National University of Computer and Emerging Sciences, Lahore

Secure Data Exchange System for Ad-hoc Networks Using Blockchain-Enhanced DSR

Ahmed Nasir 22L-6644 BSCS

Aanish Waseem 22L-6887 BSCS

Muhammad Salman Amir 22L-6830 BSCS

Supervisor: Mr. Naveed

Co-Supervisor: Dr. Rana Asif ur Rehman

Final Year Project

December 4, 2025

Anti-Plagiarism Declaration

This is to declare that the above publication was produced under the:

Title: Secure Data Exchange System for Ad-hoc Networks Using Blockchain-Enhanced DSR

is the sole contribution of the author(s), and no part hereof has been reproduced as it is the basis (cut and paste) that can be considered Plagiarism. All referenced parts have been used to argue the idea and cited properly. I/We will be responsible and liable for any consequence if a violation of this declaration is determined.

Date: 17-10-2025

Name: Ahmed Nasir

Signature: 

Name: Aanish Waseem

Signature: 

Name: Muhammad Salman Amir

Signature: 

Author's Declaration

This states the Authors' declaration that the work presented in the report is their own and has not been submitted/presented previously to any other institution or organization.

Abstract

Low-Mobility Personal Area Ad hoc Networks make use of wireless in short-range communications of sensitive data. Classical DSR is vulnerable to Sybil, Blackhole, Wormhole, and Replay attacks and requires a decentralized lightweight security. This paper proposes to deploy the blockchain-enhanced DSR protocol with a small overhead. Bayesian trust diaries make decisions based on the past experience. PKCertChain authenticates the devices using lightweight Proof-of-Work and density-aware quorum. The RouteLog Chain monitors the routes and offers a secure routing according to a multi-metric Dijkstra algorithm. The protocol is experimented to be lightweight, adaptive and secure.

Executive Summary

The project titled Secure Data Exchange System for Ad-hoc Networks Using Blockchain-Enhanced DSR offers a software approach to enable secure peer-to-peer communications in a decentralized personal area network. In this way, multiple nodes in such networks communicate directly with each other without using any centralized infrastructure. These are subject to Blackhole, Wormhole, Sybil, and Replay based Attacks affecting the Consistency of communication, Data Path, and Trust between Nodes of the network. To tackle these challenges, the system uses a secure routing protocol, Advanced Blockchain Dynamic Source Routing (ABCD) [1] and a decentralized PKI for authentication. Digital Ids Each node in the system are assigned a digital ID along with a set of cryptographic keys which can be used to prove its identity and keep the data exchanges secure. A Bayesian Trust Diary keeps appraising nodes trust worthiness from previous interactions. Malicious nodes or those that drop packets are assigned low trust values while trustworthy nodes are given higher values. All routes and trust management information are securely stored inside blockchain making it immutable and tampering proof. The system, called Linux Data Exchange (LDX), is a Linux application. To conclude, through this paper we have presented a full-fledged, strong software based product utilizing blockchain, secure routing, PKI based authentication and bayesian trust evaluation to achieve a secure data exchange in personal area ad hoc network. It is engineered to offer a real-world, scalable, and reliable approach for decentralized, peer-to-peer communication.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

Secure and efficient communication in personal area ad-hoc networks is a critical challenge in modern networking applications. The DSR protocol, widely used for routing in ad-hoc networks, faces several security vulnerabilities.These threats include blackhole, wormhole and Sybil attacks. This causes the need for a robust solution to enhance the security of DSR protocol.

To overcome these issues, our project focuses on securing the DSR routing by integrating blockchain technology to provide transparency, trust, and resilience against common routing attacks. By leveraging blockchain's decentralized and tamper-proof characteristics, our proposed solution aims to detect and prevent malicious activities within the network, ensuring secure routing decisions. By combining DSR with blockchain, along with decentralized PKI for authentication and a Bayesian Trust Diary for trust evaluation, the system ensures that nodes are verified and trustworthy before participating in data exchange.

The proposed solution is implemented as a Linux-based Data Exchange application, providing a practical, deployable product for secure peer-to-peer communication. The integration of blockchain with DSR enhances network security while maintaining efficient routing performance, offering a robust framework for secure data exchange in personal area ad hoc networks.

1.1 Purpose of this Document

This document describes our FYP, Secure Data Exchange System for Ad-hoc Networks Using Blockchain-Enhanced DSR in detail. This paper describes the ideas, design, implementation and features of a secure DSR-based routing protocol. The project combines blockchain security, decentralized authentication through PKI, and Bayesian trust evaluation to secure and trustably exchange data in a P2P among Network nodes. The document is intended to show how the system offers a binary, software solution, covering the entire stack, for providing secure communications in personal area ad hoc networks. It emphasize the feature of the product that detect and stop Blackhole, Wormhole, Sybil and Replay attack on network with better and stable routing performances.

1.2 Intended Audience

This document is intended for the FYP panel and our supervisor to analyze and evaluate our approach. Moreover, developers can also find it helpful for exploring the technical aspects, such as node ID generation, blockchain implementation, decentralized PKI authentication, and trust evaluation within DSR-based routing system.

1.3 Definitions, Acronyms, and Abbreviations

Important definitions, acronyms, and abbreviations used in this chapter are as follow:

- **FYP:** Final Year Project
- **DSR:** Dynamic Source Routing
- **PANETs**: Personal Ad Hoc Networks
- **MANETs**: Mobile Ad Hoc Networks
- **VANETs**: Vehicular Ad Hoc Networks
- **ABCD**: Advanced Blockchain Dynamic Source Routing
- **SHA-256**: Secure Hash Algorithm 256-bit
- **XOR**: Exclusive OR
- **Private Key / Secret Key** : A confidential Cryptographic key use for signatures and decryption
- **Digital Signatures**: A cryptographic technique for validating message integrity by verifying signer's private key.
- **Digital Certificates**: An electronic credential to ensure the credibility of the Vehicle, ensured by trusted authority here using Blockchain for it.
- **ECC** - Elliptic Curve Cryptography: A public-key encryption method suitable for low-resource devices.
- **ECDSA** - Elliptic Curve Digital Signature Algorithm: An algorithm based on ECC, use to verify the authenticity of data and entities.
- **PKI** - Public Key Infrastructure - Architecture for managing public-key encryption and certificates
- **Ad-hoc** : A decentralized network where nodes communicate directly
- **PK** - Public Key: Openly shared component of Public key- Private Key pair.
- **Job Queue / Scheduler** : A scheduling structure within a Operating System to manage processes and tasks.
- **Thread** : A smallest unit of task that can be scheduled / processed and useful for multi-tasking.
- **Buffer** : A temporary data storage to manage data transfer between processes, threads and hardware.

- **RTOS** - Real Time Operating System: Operating System for time-critical applications with strict timing constraints for predictable and deterministic execution
- **Sybil attack**: A node generates multiple fake nodes to disrupt routing.
- **Wormhole Attack** : Two or more nodes tunnel packet across network and mislead routing protocols
- **Blackhole Attack** : A node that absorbs packet and never transmit critical information. **AOMDV**: Ad hoc On-demand Multipath Distance Vector is an extension of the AODV routing protocol used in MANETs.

  OLSR: Optimized Link State Routing is a proactive routing protocol for ad-hoc networks that maintains up-to-date routes to all nodes by periodically exchanging topology information.

  ANFIS: Artificial Neuro-Fuzzy Inference System are Neural networks that learn patterns from data, while fuzzy logic handles approximate reasoning. Together, ANFIS enables adaptive decision-making in dynamic networks.

  Block-DSD: Distributed Blockchain-Assisted Secure Data Aggregation) is a blockchain-enabled framework that uses zone-based clustering, AI-driven cluster head selection, and lightweight cryptography.

  IPFS: Interplanetary File System is a decentralized storage and file-sharing system that uses content-addressing instead of traditional URLs.

  DPBFT: Delegated Practical Byzantine Fault Tolerance is a consensus mechanism that enhances the traditional Practical Byzantine Fault Tolerance (PBFT) algorithm by delegating validation tasks to selected nodes.

  BHO: Binary Hawks Optimization is a metaheuristic optimization algorithm inspired by the cooperative hunting strategy of hawks. In binary form, it is used for feature selection and routing optimization in network.

1.4 Conclusion

The first chapter of this research introduces the concept of ”Secure Data Exchange System for Ad-hoc Networks Using Blockchain-Enhanced DSR” as an enhanced learning platform, its intended audience, and the definitions and abbreviations of complex terms. The second chapter dives deeper into the vision of the project by discussing the need for secure DSR-based routing, the goals and objectives, and the project scope. Moreover, the constraints of this project, including software-only implementation, and the potential for future are also key points in this chapter.

Chapter 2 Project Vision

In Chapter 2 we go further and deeper with the key concepts of the project. This chapter describes the problem of security in DSR-based routing in PANETs and network security to be addressed in this project. This section gives an overview of the objectives of project, scope and limitations. Hence it enables the reader to appreciate what the study is trying to achieve and why.

2.1 Problem Domain Overview

DSR is the preferred protocol for data delivery in ad hoc networks as is but it has security holes. PANETs depend on nodes exchanging data directly in a peer-to-peer environment, however traditional DSR and other routing algorithms are almost entirely concerned with connectivity and path selection and are not security aware. Therefore, PANETs are vulnerable to attacks such as Blackhole, Wormhole, and Sybil attacks, which may disrupt communication, tamper with routing paths, and undermine trust among nodes. The above security vulnerabilities suggest that there is a big challenge to design a unified solution that combines secure routing, node authentication, and trust evaluation to have the guarantee of reliable and trustworthy data exchange in software based AODV.

2.2 Problem Statement

DSR in PANETs is not secure by default and is susceptible to Blackhole, Wormhole and Sybil attacks. This leads to nodes experiencing communication disruption, route manipulation and trust degradation. There is a high demand for a secure routing scheme to guarantee the route establishment and information scattering with dependable, authenticated and trustworthy services in a decentralized, software-only platform.

2.3 Problem Elaboration

This section of the chapter identifies different problems that the proposed system addresses in PANETs

2.3.1 Lack of security

Routing protocols such as DSR does not treat security threats including Blackhole, Wormhole and Sybil attacks. Therefore, data may be stolen, and the communication among vehicles can not be carried out. This one aims to make communication secure by introducing blockchain technology, encryption and improved route management for nodes.

2.3.2 Inefficient Routing

Optimal routing is necessary for timely and dependable information flow. The above framework is a modification of DSR using Dijkstra's algorithm and computes the best path (in terms of hops) at the route discovery phase, thus enhancing the quality of service in term of data delivery and delay.

2.3.3 Trust Management

In decentralized networks, nodes depend on cooperation, but a malicious node can send false information. A Bayesian Trust Diary based approach monitors the behavior of nodes to be trustworthy and resist the attacks on the network.

2.3.4 Lightweight Consensus and Scalability

Conventional blockchain consensus protocols are CPU-intensive and add latency to the process. The procedure uses lightweight consensus mechanisms like

- Lightweight Proof of Work that relies on a small-range hash challenge and is used to mitigate Sybil attacks.
- Quorum-Based proof where already established nodes vouch for the registration of new nodes.
- A multi-metric Dijkstra-based routing for efficient path selection

2.3.5 Real Time Scheduling and Data Freshness

Android based systems are not time-critical and deterministic. Our scheduling proposes deterministic results by prioritizing urgency. Our buffer mechanism is O(1) push/pop operation and use CPU instructions to find highest priority task instead of heap sort O(n log n)

2.3.6 Traditional PKI

Conventional PKI-based systems require the presence of a Certificate Authority, which is incompatible with the characteristics of PANETs because it would introduce a single point of failure, additional latency and demand for a static topology. The work addresses the challenge by introducing the notion of PKCertChain i.e. a blockchain for the certification issuance, renewal, public key and public key management.

