SafeID-LLM: Blockchain-Based Decentralized Identity Verification with LLM-Assisted Risk Analysis

Sinan Kasikci 78888
Zeynep Bakar 78889
Esila Bilen 78699
Rojin Karan 78765
Hasan Ceviz 78697

Abstract

Centralized identity management systems are vulnerable to security breaches and data misuse, necessitating a fundamental shift in the digital identity landscape. Decentralized Identity (DID) systems offer a solution by providing users with full control over their identity data. However, existing DID solutions often fall short in analyzing the complex and voluminous log data generated during identity verification processes. This paper proposes a novel architecture called **SafeID-LLM**. SafeID-LLM combines the security of blockchain-based decentralized identity verification with the log analysis and anomaly detection capabilities of Large Language Models (LLMs). The system continuously uses an LLM to analyze identity verification logs, user activities, and access requests. This analysis automatically identifies potential risks, inconsistencies, or suspicious activity patterns, providing system administrators with instant alerts. SafeID-LLM aims to significantly enhance the security and auditability of decentralized identity systems by offering both cryptographically secure verification and AI-assisted proactive risk auditing.

Keywords: Decentralized Identity (DID), Blockchain, Large Language Models (LLM), Risk Analysis, Anomaly Detection, Self-Sovereign Identity (SSI).

1. Introduction

1.1. Motivation and Problem Statement

Traditional identity management systems store identity data on centralized servers. This centralized structure makes the systems vulnerable to cyberattacks, data breaches, and misuse. Users lose control over their identities, while service providers bear a heavy burden of responsibility. The Decentralized Identity (DID) and Self-Sovereign Identity (SSI) paradigm has emerged as a solution to these problems [1] [2]. DID empowers users to manage and control their identities themselves, while blockchain technology ensures that these identities are recorded immutably and reliably [3].

Risks of Centralized Systems:

- **Single Point of Failure:** The concentration of all data in one place leads to catastrophic consequences in the event of a successful attack.
- **Data Privacy Violations:** Companies may use user data for their own benefit or be forced to share it with third parties under legal obligations.

- **Identity Fraud:** The risk of identity theft increases exponentially when centralized databases are compromised.

However, a new challenge arises with the adoption of DID systems: The effective auditing and analysis of identity verification and access logs. Although DIDs are cryptographically secure, processes such as verification attempts, access requests, and wallet interactions generate a large amount of unstructured log data that potentially indicates malicious or anomalous behavior. Existing DID systems lack an advanced mechanism to automatically analyze these logs and detect suspicious patterns. This makes it difficult for system administrators to manually detect anomalies and can lead to security vulnerabilities.

1.2. Proposed Solution: SafeID-LLM

This paper proposes the SafeID-LLM architecture to fill this gap in decentralized identity systems. SafeID-LLM combines the core security of DIDs with a Large Language Model (LLM) specifically trained for log analysis and anomaly detection.

Core Value of SafeID-LLM:

- **Enhanced Security:** Detects behavioral anomalies beyond cryptographic verification.
- **Proactive Auditing:** The LLM analyzes logs in near real-time, informing administrators about suspicious activities.
- **Transparency and Control:** System security is enhanced while users maintain control over their identities.

1.3. Paper Structure

The remainder of the paper is organized as follows: Section 2 presents a literature review on Decentralized Identity, Blockchain, and the security applications of LLMs. Section 3 analyzes the gap in existing DID systems and introduces the general architecture of SafeID-LLM. Section 4 describes the detailed methodology, components, and workflow of SafeID-LLM. Section 7 presents the proposed research components and metrics for system evaluation. Finally, Section 8 summarizes the conclusions and discusses future work.

2. Literature Review

2.1. Decentralized Identity (DID) and Self-Sovereign Identity (SSI)

A Decentralized Identifier (DID) is a new type of identifier standardized by the W3C, controlled by an entity (person, organization, device, etc.) [4]. DIDs form the foundation

of the Self-Sovereign Identity (SSI) philosophy. SSI advocates for individuals to have full control over their identity data and manage it without the need for a central authority [5].

The Role of Blockchain:

Blockchain serves as an ideal "trust anchor" for DID systems. DIDs typically point to a metadata structure called a DID Document, which is stored on a blockchain. The immutability and distributed nature of the blockchain guarantee the reliability and censorship resistance of identity information [6].

