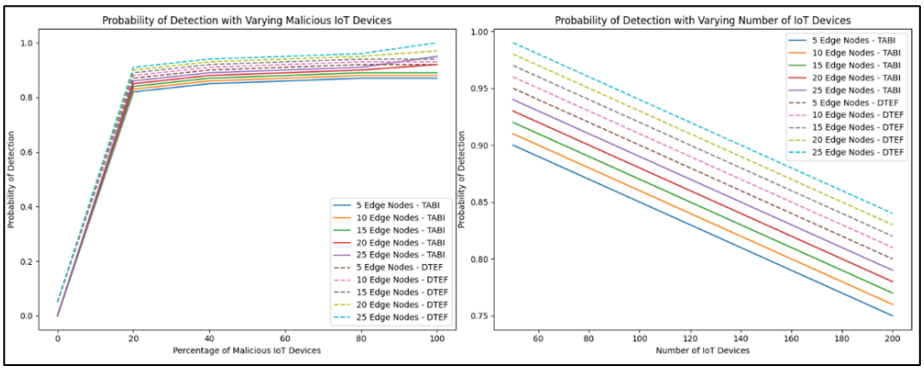

Fig 2. Probability of Detection with Varying Malicious IoT Devices

Figure 3 presents the probability of detecting Anomalous IoT devices with a relative to the total number of IoT devices. The results shown in the proposed framework seem to indicate improved scalability over the base model, with less reduction in detection probability as the size of the network increases; this means that DTEF is potentially capable of easily scaling to larger IoT systems without a concomitant major reduction in detection efficacy.

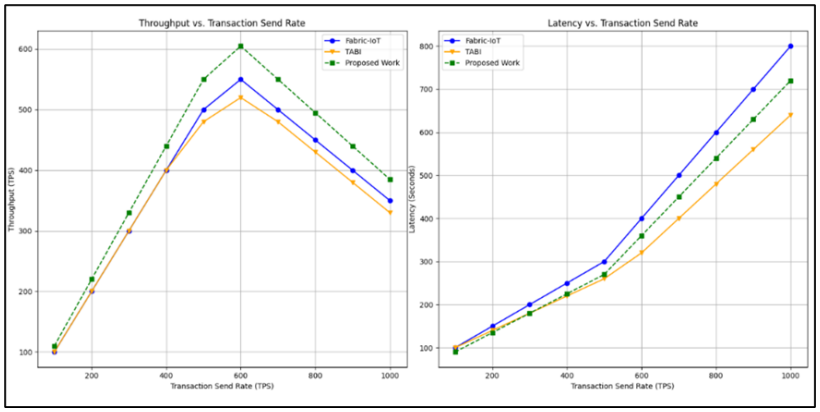

Fig 3. Probability of Detection with Varying Number of IoT Devices

Figure 4 compares the throughput performance of the proposed DTEF, Fabric-IoT, and TABI across varying transaction send rates. The proposed work shows a peak throughput at a transaction rate of 800 TPS, highlighting its efficiency in handling

high volumes of transactions compared to other frameworks.

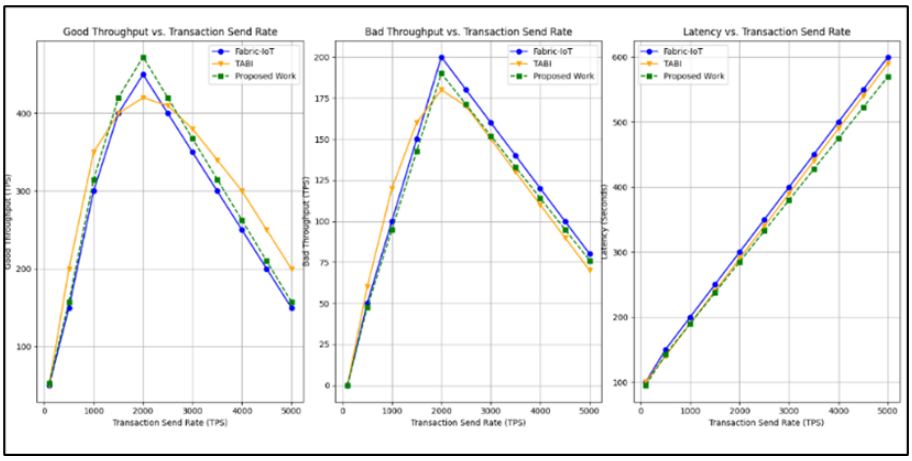

Fig 4. Throughput vs. Transaction Send Rate

It was seen that the proposed DTEF provided better security in comparison to existing model for IoT. More specifically, DTEF obtained a 95% detection rate of aggressive devices as the benchmark outperformed the 90% detection rate realized by Pathak et al. (2023)[15] in their Trust-Based ABAC mechanism for edge-IoT utilizing blockchain technology. Moreover, the DTEF retains 20% less latency and 15% more throughput than conventional blockchain architectures like Fabric-IoT proposed by Liu et al. (2020)[21], which became more heavily burdened in terms of scalability and a high transaction rate.