2.4 Goals and Objectives

The primary goals and objectives of this project are:

- Ensure data integrity and trust among vehicles using blockchain and encryption.
- Optimize data exchange and routing with DSR enhances by Dijkstra algorithm for reduced latency and efficient delivery.
- Develop a software-system capable of secure, peer-to-peer communication in PANETs.

2.5 Project Scope

The scope of this project is to design and develop a Blockchain-Enabled Data Exchange System for secure communication in PANETs. The system focuses on ensuring trust, authentication, and protection against malicious nodes, rather than optimizing mobility, routing efficiency, or large-scale network communication.

The system will allow nodes to

- Communicate securely using blockchain based trust management
- Ensure data integrity and authentication through encryption and decentralized PKI.
- Evaluate node trustworthiness using a Bayesian Trust Diary.

The project deliverables will include:

- A detailed report containing the design, implementation, results and testing process.
- The source code for all individual modules, including blockchain, PKI and trust evaluation, and how they interact.
- Software demonstrations of secure data exchange between Linux-based nodes.
- Attack and defense scenarios showing the system's effectiveness against Blackhole, Wormhole and Sybil attacks.

The project does not include the following:

- Mobility or vehicle-based communication scenarios
- Testing on physical hardware
- Optimization for high-speed or large-scale routing
- Integration with third-party platforms or cloud services

2.6 Sustainable Development Goal (SDG)

The Architecture and Technology Roadmap of the United Nations Sustainable Development Goal 9: Industry, Innovation, and Infrastructure promotes the development of resilient, inclusive and sustainable technological solutions. The proposed scheme is the first one enabling the formation of a secure and trustworthy data exchange system for PANETs by means of blockchain, bringing security, trust and reliability to the space of decentralized communication networks. This innovation leads to building robust and secure software-based communication frameworks which can enable applications in the future that need to exchange data between peers that are trusted, employ strong cybersecurity measures, and leverage a decentralized infrastructure applicable to smart digital networks in industrial, educational, and other areas.

Figure 2.1: SDG 9. Industry, Innovation and Infrastructure

2.7 Constraints

The project also has its limitations. It is all software and no hardware. The system is only tailored to small PANETs and hence, it is unable to support large networks. It is security and trust (authentication) centric, and not mobility, fast routing, or real-time network performance oriented. While it demonstrates secure data exchange and rudimentary attack mitigation, it does not connect to third party platforms or cloud services. The suggested implementation enables a practical secure communication system but not one with potentially very high scalability or correspondingly large network simulations.

2.8 Conclusion

To summarize, this chapter described the vision and logic of the proposed work. In particular it underlines the security issues of the PANETs, and some of its attacks (Blackhole, Wormhole, Sybil and Replay). The innovations of the project are that it tackles these problems through the integration of blockchain, encryption, decentralized PKI, and secure routing to realize trusted peer-to-peer data trade. Further improvements include a Bayesian Trust Management model to assess node cooperation, and lightweight consensus mechanisms (i.e., PoW and quorum-based proofs) for a secured and efficient blockchain infrastructure. This chapter serves as a solid backbone for a real software product that guarantees security, reliability and authentication, in a decentralized personal area network.

Chapter 3 Literature Review / Related Work

This chapter includes the detailed literature review of studies relevant to proposed security approach in communication protocol. It first looks at the basic studies that inspired this work and then discusses recent research on lightweight cryptography, trust management, and blockchain systems.

3.1 Detailed Literature Review

The following sections provide each review material in detail including its summary, the critical analysis and its relevancy to our work.

3.1.1 ABCD: advanced blockchain DSR algorithm for MANET to mitigate the different security threats

This foundational paper introduces the ABCD protocol [1]. It is designed to make routing in MANETs more secure. The protocol combines DSR algorithm with blockchain to store route and node information safely. A key feature is the use of homomorphic encryption, which lets nodes process data without decrypting it, keeping the information private. The system builds trust by giving each node a reliability score based on its good and bad actions. Routes that include less reliable nodes get a higher cost, so the system avoids them.

The main strength of the ABCD protocol [1] is its smart use of homomorphic encryption and a blockchain-based reputation system. However, it is not easy to use in real-time vehicular networks. Homomorphic encryption needs a lot of computing power, which makes it hard to run on small, low power devices. Also, the use of one single blockchain for all network tasks slows down performance and makes it difficult to scale.

This paper is the main reference for our research. Our project improves the ABCD protocol by fixing its main problems. We replace homomorphic encryption with a simpler and faster PKI system. the basic reliability score is upgraded to an adaptive Bayesian Trust Diary for better trust management. We also replace the single blockchain with a multilayer design that includes an Adaptive PKI Blockchain, a Routing Ledger, and a Meta Dispatcher to make the system faster and easier to scale.

3.1.2 A scalable blockchain based trust management in VANET routing protocol

This study presents a two-level layer to address. At the first level, each node evaluates its neighbors' behavior using response time, packet delivery ratio and various consistency metrics to calculate a local trust score. At the second level, RSUs act as blockchain validators, which maintains blacklist of malicious nodes. The routing protocol used is AODV [2].

One of the strengths of the study includes the two level trust evaluation system. The RSU serves as blockchain validators, which reduces the computational overhead on individual nodes. However, the framework way of determining trust threshold is quite strict. Nodes may face difficulty in entering the network. Which makes the framework less feasible where public or random nodes may want to participate in the network.

This study is quite relevant to the proposed study in terms of secure protocol design. The study relied on RSUs as validators for trust scores, the proposed study validates based on Bayesian trust diaries. Moreover, the proposed work extends the study by embedding PKI blockchain for node registration, renewal which also serves as confidentiality.

3.1.3 Traffic Prevention and Security Enhancement in VANET

The paper's summary integrates blockchain for a trusted routing environment, while deploying deep learning models for proactive traffic prediction and anomaly detection. A BHO algorithm is used, which selects the most stable and secure communication paths. This approach leverages the IPFS for decentralized data storage and a DPBFT consensus mechanism to validate transactions [3].

Critically, the strength of this research lies in its multi-faceted approach. Using blockchain to address secure routing while deploying deep learning to address Traffic problems. However, its primary weakness may be its complexity and potential computational overhead. The blockchain consensus, deep learning frameworks for a vehicular node and optimization algorithms could be resource-intensive.

The paper is quite relevant to the proposed core concept of using a blockchain-inspired architecture to establish trust and secure routing. The paper goes one step beyond. While the proposed idea focus is on the data transmission securely, the paper introduces a deep learning component for traffic intelligence. In fact, it highlights a potential research gap in the proposed idea, suggesting how data transmitted can be leveraged for traffic mitigation.

3.1.4 Lightweight Cryptography and IDS for Edge Networks

This research presents a secure and efficient group communication method for VANETs. It uses the LED, a lightweight encryption system. To work well on low-power hardware, the authors use a 64-bit key version. The key management is decentralized to protect privacy. Vehicles use fake identities to ask a group manager for a key and then create their own private keys. The system also changes keys regularly, keeping group communication secure and protecting each vehicle's identity [4].

The main strength of this work is its practical and efficient design. The use of a lightweight cipher and pseudo identities work well for vehicle edge devices. However, it depends on a group manager, which is a weakness. If the group manager fails or is attacked, the security of whole group can be at risk.

The idea of lightweight and hardware-aware security in this paper matches the goals of our protocol. Our system uses asymmetric PKI for strong authentication but still focuses on efficiency and privacy. The use of pseudo identities can also be added to our Adaptive PKI Blockchain to make nodes more anonymous.

3.1.5 Distributed blockchain assisted secure data aggregation scheme for risk-aware zone-based MANET

The study introduces a security framework named Block-DSD designed for MANETs, claimed to work in high-stakes environments like disaster management. The core arhitecture is Zone-based Clustering Approach that segments the network into zones to reduce overhead. Within each zone, an optimal Cluster Head is selected using an AI-driven ANFIS that evaluates nodes based on multiple criteria. The actual process of collecting data is secured using a combination of a Two-Step Secure method and lightweight ECC. Finally, to ensure data is transmitted efficiently between zones, an Improved Elephant Herd Optimization algorithm finds the optimal route. The blockchain serves as the decentralized trust anchor, ensuring the integrity of aggregated data and the overall state of the network [5].

The study, by providing a layered approach, directly tackles the MANET challenge of energy efficiency, which is a significant gain over ad-hoc networks. The zone-based structure is a practical solution for managing network topology and reducing broadcast storms inherent in ad-hoc networks. However, the system is heavily dependent on the availability and trustworthiness of the selected Cluster Heads, which could become localized points of failure.

The paper uses an Improved Elephant Herd Optimization algorithm for routing. This highlights the value of using metaheuristic optimization, which can find a route that is balanced across multiple factors—not just distance. The proposed idea can be expanded in future from Local Trust Diaries into a more independent trust model inspired by this.

3.1.6 VANET Secure Reputation Evaluation & Management Model Based on Double Layer Blockchain (DBREMM)

DBREMM is an advanced reputation system for VANETs that uses a double-layer blockchain. It measures trust using two parts, Direct Trust and Indirect Trust. Direct Trust comes from a vehicle's own sensor data and is calculated using Bayesian inference. Indirect Trust is based on opinions from nearby devices, which are given different weights depending on their past reputation. Both types of trust are combined to form a final score, which is then saved on the blockchain [6].

This hybrid trust model gives strong protection against both single malicious vehicles and group of attackers working together. It combines direct and indirect trust to make accurate decisions about vehicle behavior. However, it needs constant data from many fast-moving vehicles. Collecting, weighting and merging these trust scores in real time is complex and slow. This delay can be dangerous because V2V safety systems need to react instantly. Any latency in trust updates could lead to wrong decisions or late warnings during driving.

DBREMM trust model is a more complex version of our Bayesian Trust Diaries. Both use Bayesian logic to measure trust. However, our model is simpler and faster. We use an easy a/B update method that requires very little computation. The trust score we generate is used throughout our system from creation to route selection. This makes it an important and efficient part of our overall protocol design.

3.1.7 Blockchain-Enhanced Secure Routing Protocols for Vehicular Ad Hoc Networks

The paper integrates blockchain with standard DSR protocol. The blockchain stores the record of node interactions, which is used to compute trust metric for each vehicle. This trust metric is helpful in algorithm's path selection logic. The source to destination packets are encrypted using Public Key Infrastructure i.e. asymmetric cryptography [7].

One of the strengths of this research lies in the use of blockchain to create an accumulating trust value. The neighboring nodes contribute in the verification during packet forwarding without relying solely on final destination to report a problem. However, the study's limitation include also in its monitoring process. A group of malicious nodes can team up to provide false reports for one another in the network. The study also did not address mobility conditions where networks are changing rapidly.

This research is directly relevant to the proposed work, where both aims to provide security. While the study focused primarily on how a route is find based on trust values, the proposed work extends the work by providing optimized path.

3.1.8 Security in MANETs

The paper proposed solution named ”MPR Blockchain.” This approach utilize the Multi-Point Relay (MPR) selection mechanism from the OLSR protocol. Instead of requiring every node in the network to manage a computationally intensive blockchain, only the elected MPR nodes are tasked with maintaining and sharing their own local blockchains. These ”MPR Blockchains” are used to record transactions and calculate reliability estimates for nodes within their neighborhood. The decentralized zones of trust are managed by these key relay nodes [8].

The paper proposes a resource-conscious design; by limiting the blockchain overhead to a subset of nodes (the MPRs). The rest of nodes interact less intensively. Confidentiality, integrity, availability are emphasized. However, the paper is purely a proposal and, as stated by the authors, lacks any implementation or simulation.

This paper uses the ”MPR Blockchain” concept, which is analogous to a zone-based or clustered approach. This concept is similiar to idea of your multi-layer architecture, as proposed in the idea that most of the nodes interact less while the ledger is managed somewhere else. The paper also suggests that segmenting the network distributes the security in ad-hoc networks, which is adopted in the proposed idea.