Verifiable Credentials (VC):

A critical component of the DID ecosystem, VCs are digital certificates issued by an Issuer and presented by a Holder. VCs are protected by cryptographic signatures and verified by a Verifier using the DID Document. SafeID-LLM's log analysis will focus on the logs of these VC presentation and verification processes.

2.2. Large Language Models (LLM) in Security Log Analysis

In recent years, Large Language Models (LLMs) have shown significant success in analyzing unstructured data, beyond their natural language processing capabilities. In the security domain, LLMs are seen as powerful tools, especially for log analysis and anomaly detection [7].

Advantages of LLMs:

- **Semantic Understanding:** Unlike traditional rule-based systems, LLMs can grasp the semantic meaning behind log entries. This provides the ability to detect complex and previously unseen attack patterns [8].
- **Summarization and Explanation:** LLMs can summarize detected anomalies in human-readable language and provide actionable explanations to administrators [9].
- **Contextual Analysis:** By contextually analyzing a series of log entries, they can detect suspicious activity chains evolving over time (e.g., repeated failed verification attempts from different geographies for the same user), beyond a single anomalous event [10].

LLM-Based Log Analysis Mechanisms:

1. **Log Parsing:** The LLM is used to convert log entries into structured event templates. This allows for the meaningful processing of large volumes of log data.
2. **Embedding and Anomaly Detection:** Parsed log events are mapped to a vector space using the LLM's embedding layer. Vectors deviating from normal log patterns are flagged as anomalies.

3. **Chain-of-Thought (CoT) Reasoning:** The LLM can combine a series of log events to construct an attack scenario and explain this scenario to administrators. This forms the basis of SafeID-LLM's risk explanation capability.

This literature review solidifies the two fundamental pillars of SafeID-LLM: the reliability of blockchain for decentralized identity and the analytical power of LLMs for proactive security auditing.

3. Current State Analysis and Proposed Solution

3.1. Gap Analysis

Existing DID systems focus on cryptographic security for identity verification. However, these systems have the following critical gaps:

1. **Lack of Automated Log Analysis:** DID verification processes generate logs containing critical information such as successful and failed verification attempts, access permissions, and wallet interactions. These logs are vital for understanding whether the system is being misused. However, current systems lack a mechanism to automatically analyze these logs, detect suspicious patterns, and alert administrators.

2. **Behavioral Anomaly Detection:** Even a cryptographically valid verification can pose a security risk if there are anomalies in user behavior (e.g., access from unusual times or locations). Traditional DID systems are insufficient in detecting such behavioral anomalies.

As shown in this diagram, SafeID-LLM proposes to fill the Lack of Automated Log Analysis gap in existing DID systems with the LLM Risk Analysis component.

3.2. SafeID-LLM Project Idea

SafeID-LLM is an integrated platform that combines decentralized identity management with AI-assisted security auditing.

Core Components:

- **DID Generation & Blockchain:** The layer where users create their Decentralized Identifiers (DIDs) and the core data of these DIDs (DID Document) is immutably stored on the blockchain.

- **Identity Verification Service:** The service that performs identity verification using cryptographic proofs. All operations of this service generate detailed logs.

- **Log Storage:** A storage unit where identity verification and user activity logs are securely and auditable stored.

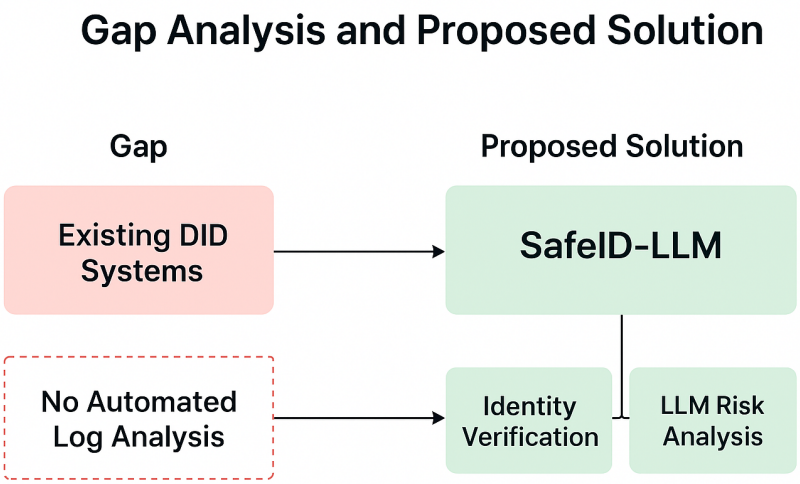

Figure 1: Gap Analysis and Proposed Solution

- **LLM Risk Analysis Engine:** The Large Language Model-based engine that analyzes stored logs, user activities, and access requests, performs anomaly detection, and generates risk scores.
- **Admin Dashboard:** The interface where administrators manage identities and visualize the risks, inconsistencies, and suggestions flagged by the LLM.