Different from the existing solutions that mainly address the issues on static trust policies, the DTEF includes the adaptive policy management feature as well as the real time trust evaluation process due to blockchain coupling with ABAC. This approach provides a more effective solution than previous models, which were unable to adapt to behavioral changes in IoT devices in real time. For example, TABI by Pathak et al. restrained from dynamic trust metrics and their update mechanism[15]. Conversely, the DTEF performs trust assessment continuously, which leads to anticipatory action regarding possible threats hence improving the security of the network and its strength.

The incorporation of the ECDSA increases the validity of the data in DTEF by enhancing the ABAC when compared to the RBAC used by Usman et al. (2024) in the framework and which it lacked adequate context-awareness[16]. The two in combination for access control enhance the security and the context aspects to give account to the risks of an unauthorized access.

Thus, this study sets itself unique as it presents an integrated end-to-end framework for real-time trust evaluation, adaptive policy management and anomaly detection based on blockchain technology. Previous works focused on the elaboration of aspects of the IoT security in isolation, but they did not offer a complete solution or scalable approach. However, the existing centralized and static trust evaluation systems do not fit the dynamic and heterogeneous nature of IoT applications which creates a gap.

4 Conclusion

The proposed DTEF offered a new approach to IoT security by integrating blockchain technology. Unlike previous studies, which often relied on static or centralized trust mechanisms, DTEF provides a decentralized and dynamic security framework. The experimental results demonstrated the framework's robustness, achieving over 95% accuracy in detecting malicious devices, reducing latency by 20%, and increasing throughput by 15% compared to existing models. These quantitative results validate the framework's superior scalability and effectiveness in dynamic IoT environments.

The primary strength of DTEF lies in its ability to adapt to evolving IoT device behaviors and environmental conditions, ensuring proactive security measures. Furthermore, the integration of ABAC with blockchain enhances access control precision and data integrity. However, the framework's dependence on blockchain may introduce higher resource consumption, and the evaluation was limited to simulated environments, leaving real-world applicability unexplored.

Future improvements could focus on integrating advanced machine learning algorithms for enhanced anomaly detection and incorporating innovative cryptographic techniques to further strengthen data security. Open questions remain regarding the scalability of the framework in extremely large IoT ecosystems and its performance under diverse real-world scenarios.

Further research should explore the application of DTEF in various domains, such as smart cities and industrial IoT, to validate its versatility and effectiveness in addressing real-world challenges. This study provides a foundational step toward a more secure and adaptive IoT ecosystem.

Data Availability Statement:

All the data is collected from the simulation reports of the software and tools used by the authors. Authors are working on implementing the same using real world data with appropriate permissions.

Author's Contributions:

Author 1: He Performed the Analysis the overall concept, writing and editing.

Author 2: He participated in the methodology, Conceptualization, Data collection and writing the study.