3.1.9 Clustering strategy (ICRA)

In 2023, Jingjing Guo et al. proposed several innovative methods aimed at improving routing protocols in UAV ad hoc networks. The key methods they discussed is Clustering module. This module is responsible for organizing the UAV nodes into clusters. Each node calculates its utility, which helps in determining its role within the cluster. They also devised a clustering strategy adjustment module. This module employs reinforcement learning to continuously adapt the clustering strategy based on the current network state. It learns the benefits of different strategies over time, allowing it to calculate the utility of nodes more accurately. Then comes the the routing phase where message forwarding takes place between different clusters [9].

The key features of the paper is that clustering method aims to reduce end-to-end delay and enhance the packet delivery rate, addressing common challenges in UAV networks

The paper and the proposed idea goals are similar but different in approach. The ICRA research paper addressing high mobility in V2V environment by adaptive clustering architecture. Whereas the proposed idea relies on PKI Blockchain and Bayesian Trust Diaries to address mobility as the idea is to adopt to unknown nodes.

3.1.10 Cryptography based Clustering

Roda Mohindra et al. proposed a Secure Cryptography Grouping Mechanism for MANET, including decryption, encryption, signature generation. Nodes are grouped into clusters (regions). Each cluster has a Region Head (RH) chosen based on energy, bandwidth, mobility, and lifetime. Member nodes register with RH. Packets are encrypted via ECC. Paths are discovered by AOMDV protocol [10].

The primary strength includes the paper's successful mitigation in common ad-hoc network attacks. However, The framework used ECC but lacked comparisons with other techniques, suggesting the need for recent comparisons to better establish the framework's efficiency.

The study used cryptography to achieve end-to-end encryption and integrity. The proposed study also achieves integrity by lightweight means. The proposed study used PKI for end-to-end delivery while maintaining integrity. The proposed study also extended this work by finding shortest delivery to destination.

3.1.11 Introduction to PBAG Protocol

The authors propose a blockchain-based authentication protocol (PBAG) that enhances the security of vehicle authentication in IoV environments. This protocol addresses the problems of existing schemes that require direct intervention with the blockchain for authentication, which was increasing latency [11].

A key feature of the PBAG protocol is the use of a public global commitment generated by the Root Authority. This global commitment is based on all valid certificates issued to authorized nodes, allowing for efficient verification without the need to ask the blockchain directly. Moreoever, PBAG protocol ensures strong privacy features, including unlinkability. It enables nodes to authenticate themselves anonymously using evaluation proofs, while also providing a mechanism for traceability in case of disputes. However, the study does not consider power consumption on ad-hoc network devices and did not optimize the methologies.

The fact that PBAG addresses the most frequent and time-sensitive task: vehicle authentication by RA. The proposed study also uses concept similiar to RA i.e PKI Blockchain. Which not only stores secure certificates but also helps in timely delivery of node authentication. In fact, PBAG approach of providing this fast service of authentication is inspired in the proposed idea so that a specialized layer operates specially for this purpose.

3.1.12 Cluster Based MANET Security

Richa Agrawal et al. proposed an N-th Degree Truncated Polynomial Ring (NTRU) based key agreement scheme to determine pairwise and session keys for fully distributed MANET environments. The paper first organizes the network into a clustered topology. Within this structure, it implements the (NTRU) public key cryptosystem, a lattice-based cryptography method known for its resistance to quantum attacks. The framework uses NTRU for key agreement to establish secure communication between pairs of nodes and to generate group session keys. It employs a sophisticated secret sharing scheme based on Shamir's scheme and NTRU to distribute the secret keys of the vital clusterhead nodes among the participants [12].

The important strength is the paper's forward-thinking approach to security by using NTRU, which is faster than traditional cryptosystems like ECC and also secure against quantum threats. Structuring the network into clusters is also an effective way to manage a large network. However, a potential weakness is the architectural complexity in large networks. If a sufficient number of nodes within a cluster are subjected to common ad-hoc attacks, they could potentially reconstruct the clusterhead's key, disturbing the whole network.

The paper's strategy of moving away from a flat network topology to a more hierarchical one to improve manageability, is relevant with the proposed multi-layer blockchain concept. Instead of one blockchain handling every task (identity, routing, trust scores, etc.), the proposed idea divides the task. The Routing Ledger is a specialized authority for path data. Just as the clusterhead unburdens regular nodes, the PKI layer unburdens the Routing Ledger from the backend computations of certificate or verification.

3.1.13 Graph Neural Lasso for Dynamic Network Regression

The GNL model is designed to predict how nodes in a changing network behave over time. It learns both the connections between nodes and their time based patterns together. A special unit called the GDU handles time information, while an attention layer identifies which nodes influence each other. It was tested on traffic datasets and showed better prediction results than older models [13].

The strength of GNL is its ability to learn changing relationships between nodes over time. It captures both time and structure information in a smart way. The sparsity feature also removes weak links, making the model lightweight. However, it needs high computing power, as it keeps updating both the node data and graph structure. The learned connections are also hard to explain, which reduces clarity.

The study GNL is relevant to the proposed system as both also works in dynamic and decentralized network. In our secure design, nodes may or may not move fast and make temporary links. The idea of learning changing relationships can help predict trust and route stability between nodes.

3.1.14 A Novel Trust Mechanism for Addressing Black hole Attacks

The trust mechanism by the authors is designed to prevent blackhole attacks in MANETs. The concept used is direct and indrect trust. Direct trust is the node's own experience with its neighbors while the indirect trust is based on recommendations obtained from other nodes. The system combines both to calculate a node's trust score. They demonstrated an improved packet delivery ration and less data loss compared to traditional trust mechanisms [14].

The main strength of this mechanism is its use of indirect trust to identify hidden or smart attackers. It works even when direct observation is not enough. It also improves network performance by finding and avoiding malicious nodes quickly. However, it has some weaknesses. Sharing recommendations increases communication overhead and may cause delay. It also depends on honest nodes to give correct reports. If some nodes give false trust values, accuracy may drop.

The study is useful for our FYP because we also aim to handle blackhole attacks in a decentralized network. The idea of combining direct and indirect trust can help improve our Bayesian trust diary model. In our system, this concept can be integrated through trust weighted updates or signed feedback stored in blockchain logs. While this paper uses cluster based indirect trust, our model works fully peer-to-peer.

3.1.15 A regression-based technique for link failure time prediction in MANET

The regression based technique is used to predict when a link in a MANET will break. It uses a least squares polynomial regression model to estimate link failure time. The method studies how signal strength changes as nodes move apart. It creates a polynomial curve to guess when the signal will become too weak for communication. The results show that it gives more accurate predictions of link failure time [15].

The main strength of this method is its simplicity and low computation cost. It does not need any special hardware or large infrastructure. It helps predict link breaks before they happen. However, it assumes that signal strength always reflects link stability, which is not true in all cases. Interference, noise and obstacles can affect the signal. Polynomial regression can also overfit when the data is noisy or limited.

This study is relevant to our FYP because our system also needs to handle link changes in a secure way. A similar regression model can help predict unstable connections between nodes. In our system, this can work together with Bayesian trust diaries and blockchain logs. When the regression predicts a weak link, the system can lower its trust score or change the route early. This helps make routing more stable and reliable. The method is lightweight, which fits the limited resources of nodes in our design.

3.2 Literature Review Summary Table

Following table 3.1 provides findings of past articles related to VANET protocols and security.

Table 3.1: Literature Review Summary Table

Author	Method	Results	Limitations
Majumder et al. [1]	DSR with blockchain, homomorphic encryption, and reputation system.	Delay reduced by 18.5%, 89.6% detection; CPU +30%.	High cost from homomorphic encryption; poor scalability.
Kudva et al. [2]	Two-level trust via AODV; RSUs as validators.	92.4% PDR, 95% detection, 21% less overhead.	Strict trust threshold; low anonymity.
Swamy et al. [3]	Blockchain + deep learning + BHO optimization.	97.4% accuracy, 96.1% PDR, 42 ms latency.	High computational complexity.
Morin et al. [4]	Lightweight LED cipher, pseudo-identities, group key mgmt.	1.7 ms enc/dec, 40% lower energy vs AES-128.	Centralized group manager (single failure point).
Sugumaran et al. [5]	Block-DSD using clustering, ANFIS, ECC, IEHO.	+27% lifetime, 95.6% PDR, -30% energy.	Relies on cluster heads; localized failures.
Hou et al. [6]	DBREMM: double-layer blockchain for trust.	96.3% accuracy, 94.7% detection.	Latency in real-time trust aggregation.
Hiba et al. [7]	Blockchain + DSR + PKI encryption.	93.8% PDR, 31 ms latency.	Vulnerable to collusion attacks.
Mouchfiq et al. [8]	MPR nodes maintain local blockchains.	60–70% reduced ledger load (conceptual).	No simulation; theoretical only.
Guo et al. [9]	RL-based ICRA clustering for UAVs.	94.8% PDR, +22% stability.	RL overhead not ideal for small devices.
Mohindra et al. [10]	ECC encryption + AOMDV multipath routing.	91% PDR, 86% detection.	No comparison with newer crypto schemes.
Feng et al. [11]	PBAG: blockchain auth via global commitment.	14.3 ms delay, 99.2% success.	Ignores device power consumption.
Agrawal et al. [12]	NTRU + Shamir scheme for cluster security.	35% faster key exchange; quantum-safe.	Complex architecture for large networks.
Liu et al. [13]	GNL: temporal graph learning via Lasso.	97.1% accuracy, 21% faster runtime.	High computational cost; unsuitable for real-time.
Ghaleb et al. [14]	Direct + indirect trust for blackhole attack detection.	92.5% detection, 93.7% PDR.	High overhead in recommendation sharing.
Patel et al. [15]	Polynomial regression for link failure prediction.	95.2% accuracy, -17% packet loss.	Signal-based prediction unreliable in interference.

3.3 Related Work

Industry applications used blockchain to secure the communication. Many of them applied security on vehicular communication and information sharing systems. The applications reveal the practicality of applying blockchain and distributed ledger technologies to enhance the trust, authentication, and security of data in ad hoc networks (PANETs).

3.3.1 Filament (IoT + Ad-Hoc Mesh + Blockchain)

Filament builds mesh-network devices called “Taps” for industrial IoT. These devices use long-range radios and cryptographic chips to set up temporary mesh networks without needing a central system. They also use blockchain technologies such as Bitcoin and Ethereum to create decentralized trust, keep data secure, and support smart contracts. This matters for your PANETs context because the nodes talk to each other directly and use blockchain to handle trust and data checks[16].

3.3.2 Lightweight Scalable Blockchain for IoT

LSB is a research architecture designed for IoT devices that have limited resources. It uses a tiered or clustered blockchain structure, so only some devices handle full blockchain validation. High-resource nodes serve as cluster heads. The consensus and trust mechanisms are set up to lower cost, latency, and overhead. This method is similar to your idea of combining blockchain, trust, and routing in networks with limited resources[17].

3.3.3 Blockchain-Based Lightweight Trust Management in MANETs

There is a published system that uses blockchain to create distributed trust among routing nodes in a MANET. It works with the OLSR routing protocol and uses a lightweight consensus method to keep validation costs low. This approach is closely related to your work because it also uses blockchain-based trust management in a fully ad hoc, multi-hop network[18].

3.3.4 Briar(Secure Mesh Messaging)

Briar is an open-source app that allows peer-to-peer messaging using Bluetooth, Wi-Fi, or removable storage. It is designed for strong and decentralized communication without central servers. It also offers end-to-end encryption for added security. While it doesn’t use blockchain, it serves as a solid example of secure and fully decentralized communication in an ad hoc network[19].