4. SafeID-LLM Architecture and Methodology

4.1. SafeID-LLM Architectural Diagram

The SafeID-LLM architecture clearly separates the decentralized identity verification process and the LLM-assisted risk analysis loop.

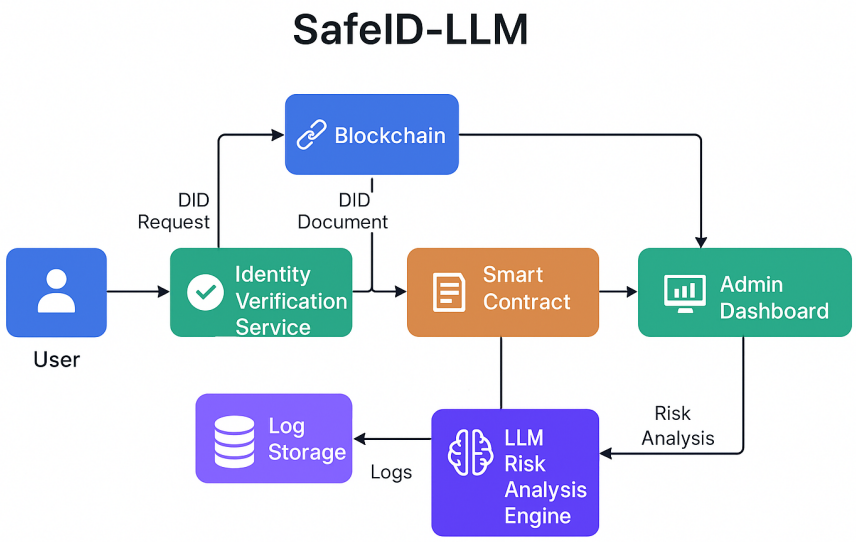

Figure 2: SafeID-LLM Architectural Diagram

Function of Architectural Components:

Table 1: SafeID-LLM Components in the Architecture

Component	Function	Technical Implementation Suggestion
User	Submits a DID creation request and initiates identity verification processes.	DID Wallet Application (e.g., Hyperledger Aries)
IdentityVerificationService	Receives the user's DID request, verifies cryptographic proofs, and creates the DID Document. Logs all operations.	RESTful API Service (e.g., Node.js/Python)
Blockchain	Immutably records the DID Document and triggers the Smart Contract.	Ethereum, Polygon, or Hyperledger Fabric
Smart Contract	Manages DID creation/update operations and instructs the LLM Risk Analysis Engine to process logs.	Solidity-based smart contract
Log Storage	Stores all logs (successful/failed verification, access attempts) generated by the Identity Verification Service.	Decentralized Storage (e.g., IPFS) or Secure Centralized Log Management (e.g., ELK Stack)
LLM Risk Analysis Engine	Pulls data from Log Storage, performs semantic and behavioral anomaly detection using the LLM, and generates risk scores	Fine-tuned GPT/BERT model

4.2. SafeID-LLM Workflow (Flowchart)

The operational flow of the system covers the DID creation, verification, and risk analysis cycle.