References

1) Huan NTY, Zukarnain ZA. A Survey on Addressing IoT Security Issues by Embedding Blockchain Technology Solutions: Review, Attacks, Current Trends, and Applications. *IEEE Access*. 2024;12:69765 –69782. Available from: https://doi.org/10.1109/ACCESS.2024.3378592.
2) Ahakonye LAC, Nwakanma CI, Kim DS. Tides of Blockchain in IoT Cybersecurity. *Sensors*. 2024;24(10):1–27. Available from: https://doi.org/10.3390/s24103111.
3) Nguyen T, Nguyen H, Gia TN. Exploring the integration of edge computing and blockchain IoT: Principles, architectures, security, and applications. *Journal of Network and Computer Applications*. 2024;226:1–24. Available from: https://doi.org/10.1016/j.jnca.2024.103884.
4) Hossain MI, Bhuiyan MKI, Sumon SA, Akther S, Hossain MI, Akther A. Enhancing Data Integrity and Traceability in Industry Cyber Physical Systems (ICPS) through Blockchain Technology: A Comprehensive Approach. *Advances in Artificial Intelligence and Machine Learning* . 2024;4(4):2883–2907. Available from: http://dx.doi.org/10.54364/AAIML.2024.44168.
5) Lmarri S, Aljughaiman A. Blockchain Technology for IoT Security and Trust: A Comprehensive SLR. *Sustainability* . 2024;16(23):1–35. Available from: https://doi.org/10.3390/su162310177.
6) Gopalan SH, Manikandan A, Dharani NP, Sujatha G. Enhancing IoT Security: A Blockchain-Based Mitigation Framework for Deauthentication Attacks. *International Journal of Networked and Distributed Computing*. 2024;12:237–249. Available from: https://doi.org/10.1007/s44227-024-00029-w.
7) Sharma KV, Sarada C, Vasavi M, and KA. Securing IoT Devices from DDoS Attacks through Blockchain and Multi-Code Trust Framework. In: International Conference on Renewable Energy, Green Computing and Sustainable Development (ICREGCSD 2023);vol. 472 of E3S Web Conf.. 2024;p. 1–12. Available from: https://doi.org/10.1051/e3sconf/202447203001.
8) Lakshmi GV, Vaishnavi P. A trusted security approach to detect and isolate routing attacks in mobile ad hoc networks. *Journal of Engineering Research*. 2024;12(3):379–386. Available from: https://doi.org/10.1016/j.jer.2023.100149.
9) Raj A, Prakash S. Privacy preservation of the internet of medical things using blockchain. *Health Services and Outcomes Research Methodology*. 2024;24(1):112–139. Available from: https://doi.org/10.1007/s10742-023-00306-1.
10) Ismail S, Nouman M, Dawoud DW, Reza H. Towards a lightweight security framework using blockchain and machine learning. *Blockchain: Research and Applications*. 2024;5(1):1–12. Available from: https://doi.org/10.1016/j.bcra.2023.100174.
11) Chesuh LN, Fernández-Diaz RA, Alija-Perez JM, Benavides-Cuellar C, Alaiz-Moreton H. Improve quality of service for the Internet of Things using blockchain & machine learning algorithms. *Internet of Things*. 2024;26:1–14. Available from: https://doi.org/10.1016/j.iot.2024.101123.
12) Taloba AI, Elhadad A, Rayan A, El-Aziz RMA, Salem M, Alzahrani AA, et al. A blockchain-based hybrid platform for multimedia data processing in IoT-Healthcare. *Alexandria Engineering Journal*. 2023;65:263–274. Available from: https://doi.org/10.1016/j.aej.2022.09.031.
13) Khan AA, Shaikh AA, Laghari AA. IoT with multimedia investigation: A secure process of digital forensics chain-of-custody using blockchain hyperledger sawtooth. *Arabian Journal for Science and Engineering*. 2023;48(8):10173–10188. Available from: http://dx.doi.org/10.1007/s13369-022-07555-1.
14) Sivaselvan N, Bhat KV, Rajarajan M, Das AK. A new scalable and secure access control scheme using blockchain technology for IoT. *IEEE Transactions on Network and Service Management*. 2023;20(3):2957 –2974. Available from: https://doi.org/10.1109/TNSM.2023.3246120.
15) Pathak A, Al-Anbagi I, Hamilton HJ. TABI: Trust-Based ABAC Mechanism for Edge-IoT Using Blockchain Technology. *IEEE Access* . 2023;11:36379–36398. Available from: https://doi.org/10.1109/ACCESS.2023.3265349.
16) Usman M, Sarfraz MS, Aftab MU, Habib U, Javed S. A Blockchain Based Scalable Domain Access Control Framework for Industrial Internet of Things. *IEEE Access*. 2024;12:56554 –56570. Available from: https://doi.org/10.1109/ACCESS.2024.3390842.
17) Mahalingam N, Sharma P. Secure monitoring model for smart agriculture using an optimized attribute-based access control centralized authority system. *Multimedia Tools and Applications*. 2024;83:44781–44798. Available from: https://doi.org/10.1007/s11042-023-17052-8.
18) Khordadpour P, and SA. Security and Privacy Enhancing in Blockchain-based IoT Environments via Anonym Auditing. 2024. Available from: https://doi.org/10.48550/arXiv.2403.01356.
19) Zaid T, Garai S. Emerging Trends in Cybersecurity: A Holistic View on Current Threats, Assessing Solutions, and Pioneering New Frontiers. *Blockchain in Healthcare Today*. 2024;7(1):1–14. Available from: https://doi.org/10.30953/bhty.v7.302.
20) Jimmy FNU. Cyber security Vulnerabilities and Remediation Through Cloud Security Tools. *Journal of Artificial Intelligence General science (JAIGS)*. 2024;2(1):129–171. Available from: https://doi.org/10.60087/jaigs.vol03.issue01.p233.
21) Liu H, Han D, Li D. Fabric-IoT: A blockchain-based access control system in IoT. *IEEE Access*. 2020;8:18207–18218. Available from: https://doi.org/10.1109/ACCESS.2020.2968492.