3.4 Conclusion

The relevant studies worked on applying different technologies and implement security in ad-hoc network. Different technologies like cryptographic methods, intelligent clustering and use of blockchain has been explored. Many of them achieved malicious node detection accuracy by 90%. However, a study of literature also revealed gaps. A major challenge is the performance issue associated with integrating blockchain e.g high computational overhead and latency, which make it impractical for real time vehicular environment, as seen in ABCD framework [5]. Also, relying on specific nodes for validation centralized points of failure, a significant risk in a decentralized network [2]. Our proposed protocol addresses the scalability issue by replacing the single, slow ledger with a specialized multi-layer blockchain architecture. Moreover, our system's Bayesian Trust Diary for reputation management ensures that there is no single point of failure. By opting for PKI system over heavy encryptions, our protocol is explicitly designed for the resource-constrained nature of nodes.

Chapter 4 Software Requirement Specifications

This chapter details the software requirements for the project "Secure Data Exchange System for Ad-hoc Networks Using Blockchain-Enhanced DSR" The document defines functional and non-functional requirements, key features, quality standards, assumptions, and risk analysis, all aligned with the goal of securing DSR routing against blackhole, wormhole, and Sybil attacks.

4.1 List of Features

The system has several security features focused on safety:

- End-to-end PKI-based encryption keeps data secure during transmission.
- Each node creates key pairs and gets digital certificates through blockchain verification.
- A multilayer blockchain consists of three layers: the PKI layer, the routing ledger, and a meta-layer dispatcher.
- It can detect and reduce attacks like blackhole, wormhole, and Sybil attacks using blockchain validation and trust scores.
- A Bayesian trust diary assesses the trustworthiness of nodes based on their past behavior.

4.2 Functional Requirements

This section outlines the functional requirements across core functionalities, including path management, the integration of blockchain, and security measures.

4.2.1 Security and Encryption

- The system shall implement encryption for data packets.
- The system shall encrypt data packets before blockchain storage.
- The system shall decrypt data packets only at the final destination node.
- The system shall use Signature Algorithm for unique certificate generation.
- The system shall verify digital signatures for neighbor list publications.
- The system shall implement cryptographic signatures for all route messages.

4.2.2 Attack Mitigation

- The system shall detect and mitigate blackhole attacks.
- The system shall detect and mitigate wormhole attacks (open folded, half open, and closed types).
- The system shall detect and mitigate Sybil attacks.
- The system shall detect and mitigate Replay attacks
- The system shall calculate trust scores for each node and should be stored in routing information.

4.2.3 Node Registration and Authentication

- The system will create key pairs for each node.
- The system will register nodes by sending out their ID and public key.
- Certificates will be given out after lightweight Proof of Work and Quorum vouches.
- The system will provide certificates to verified nodes.
- The system will check node certificates before allowing them to participate in routing.

4.2.4 Receipt Generation and Management

- The system shall generate receipts for all route discovery participants.
- The system shall store received RREQ messages and signatures as receipts.
- The system shall include secure data (SD) in receipt messages.
- The system shall timestamp all receipt transactions.

4.3 Quality Attributes

The algorithm aims to incorporate the following fundamental quality attributes in order to ensure the prevention of defects and adopting to large-scale.

4.3.1 Security

Blockchain is used to securely store node identities, route records, and trust scores in a way that cannot be tampered with. Encryption is applied to make sure that all data sent between nodes stays private and unchanged. The system is also designed to protect the network from common attacks like blackhole, wormhole, Replay and Sybil attacks.

4.3.2 Reliability

The distributed blockchain design ensures that there is no single point of failure when authenticating nodes or evaluating trust in the network. Even if some nodes behave maliciously, the system uses trust diaries to maintain a consistent and reliable reputation score for each node.

4.3.3 Performance and Efficiency

The system's performance is mainly focused on security-related tasks such as verifying trust scores, issuing certificates, and encryption or decrypting data packets. It does not make any claims about improving routing efficiency, network speed, or support for mobile nodes.

4.3.4 Scalability

The multi-layer blockchain design allows new nodes to join the network without needing any central authority. Each node is able to evaluate trust and verify certificates on its own, making the system more flexible and independent.

4.3.5 Non-Functional Requirements

This section mentions several non-functional requirements to ensure performance and efficiency of the system.

4.3.5.1 Performance

- The system shall establish a secure route from source to destination node in under 500 milliseconds such that blockchain, encryption and validation does not introduce excessive delay for DSR.
- The system shall complete data packet encryption in under 12 milliseconds per packet on average to maintain data confidentiality without introducing unnecessary delays.
- The system shall complete blockchain block creation in under 8 milliseconds on average.
- Trust evaluation and detection of blackhole, wormhole, or Sybil attacks shall be completed promptly to prevent security breaches.
- Route receipts and blockchain updates shall occur without compromising the integrity of the security layer.

4.3.5.2 Usability

- The system shall provide security-focused visualization, such as color-coded indicators for detected malicious nodes and validated route receipts.
- Trust scores for all nodes shall be displayed in a secure interface, allowing network administrators or users to monitor node reliability.

4.3.5.3 Scalability

- The security framework shall support up to 50 nodes per network without central authority.
- Each node shall maintain security-related neighbor lists for 5–20 neighbors with configurable limits.
- The system shall support encrypted packet sizes of up to 400 bits while maintaining secure transmission integrity.

4.3.5.4 Security

- Blockchain shall maintain 100% tamper detection for all stored transactions and certificates
- All nodes shall be authenticated via PKI blockchain before participating in the network.
- Unique cryptographic key pairs shall be generated using secure random number generation.
- Trust scores shall be calculated continuously, and nodes with scores below 0.4 shall be flagged as malicious.

4.4 Assumptions

The system assumes a network of up to 50 node so that the security processes remain manageable. Each node is considered to have limited processing power but still capable of performing encryption, digital signatures, trust score calculations, and lightweight blockchain tasks. The security model also assumes that PANETs face attacks similar to MANETs, such as blackhole, wormhole, and Sybil attacks. All nodes communicate directly in a peer-to-peer manner without any central server or authority. It is also assumed that nodes will participate honestly in blockchain consensus and trust reporting, although some may still try to behave maliciously. The model focuses only on securing data exchange and does not aim to improve routing speed, network efficiency, or mobility handling. Finally, all cryptographic and trust-related operations are expected to run smoothly on standard Linux PCs and using the data exchange application.

4.5 Use Cases

Following are the use cases identified in proposed system.

Table 4.1: Use case for secure route discovery using Dijkstra's algorithm and blockchain for trusted path verification.

Name	Route Discovery		
Actors	User		
Summary	The source node initiates a secure route discovery process to find the shortest path to the destination node using Dijkstra's algorithm and blockchain validation.		
Pre-Conditions	Source node is registered in the network with a valid key pair.		
Post-Conditions	All intermediate nodes have verified the route, and route information is stored in the blockchain.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The Source Node initiates a route discovery request to send data to a specific destination node.	2	The system triggers Dijkstra's algorithm to compute the shortest and most efficient route across available nodes.
3	The source node digitally signs the Route Request (RREQ) and broadcasts it to neighboring nodes.	4	The system attaches the route and security data to the blockchain ledger.
5	Source node broadcasts RREQ message.	6	System broadcasts RREQ to all neighbor nodes in the shortest path.
		7	System checks the certificate and signature at each hop.
9	The source node receives the RREP.	8	System confirms successful route discovery once the destination node gets the RREQ and an RREP is generated back along the path.
Alternative Flow			

	The source node signs the RREQ again and broadcasts it.	7-A	If blockchain verification fails, the route discovery process is aborted, and a new RREQ is initiated.

Table 4.2: Use case for secure data transmission using encryption and blockchain-based signature verification across each hop.

Name	Data Packet Transmission		
Actors	Source Node, Intermediate Nodes, Destination Node		
Summary	The source node sends an encrypted and signed data packet to the destination node along the pre-established and blockchain-verified route.		
Pre-Conditions	A secure and verified route between the source and destination nodes has been discovered and logged in the blockchain.		
Post-Conditions	The data packet is successfully received and decrypted by the destination node, with the transmission record updated in the blockchain.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The Source Node encrypts the data packet with the destination's public key and signs it with its private key.	2	The system forwards the packet to the first hop in the established route.
3	Each Intermediate Node receives the packet.	4	The system at each hop verifies the source's signature and the integrity of the route by checking the blockchain.
		5	If verification is successful, the node forwards the packet to the next hop.
6	The Destination Node receives the encrypted packet.	7	The system uses the destination's private key to decrypt the packet and verifies the source's signature.
No Alternative Flow			

Table 4.3: Use case for new node registration, ensuring only verified nodes join the network for secure communication and data transmission.

Name	Node Registration		
Actors	New Node, Existing Network Nodes		
Summary	A new node requests to join the network, generates a key pair, and has its identity certificate validated and recorded on the blockchain by existing trusted nodes.		
Pre-Conditions	The new node has the necessary software and is within communication range of the network. The network has an established blockchain.		
Post-Conditions	The new node's public key and certificate are stored on the blockchain, allowing it to participate in secure network activities.		
Special Requirements	Access to a trusted Certificate Authority (CA) or a decentralized trust mechanism.		
Basic Flow			
Actor Action		**System Response**	
1	A New Node generates a public-private key pair and a Certificate Signing Request (CSR).	2	The system broadcasts a "join request" to neighboring nodes, including its public key and certificate.
3	Existing nodes receive the join request.	4	The system validates the new node's certificate against the trusted CA. A consensus protocol is initiated among existing nodes to approve the new member.
5	The new node receives confirmation of successful registration.	6	Upon reaching consensus, the system adds the new node's public key, certificate, and initial trust score to a new block on the blockchain.
Alternative Flow			

	The new node is denied entry to the network.	4-A	If certificate validation fails or consensus is not reached, the system rejects the registration request and logs the failed attempt.

Table 4.4: Use case for route recalculation after a link failure, ensuring continuous and secure communication through an alternate path.

Name	Dynamic Route Recalculation		
Actors	Intermediate Node, Source Node		
Summary	When a link in an established route breaks, an intermediate node detects the failure, notifies the source, and a new secure route discovery is initiated to restore communication.		
Pre-Conditions	An active and blockchain-verified route exists, and data packets are being transmitted between a source and destination node.		
Post-Conditions	A new, secure route to the destination is established and recorded on the blockchain, or the destination is marked as unreachable if no new path can be found.		
Special Requirements	Nodes must be able to detect link failures (e.g., via missing acknowledgments).		
Basic Flow			
Actor Action		**System Response**	
1	An Intermediate Node fails to forward a packet to the next hop in the route (e.g., link break detected).	2	The system generates a Route Error (RERR) message and sends it back towards the source node along the path the packet came from.
3	The Source Node receives the RERR message.	4	The system invalidates the broken route in its records and initiates a new ”Secure Route Discovery” process to find an alternative path to the destination.

5	A new route is successfully discovered.	6	The system updates the blockchain with the new route information, and data transmission resumes.
Alternative Flow			
	The source node is notified that a new route cannot be found.	4-A	If the route discovery process fails to find any valid path to the destination, the system marks the destination as unreachable and informs the source node.

4.6 Hardware and Software Requirements

4.6.1 Hardware Requirements

Table 4.5: Hardware Requirements

Component	**Specification**	**Function**
Linux PC	intel core i5, 8GB RAM, 256GB SSD, Ubuntu 22.04 LTS	Each PC acts as a node in the proof-of-concept network. Handles packet encryption/decryption, blockchain operations, PKI management, trust diary updates, and attack simulation.
Number of PCs	3-5 nodes	Supports small-scale PANET simulation for security and trust validation experiments

4.6.2 Software Requirements

Table 4.6: Software Requirements

Software	Specification	Function
Ubuntu Linux	22.04 LTS	Operating system for running blockchian nodes, Python scripts and simulations.
Blockchain Framework	Pure C/C++, g++ and clang	Implements Multi-layer blockchain, certificate issuance, consensus mechanism, and logging of trust scores.
PKI Tools	OpenSSL or similar	Generates key pairs, signs certificates, and manages public-private key infrastructure.
Network Simulation Tools	Custom Minimal C++ Simulations	Simulates PANET environments for testing attack scenarios, route verification, and trust propagation.