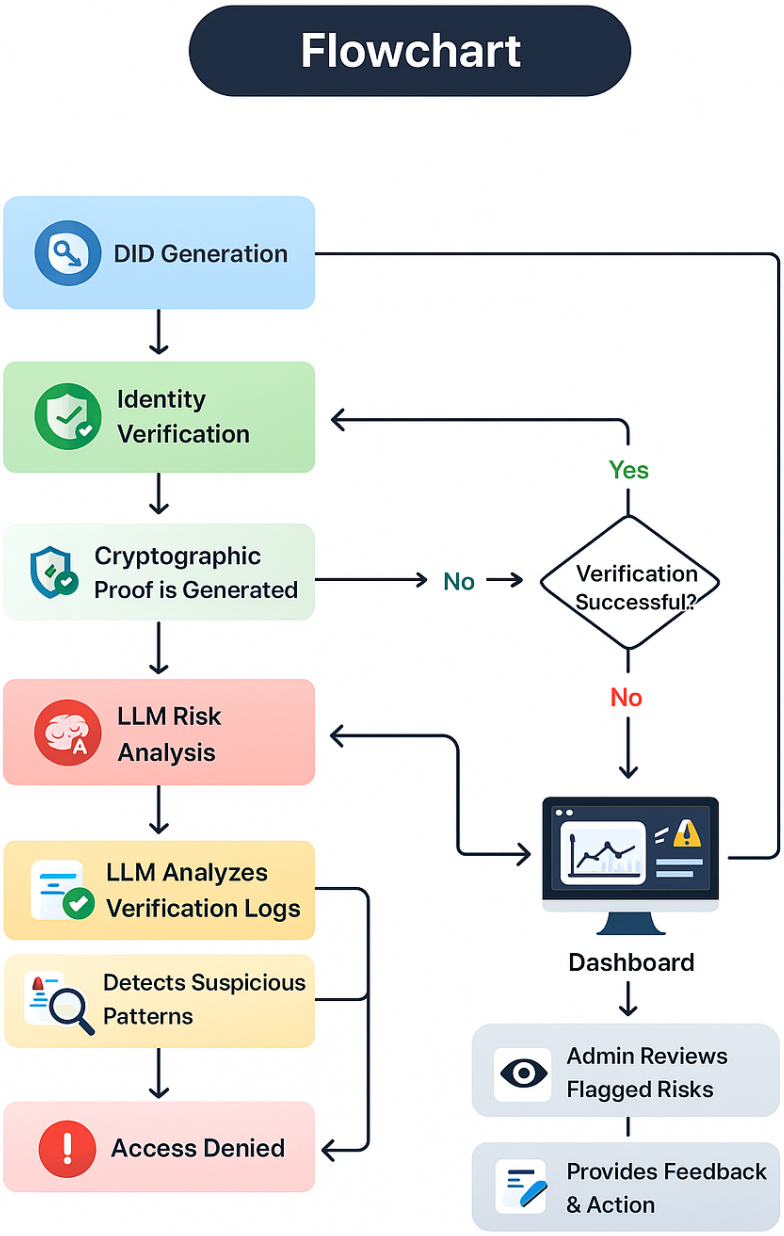

Figure 3: SafeID-LLM Workflow (Flowchart)

Detailed Analysis of Workflow Steps:

1. **DID Generation:** The user initiates the decentralized identity creation process via a wallet application. This request is forwarded to the Identity Verification Service.

2. **Identity Verification:** The service verifies the user's identity using cryptographic

proofs (e.g., may request proof of an email or phone number). If verification is successful, a cryptographic key pair is generated.

3. **Cryptographic Proof is Generated:** The created DID Document is recorded on the Blockchain via the Smart Contract. This process immutably links the DID's public key and service endpoints.

4. **Logging and LLM Risk Analysis Trigger:** The verification process (successful or failed) is recorded as a detailed log entry in Log Storage. The Smart Contract triggers the LLM Risk Analysis Engine following this log entry.

5. **LLM Analyzes Verification Logs & Detects Suspicious Patterns:** The LLM processes the new log entry and the relevant DID's past logs (for contextual analysis). The LLM detects semantic inconsistencies in the log texts and behavioral anomalies (e.g., speed, geography, number of attempts).

6. **Dashboard & Admin Reviews:** The risk score and explanation generated by the LLM are sent to the Admin Dashboard. Administrators review these flags based on priority.

7. **Provides Feedback & Action:** The administrator takes action based on the LLM's suggestion. These actions may include temporarily suspending the DID, requesting additional verification, or issuing an Access Denied decision.

4.3. Data Model (ER Diagram)

The core data entities of SafeID-LLM and their relationships support the system's auditing and management capabilities.

Entities and Relationships:

- **User:** The main entity that creates and uses the decentralized identity. Connected to DID by the `generates` relationship.
- **DID (Decentralized Identifier):** The cryptographic identifier created by the user and recorded on the blockchain. All verification and access operations related to the DID are reflected on the Dashboard via the `logged` relationship.
- **Dashboard (Management Panel):** The interface that manages User and DID data, and also displays the logged data from the LLM.