4.7 Risk Analysis

In this part we highlight some of the risks that are associated with development, security, and performance.

4.7.1 Security Risks

The protocol aims to provide high security using blockchain and encryption, but certain risks remain. Future cryptographic weaknesses could emerge if current encryption or hashing algorithms become outdated. The blockchain layer could also be exposed to consensus-based threats, such as a 51% attack, if a majority of miner nodes become compromised. Moreover the network is vulnerable to Quantum, AI/ML attacks and Behavioral Assessment based attacks, thought these attacks are beyond of our scope

4.7.2 Operational Risks

Operational risks arise when moving to large simulations. Large topologies can lead to route instability, delayed synchronization, or overloaded blockchain state. Moreover, the project assumes low mobility, however it may not be stable for medium to high mobility scenarios.

4.7.3 Performance Risk

As the network grows, the system may suffer from latency, increased processing overhead, and larger packet sizes. If the blockchain grows rapidly, nodes having limited memory and computational power would struggle to maintain synchronization.

4.8 Conclusion

This chapter explains the main software requirements of the proposed system. It describes the system's features, performance goals, and working conditions. It also discusses both functional and non-functional requirements to ensure reliability and security. These details give a clear understanding of what the system does and prepare the base for its design and development in the next chapter.

Chapter 5 Proposed Approach and Methodology

This chapter explains the proposed approach and working methodology of the system. It describes the main components of the Data Exchange System and how they interact with each other. The chapter also presents the Blockchain System, Blockchain Structures, and security framework used for secure and efficient node to node communication. It further outlines how trust, authentication, and blockchain are used to ensure reliable data exchange and protection against common network attacks in a PANET environment.

5.1 PANET Security Framework

5.1.1 Environment Assumption

- Pure ad-hoc and no infrastructure
- Low mobility and resource constrained environment
- Requires fast, compact and secure communication methods

5.1.2 Security Threats in PANETs

Following are some security threats in PANETs:

- In a Sybil attack, an attacker makes several fake nodes.
- In a Blackhole attack, bad nodes take in packets but do not send them on.
- a Wormhole attack, attackers set up hidden shortcut paths that reroute traffic to bad nodes.
- Replay attacks happen when someone captures packets and sends them again to overwhelm or confuse the network.

Following are the important components of the Security Framework

5.1.3 Local Trust Diaries

- Stores `< ID, Acks, s, p>` per node:
  - `Acks, NAcks` : successful/ unsuccessful routing attempts.
  - `s` : Strength of previous trust
  - `p` : Previous trust probability
  - default p = 0.3, s = 1, Acks = NAcks = 0

- **Trust Update (Bayesian / Beta-Bernoulli):**

$$Alpha = p * s \tag{5.1}$$

$$Beta = (1-p) * s \tag{5.2}$$

$$p_{new} = \frac{Alpha + Acks}{s+n} \; where \; n = Acks + NAcks \tag{5.3}$$

$$s_{new} = s + n; \; Acks = Nacks = 0 \tag{5.4}$$

This proposed trust update mechanism is based on Bayesian Probability and Bernoulli distribution. Bayesian inferences allow trust to represent as a probability distribution on the basis of prior belief and new evidence. The Beta-Bernoulli equation supports incremental learning without requiring historical data.

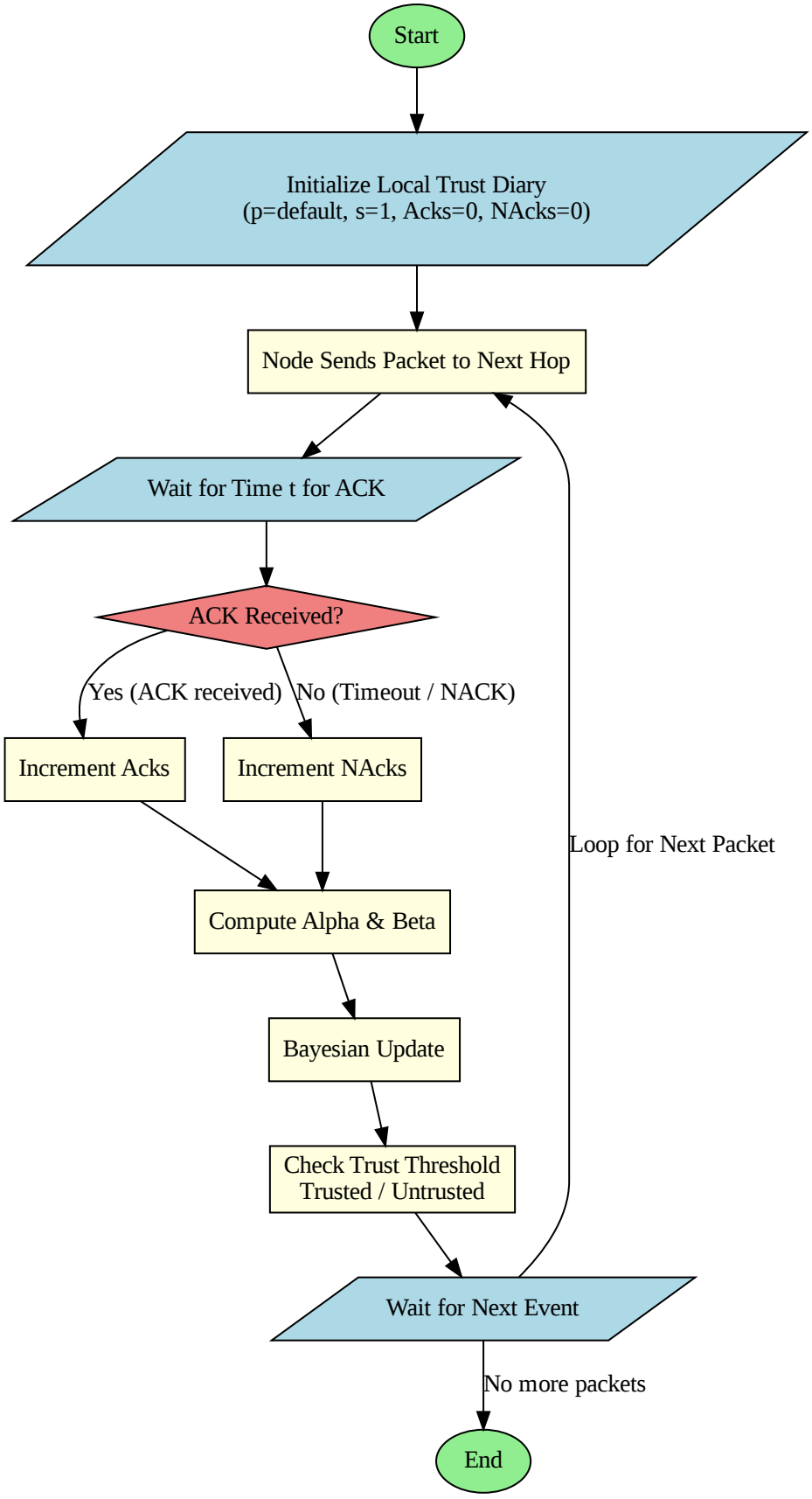

Figure 5.1: Local Trust Diary Flowchart. Trust is represented as a probability distribution and computed on each node.

5.1.4 PKCertChain

This blockchain is responsible for node certification, registration and certificate renewal

5.1.4.1 Node Registration Phase

- Broadcast `<ID, PK, Signature>` using ECC+ECDSA and will keep its own `SK` to itself
- **Consensus Mechanism:**

  Consensus Mechanism consist of 2 steps:

  1. Number Hash Inversion Challenge (Proof of Work): It will simply provide a random number hash in a positive number range and the new node will try to find that number
  2. Quorum-Based-Stratgy: Initially `MinThreshold = 0` ; if the number of nodes vouch for the new node and equal to MinThreshold then generate a certificate.
  3. MinThreshold Update Policy:

     Threshold Update Equation:

     $$\text{MinThresh}_i = \text{PrevMinThresh} + k_1 * \text{Density Change} + k_2 * \text{Network Reception Change} \tag{5.5}$$

     Intermediate Variables:

     $$N_t = TotalCertificatesIssued(t) - TotalCertificatesExpired(t) \tag{5.6}$$

     $$P_t = \text{Number of Neighbors received the Hello Packet}$$

  4. Derived Quantities :

     $$DensityChange = \frac{N_t - N_{t-1}}{N_{t-1} + 1} \tag{5.7}$$

     $$NetworkReceptionChange = \frac{P_t}{N_t} - PrevMinThreshold \tag{5.8}$$

  5. Constant Quantities:

     $$k_1 = 0.2 \; ; \; k_2 = 0.6$$

6. Clamping:

$$\text{MinThresh}_t \in [0.2, 0.9]$$

7. It is based upon Adaptive Feedback Principle, Density change - Environmental Feedback , Performance Feedback and Bayesian adaptation. After this digital certificate will be generated and a block will be broadcast to update the blockchain globally. Certificates will be issued temporarily.

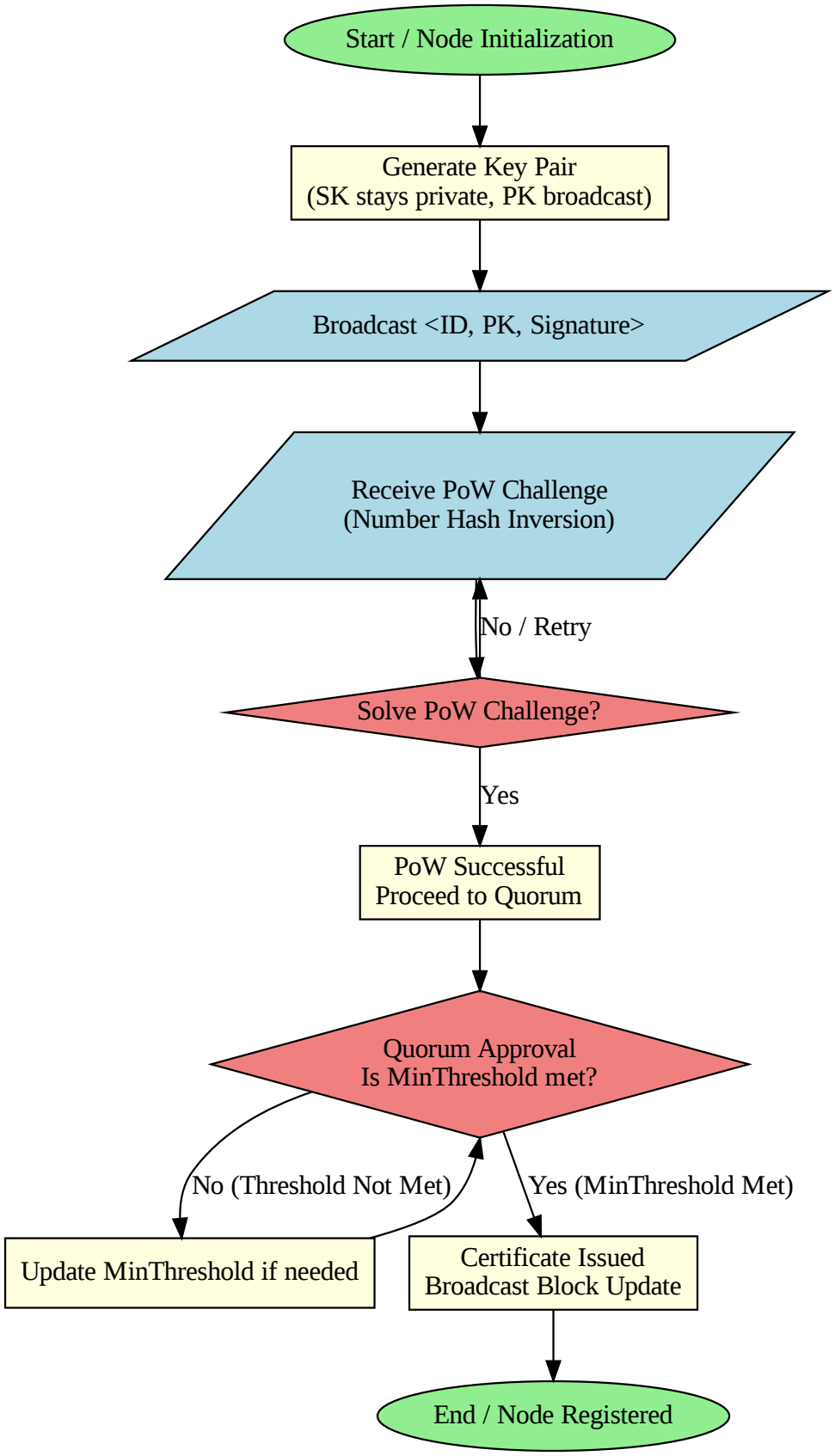

Figure 5.2: PKCertChain

5.1.4.2 Certificate Renewal

If a node's certification has expired then it will be sent to blockchain (like to another node). It will send `<ID, PK, Signature, PrevCert>` if PrevCert is validated then it will be automatically assign another certification by preserving the certificate link.

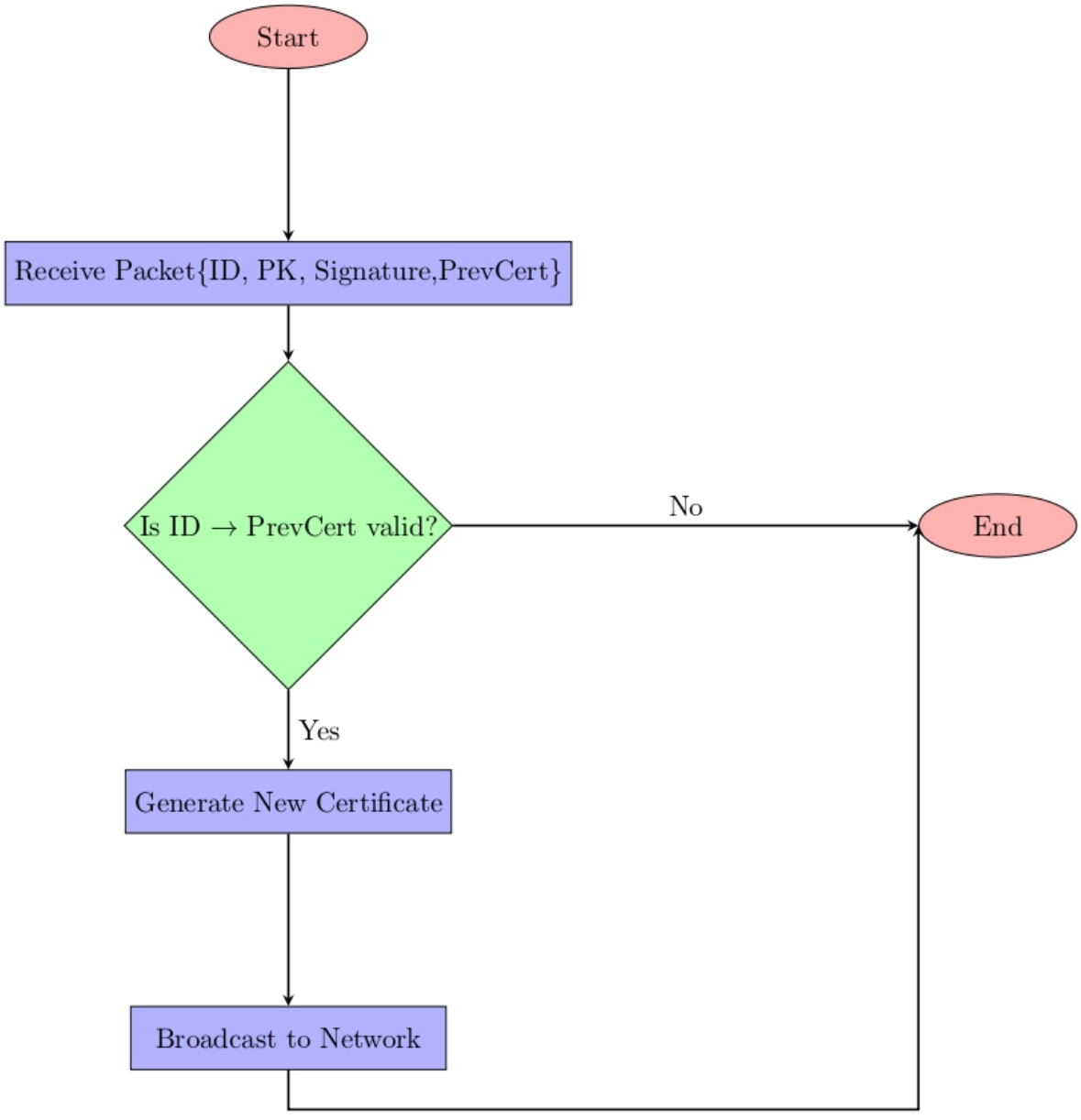

Figure 5.3: Certification Renewal takes place when it has expired. The request will be sent to blockchain.

5.1.5 RouteLogChain (Packet Routing and Path Trust)

1. Nodes must be registered to the PKCertChain
2. Sender will generate RREQ and broadcast packet using DSR algorithm, to the Destination Node
3. Destination Node will receive all the candidate paths and each RREQ will contain all the nodes in the path. Now, Destination Node will find the optimal path using multimetric Dijkstra's Algorithm
4. **Consensus Mechanism:**

$$Score = \alpha \cdot \text{TotalTrust} + \beta \cdot \text{Latency} + \gamma \cdot \text{Hops} + \varepsilon \cdot \text{Bandwidth}$$

   Find the most optimal path and then send RREP packet for the selection of the path.

5. **Packet Transmission:**
   Now sender will send the packet through that path selected by the destination node.

   Sender Packet: Sender will encrypt the Message with the Destination's Public Key and append its Signature.

$$Packet_{sender} = Enc_{PK_{DESTINATION}} || Sign_{Sender}(Message)$$

   Intermediate Nodes Packet and Rolling Signatures: Now at îth node will receive packet:

$$Packet_{i-1} = Enc_{PK_{DESTINATION}} || Sign_{I-1}(Packet_{i-2})$$

   Each Node verification process: Now every node has to verify signature in the same way it moves to it because the previous whole packet is signed and replaced with older signature. In this way blackhole and wormhole attacks won't be happened. If packet not received then re-transmit packet. If packet is re-transmitted, signatures are not authentic or certificates are expired then increment NACK of respective node

   Destination Node: It will verify by using Rolling Signature method and then decrypt the message with its Secret Key

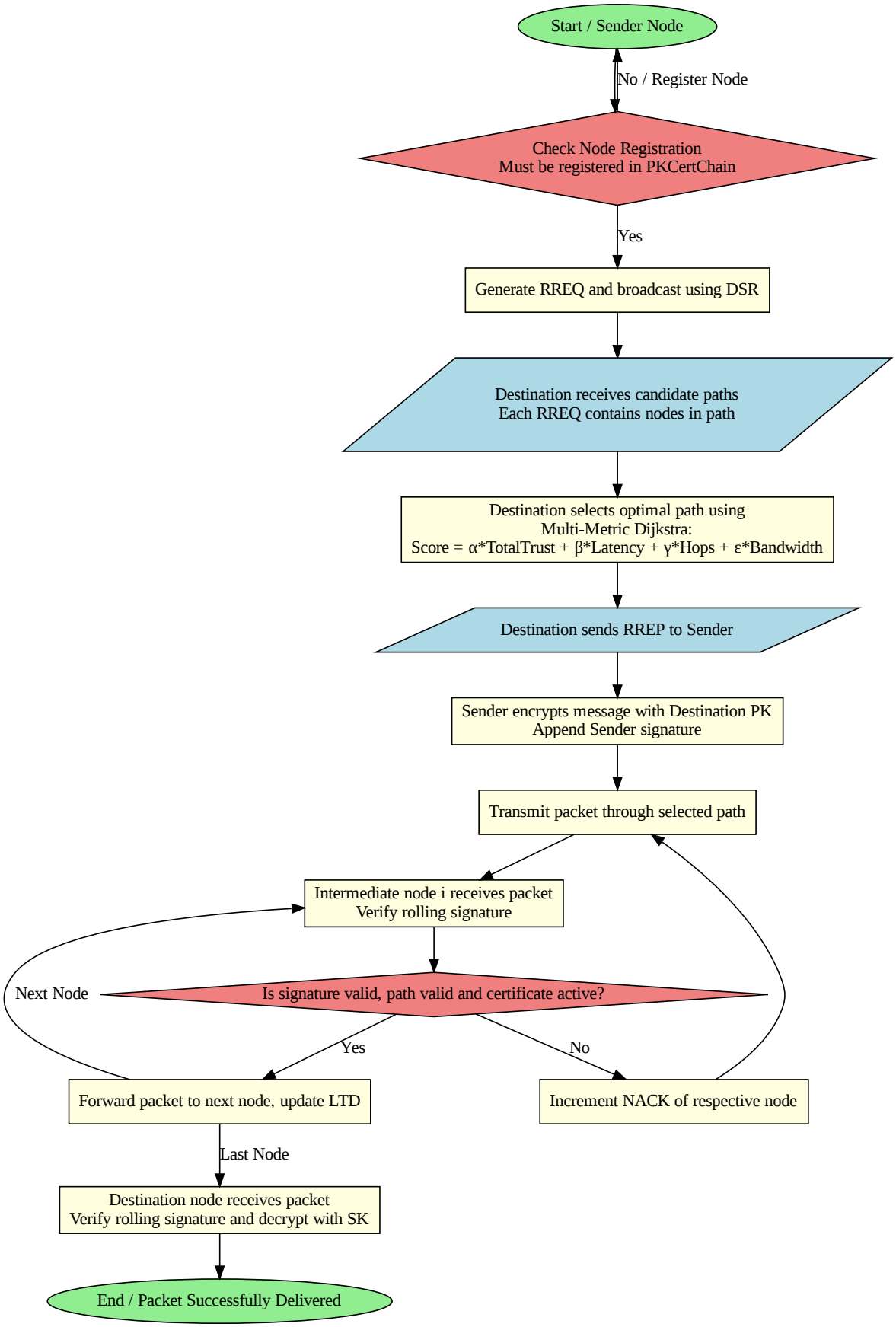

Figure 5.4: Every node has to verify signature. If packet not received then re-transmit packet. The Destination node will verify by using Rolling Signature.

5.2 Meta Blockchain

It is for auditing purpose and dispatches the block to respective blockchain.

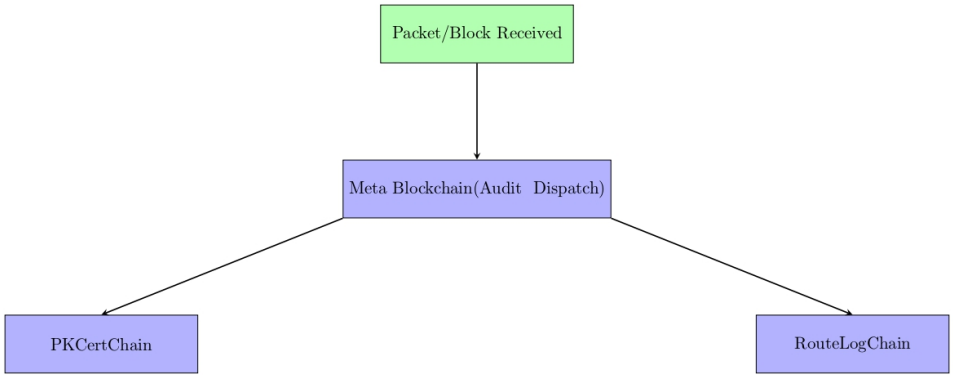

Figure 5.5: MetaBlockchain is responsible for routing either to PKI Blockchain or Route Log Blockchain

5.3 Route Cache

The DSR protocol itself manages the route cache based on history to reduce RREQ flooding. We will adopt this concept, but more secure, dynamic and adaptive way.