5. Implementation and Technical Details

5.1. DID Generation and Blockchain Integration

SafeID-LLM suggests using a Permissioned Blockchain or a Layer 2 solution to record DIDs. This increases transaction speed while maintaining network reliability.

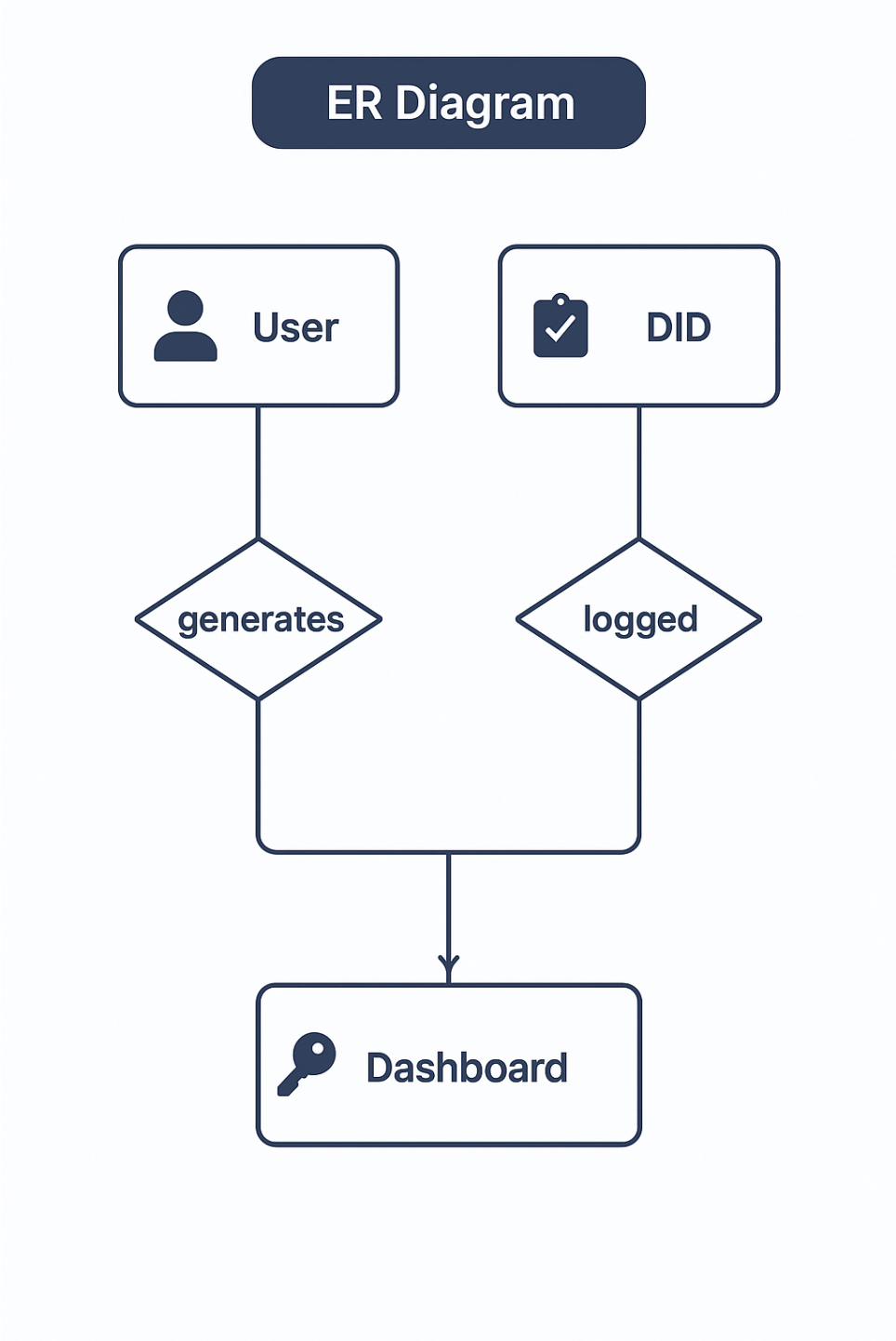

Figure 4: ER Diagram

Technical Steps:

1. **DID Method Selection:** A W3C-compliant DID method (e.g., `did:ethr` or `did:ion`) is chosen. Layer 2 solutions like `did:ion` offer the advantage of high transaction volume and low cost.

2. **Smart Contract Development:** A Smart Contract is used to store the hash and reference of the DID Document (public key, service endpoints). This contract

manages the creation, updating, and revocation of the DID.

3. **Cryptographic Verification (VC/VP):** When a user wants to access a service, they present a Verifiable Presentation (VP) using the private key associated with their DID. The Identity Verification Service checks the validity of this VP and whether it matches the DID Document on the blockchain. This process is a critical point for logging.

5.2. LLM Risk Analysis Engine

The LLM Risk Analysis Engine is the innovative core of SafeID-LLM.

Input Data and Pre-processing:

- **Log Data:** Timestamp, source IP, DID, transaction type (verification successful/failed), type of VC used.
- **Contextual Data:** Geographical location, type of device used, number of previous successful/failed attempts.
- **Pre-processing:** Before being fed to the LLM, log entries are anonymized (e.g., the last octets of IP addresses are masked) and converted into a standard JSON format.

LLM Tasks and Prompt Engineering:

1. **Log Normalization and Semantic Analysis:** The LLM standardizes unstructured log texts and analyzes the semantic content of each log event (e.g., "Key mismatch" or "Expired VC").
2. **Behavioral Anomaly Detection:** The LLM detects unusual access patterns (e.g., verification attempts from 3 different IPs within an hour) by comparing them with the specific DID's past activities (contextual analysis).
3. **Risk Scoring and Explanation Generation:** The LLM assigns a risk score (0-100) based on the detected anomalies and generates a short, actionable text explaining the reason for this score (e.g., "High Risk: Detected failed verification attempts from 3 different countries in the last 5 minutes.").

Prompt Example (Instruction Given to the LLM):

> "Analyze the following log entries. The logs show verification attempts for DID: [DID_ID] in the last 15 minutes. Normal behavior is successful attempts coming from the [Normal_IP_Range] IP range. If you detect an anomalous pattern, inconsistency, or suspicious activity, return a risk score between 0-100 and an explanation not exceeding 20 words."

5.3. Admin Dashboard Development

The Admin Dashboard is designed to enable administrators to effectively use the LLM's analysis results.

Dashboard Features:

- **Risk Prioritization:** Sorting events based on the risk score assigned by the LLM.
- **Contextual Visualization:** Visualization of the suspicious DID's past activities (geographical location map, time series graph).
- **Action Modules:** The ability for the administrator to suspend the DID, request additional verification, or flag the LLM's alert as a "false positive" with a single click.

6. Technology Stack

The technology stack used for SafeID-LLM is summarized below.

Category	Technology	Justification
Programming Language	Python	Offers the widest tool support for LLM and RAG development.
Agentic Framework	LangGraph	Manages complex task flows with controllable agent structures.
LLM API	OpenAI GPT-4o	Provides high-accuracy LLM capabilities for risk analysis and text interpretation.
Vector Database	Pinecone	Enhances RAG performance with fast and consistent embedding searches.
RAG Pipeline Orchestration	LangChain	Unifies retrieval and LLM processes within a single modular pipeline.

Figure 5: Technology Stack Overview

7. Evaluation and Research Components

The following research components and metrics are proposed to prove the effectiveness of SafeID-LLM:

7.1. Accuracy and Relevance

Table 2: Accuracy and Relevance Metrics

Metric	Description	Target
IdentityVerificationAccuracy	The rate at which the cryptographic verification mechanism correctly distinguishes between valid and invalid credentials.	100% (Cryptographic assurance)
LLM Risk Flag Precision	The rate at which events flagged as "risky" by the LLM are genuinely anomalous or malicious.	High (Minimize false positives)
LLM Risk Flag Recall	The rate at which all genuinely anomalous events are detected by the LLM.	High (Avoid missing real risks)
Explanation Relevance	How much the LLM's generated risk explanations help the administrator understand the situation and take action (Measured by administrator feedback).	High

Research Method: To evaluate the LLM's performance, a dataset of real-world DID logs (containing synthetically generated anomalies) will be used. A comparative analysis will be conducted with traditional rule-based anomaly detection systems.