DSR protocol is a reactive protocol. If a route is not present it will broadcast RREQ packet, then store best path(s) in route cache and whenever there is a need of routing to Destination again then it will select path from cache and send DATA packet. If the path does not active then it will select other path or broadcast RREQ packet. In our security driven system, route cache will contain the knowledge of all nodes in topology thanks to PKCertChain Blockchain. Now the node will have the list of all nodes so, each node need to discover path. There are three scenarios for it:

- The reactive method is applied for undiscovered destination nodes' paths. Broadcast RREQ packet and then the destination node will select one path and then send it to the sender node. In this way, route cache will update the selected path (simple DSR way).
- It is possible that a node is inactive or there is a more trustworthy path after some time, we devised a reactive dynamic criteria. Thus it can be done in a parallel RREQ packet forwarding method by selecting top K paths and send RREQ based on these paths. Now, the destination node will decide to select any one of these paths. The selection criteria for top K paths is dynamic and adaptive. Consider T1 is the set of paths.
  - Initially T1 contains 100% number of paths (based on multi metric Dijkstra).
  - Sender will generate RREP for all paths in T1.

  - If the selected route is from T1 then there will be two scenarios:
    * If the selected path is from first half of T1 then shrink T1.
    * if selected path is from second half of T1 then expand T1

- Now the network may contain some undiscovered path and null paths for a destination node, so there is a need of a periodic proactive approach. Each node after t interval will broadcast RREQ for path discovery. In this way, it will calculate RREP and store in Route Cache. The time t will decrease if there is a change in network (new node registration, trust change, new path discovery) otherwise it will increase

In this way route cache will be dynamic and network-feedback dependent that adapt itself as per network conditions

5.4 Synchronization

The most important and expensive part of the methodology is the synchronization of the PKCertChain, RouteLogChain, and Local Trust Diaries. The overall security part of the synchronization is that each receiver will multiple same blocks and it should chose the majority same blocks and diaries.

PKCertChain Sync: Whenever a new node is registered it will broadcast to all nodes.

RouteLogChain Sync: There will be two time pointers t1 and t2 such that t1 is the time of previous blockchain sync while t2 is the new time for blockchain sync. Now after t2, all the delta blocks that means block generated during this interval will be broadcasted.

Local Trust Diaries: It is a two- way system such that each node will broadcast Local Trust Diaries and then update the overall trust for converge and consistency.

5.5 Conclusion

This chapter described the complete working method of the proposed system. The integration of blockchain and trust management strengthens the system's security, making it resistant to attacks and suitable for decentralized ad hoc networks. Moreover the system is security focused and optimized later approach.

Chapter 6 High-Level and Low-Level Design

This chapter explains how the proposed system is designed. It describes both the high-level structure and the low-level working of the secure routing protocol. The design shows how Dijkstra's algorithm, the DSR protocol, and blockchain technology work together to provide safe and efficient communication. It also explains the main modules, design considerations, and methods used for development.

6.1 System Overview

In this section we will explore the general description of protocol.

6.1.1 System Description

The system fundamentally solves the problem of establishing secure optimal paths between nodes in a decentralized setting. The protocol does so by integrating mathematical path optimization of Dijkstra algorithm and the DSR protocol of an ad-hoc network. Subsequently, storage of blockchain supports the system. The system decouples the intricacies of the physical network as well as the distributed ledger. Briefly, its essence is to facilitate real time route finding, dynamic adaptation to the low mobile topology of the network and also the maintenance of a trust ledger, which guarantees integrity and node authentication along with attack resiliency of data.

6.1.2 Functional Overview

The protocol performs several significant functionalities, which interact to create a secure environment of communication. The former is the Secure Route Discovery that is implemented prior to the traditional pathfinding. It extends the pathfinding according to the Dijkstra algorithm utilizing real-time trust rating according to the Bayesian Trust Diary of each node. In this, a reliable and shortest path is discovered. The system also does not respond to topological changes or the interference of malicious nodes whereby the system automatically responds to the loss of links or rather responds by adding negative variation to the trust score of a node. All this is handled by storage managing a Blockchain Integration, in which a multi-layered registry captures identities of the nodes, their actions, and modification of their trust in an irrevocable fashion. The second tier of Blockchain is the certificate management of PKI and is based on rolling digital signatures in order to provide the safety of every packet communication. These functions have the potential to be successfully combined to enable Attack Mitigation, where the protocol can be capable of actively detecting, isolating and preventing malicious nodes attempting to employ Sybil, wormholing or blackhole attacks.

6.2 Design Considerations

In this section, the key issues and considerations to be followed during the protocol development are discussed.

6.2.1 Assumptions and Dependencies

The design of the protocol is based on the following assumptions:

- The nodes are stationary or have low speeds (like that of pedestrian).
- The system is based on the secure registration, authentication and relies on Public Key Infrastructure (PKI) and revocation of node identities, which is operated by the PKCertChain Blockchain layer.
- It is assumed that every new node that joins the network has its initial neutral trust score. It dynamically in the long-term according to the behaviour of the node, a record of which is kept in the Bayesian Trust Diaries of the neighbours.

6.2.2 General Constraints

There are a number of real-world constraints that define the design of the protocol.

- The nodes are near to stationary such that it is not supposed to corrupt the trust diaries or interrupt the whole blockchain synchronization process.
- Characteristics of energy efficiency is also a consideration. The protocol is structured in a way to reduce cryptographic computations.
- The protocol should be tested in a simulation space concerning its performance and security.

6.2.3 Goals and Guidelines

The following design principles have guided the technical decisions.

- The goal is to create a system that is responsive to common ad-hoc attacks like Warmhole, Sybil and Blackhole attacks.
- The protocol is designed to optimize routing decisions with minimal computations. For this, the algorithm uses Dijkstra for path quality and DSR for speed.
- A core principle of the design is the complete avoidance of single points of failure. Trust management, identity verification, and routing are all distributed across the nodes via the blockchain.

6.2.4 Development Methods

The design and development of the protocol is a flexible and step-by-step way to build and improve. With incremental development and testing approach. Each core module (e.g., Routing Engine, Blockchain Module, Attack simulations) is developed in distinct cycles. The correctness of the design will be validated through simulation and performance analysis. This allows us to test the protocol under a wide range of network conditions and attack scenarios that would be impractical to replicate in the real world.

6.3 System Architecture

This section details the internal as well as external design of the proposed protocol.

6.3.1 Overview

This protocol aims at having a safe and effective routing system that combines the optimization principles of Dijkstra algorithm using Dynamic Source Routing (DSR) protocol. The architecture is meant to execute a number of key roles:

- Find the most trustworthy and efficient path between a source and a destination.
- Adapt quickly to frequent topological changes caused by node movement.
- Verify the identity of participating vehicles without a single central authority.
- Maintain an immutable, auditable record of node behaviour and trust scores.

6.3.2 System Decomposition

To achieve its objectives, the system is decomposed into a set of high-level modules.

- Routing Engine is the central intelligence of the protocol. It implements the hybrid routing logic by orchestrating its internal subsystems: the Trust Evaluator, the Dijkstra Processor, and the DSR Handler, as shown in 6.6
- Another important component is Mutli layer Blockchain which manages all node registration on the PKI Blockchain. Also trust metrics to the Routing Ledger are stored along with routes. 6.1 shows an high level overview how transactions are made in blockchain.
- Each node also has a local storage which tracks the node's neighbor table, monitors link stability, and manages the "Local Trust Diaries." It provides the Secure Routing Engine with the real-time topological and trust data needed to make informed routing decisions.

- A storage or mobile device is connected with each node which performs end-to-end encryption and decryption of messages. It also applies the rolling digital signatures on packets.

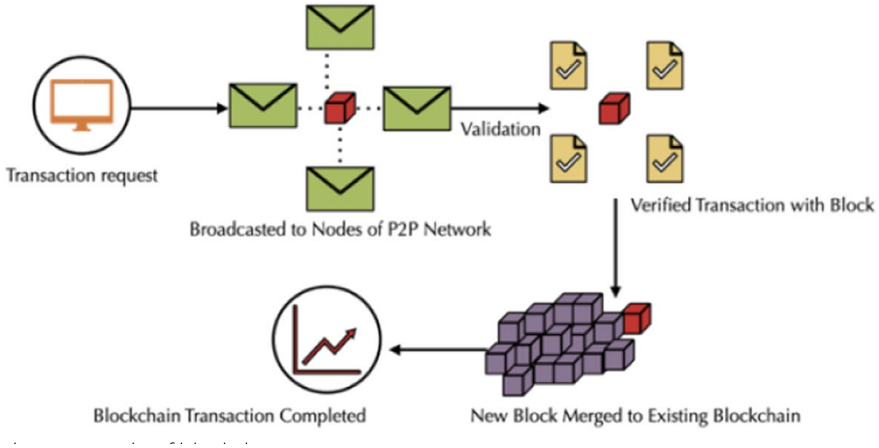

Figure 6.1: Transaction request is generated by Minor. Once broadcasted through tokens and verified by Intermediate nodes, it is then stored in the blockchain.