7.2. Performance

Table 3: Performance Metrics

Metric	Description	Target
Blockchain Transaction Speed	The time (latency) it takes for the DID Document to be recorded on the blockchain.	Low (Fast recording)
LLM Inference Latency	The time between sending logs to the LLM and receiving the risk score.	Near real-time (Milliseconds)
System Scalability	The number of identity verification requests that can be processed per second (TPS).	High

Research Method: The system's response time and resource consumption will be measured at different load levels (e.g., 50, 100, 200 verification requests per second). The LLM's inference latency will be correlated with the model size and the hardware used(GPU/CPU).

7.3. User Trust and Usability

- **Admin Dashboard Usability:** How easily administrators can understand and manage the risks flagged by the LLM (Task Completion Time, Error Rate).
- **User Trust:** The trust users place in the system's transparency and control, despite their identities being audited by an LLM (Measured by surveys).

Research Method: Focus group studies and surveys will be conducted with system administrators and end-users. The clarity and actionability of the LLM's risk explanations will be the main focus of usability tests.

8. Conclusion and Future Work

SafeID-LLM presents a hybrid architecture with the potential to transform the security and auditability of decentralized identity systems. By combining the immutability guarantee of the blockchain with the semantic log analysis capability of LLMs, it closes the critical gap (lack of automated risk auditing) in existing DID solutions.

Main Contributions:

- Integration of AI-assisted proactive risk analysis into blockchain-based DID systems.

- Use of LLM to detect semantic and behavioral anomalies in log data.
- An auditing panel that provides administrators with actionable risk explanations.

Future Work:

- Expanding the LLM to adapt to different DID methods (e.g., `did:web`, `did:peer`).
- Training the LLM's risk scoring model using Federated Learning across log data from different organizations, while preserving data privacy.
- Adding an automation layer that combines the risk explanations generated by the LLM with the ability to take automatic action (e.g., temporarily blocking requests from a specific IP).

References

[1] W3C DID Specification. (Access address will be added here)

[2] Self-Sovereign Identity Principles. (Access address will be added here)

[3] Blockchain's Role in DID. (Access address will be added here)

[4] W3C Decentralized Identifiers (DIDs) v1.0. (Access address will be added here)

[5] What is Decentralized Identity. (Access address will be added here)

[6] Decentralized identity management (DID) using blockchain. (Access address will be added here)

[7] LogLLM: Log-based Anomaly Detection Using Large Language Models. (Access address will be added here)

[8] Review AIOps for log anomaly detection in the era of LLMs. (Access address will be added here)

[9] CLogLLM: A Large Language Model Enabled Approach to Log Analysis. (Access address will be added here)

[10] Boosting Your Anomaly Detection With LLMs. (Access address will be added here)

[11] LLMs are helpful —LogLMs are better for incident identification. (Access address will be added here)

[12] Decentralized identity: Where did it come from and where is it going? (Access address will be added here)

[13] SS-DID: A secure and scalable Web3 decentralized identity utilizing multilayer sharding blockchain. (Access address will be added here)

[14] Decentralized identity (DID): new technology adoption and diffusion in South Korea. (Access address will be added here)

[15] Llmelog: An approach for anomaly detection based on llm-enriched log events. (Access address will be added here)

[16] Log anomaly detection by leveraging LLM-Based parsing and embedding with attention mechanism. (Access address will be added here)

[17] AnomalyGen: An Automated Semantic Log Sequence Generation Framework with LLM for Anomaly Detection. (Access address will be added here)

[18] Web Technologies for Decentralised Identity. (Access address will be added here)

[19] Towards Self-Sovereign Identity. (Access address will be added here)

[20] W3C Verifiable Credentials Data Model. (Access address will be added here)

[21] The Role of Smart Contracts in DID Management. (Access address will be added here)

[22] Federated Learning for Privacy-Preserving Anomaly Detection. (Access address will be added here)

[23] Log Anomaly Detection with LLM-based Zero-Shot Learning. (Access address will be added here)

[24] Performance Benchmarking of DID Methods. (Access address will be added here)

[25] Usability and Trust in Decentralized Identity Systems. (Access address will be added here)

[26] Security Auditing in SSI Ecosystems. (Access address will be added here)

[27] The Impact of LLM Latency on Real-Time Security Systems. (Access address will be added here)

[28] DID Revocation Mechanisms and Security Implications. (Access address will be added here)

[29] Privacy-Preserving Log Analysis Techniques. (Access address will be added here)

[30] Blockchain Scalability Solutions for Identity Management. (Access address will be added here)