6.3.3 Sybil Attack Mitigation

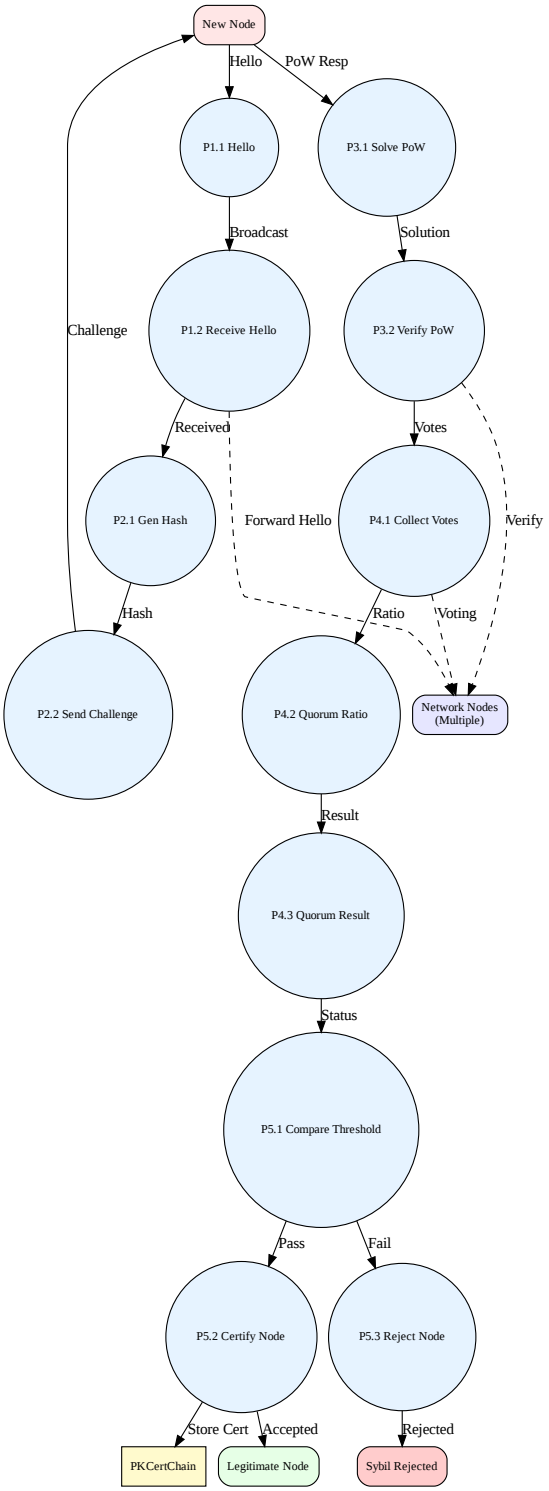

Figure 6.2: Level-2 DFD for Sybil Attack Mitigation

6.3.4 Blackhole Attack Mitigation

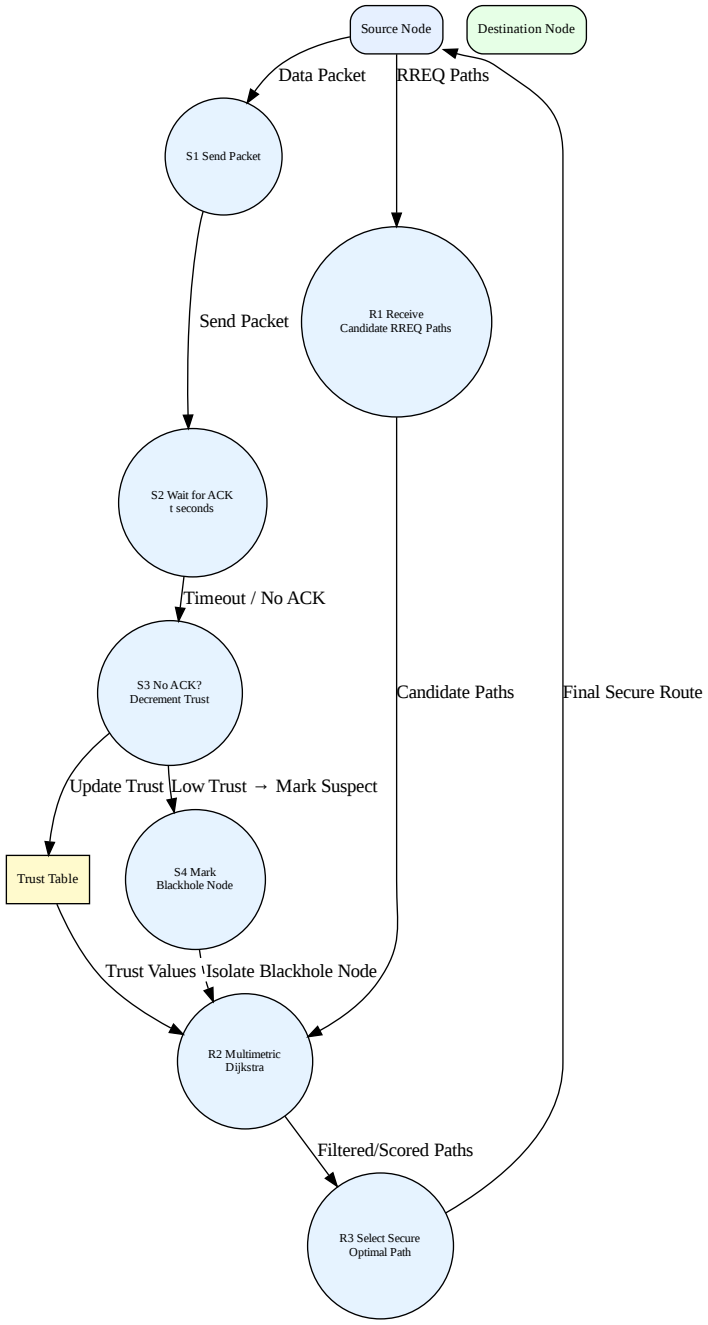

Figure 6.3: Level-2 DFD for Blackhole Attack Mitigation

6.3.5 Wormhole Attack Mitigation

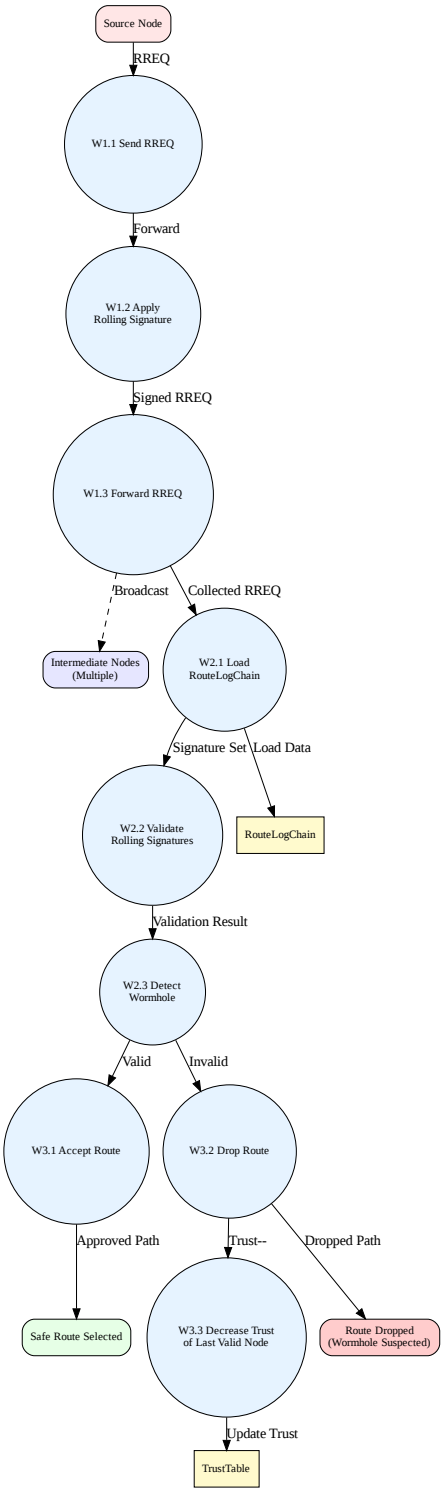

Figure 6.4: Level-2 DFD for Wormhole Attack Mitigation

6.3.6 Replay Attacks

It simply check the data packet duplications in the RouteLogChain and decrease trust.

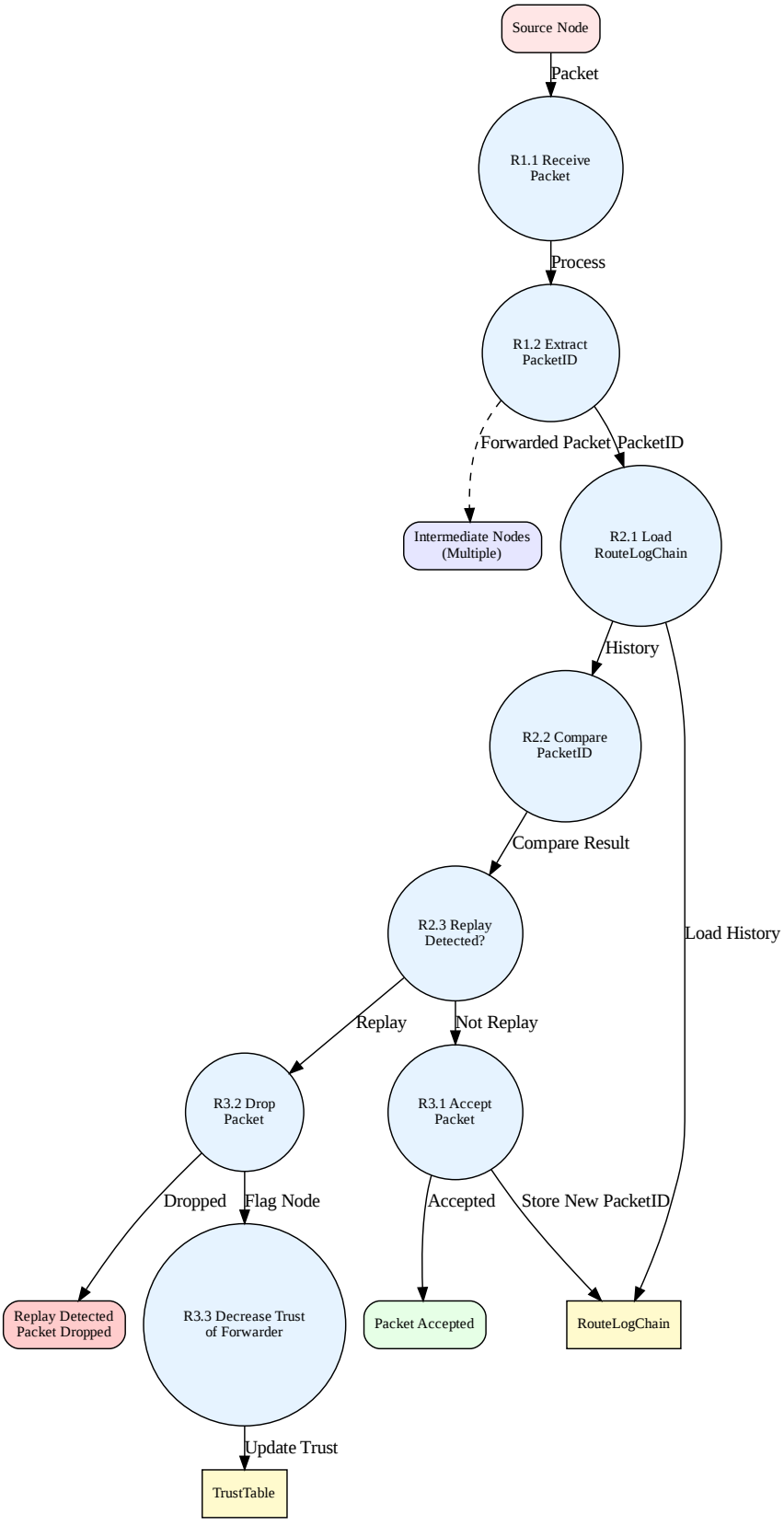

Figure 6.5: Level-2 DFD for Replay Attack Mitigation

6.4 Architectural Strategies

The architecture employs various strategies to shape its architecture and the decision making processes.

6.4.1 Blockchain as Data Storage

We use the distributed multi-layer for high-value, slow-changing data such as node identities (PKI certificates) and long-term reputation scores. This is ideal for ensuring the integrity and audit-ability of the system's trust. Conversely, highly dynamic, real-time data like neighbor tables, DSR route caches, and the raw data for the "Local Trust Diaries" are stored in memory connected with a node. This decision is critical for performance, as real-time routing cannot wait for blockchain consensus. The figure is shown by 6.1 illustrates how a new block is appended inside the blockchain.

Secondly, storing transient data locally is faster but less secure against local memory attacks. This trade-off is acceptable because the most critical data (identity) is stored to the blockchain, and any local data tampering would be quickly detected through mismatches in behavior.

6.4.2 Algorithmic Reuse and Scalability

We chose to adapt Dynamic Source Routing (DSR) and Dijkstra's algorithm because they are proven, well-understood, and provide a solid baseline for on-demand routing and path optimization. Reusing these components allows us to focus our contributions on the gap of integration of the trust mechanism and the blockchain layers, rather than on developing an entirely new routing scheme.
However, DSR can also be replaced with some other ad-hoc routing protocol. Future possibilities can be explored with AODV protocol. This is purely on experiment basis which works best according to situation.

6.4.3 Wireless Communication

Communication is entirely packet-based over a wireless medium. The protocol is designed for resilience in an unreliable network where links are vulnerable.
The DSR in the routing process is responsible for detecting link failures by generating a hello messages. This allows the source to immediately invalidate the broken route from its cache and initiate a new route discovery process.

6.5 Process diagram

The routing flow is shown by the figure 6.6, the routing logic is based on a combination of on-demand character of Dynamic Source Routing (DSR) protocol and trust-weighted Dijkstra algorithm.

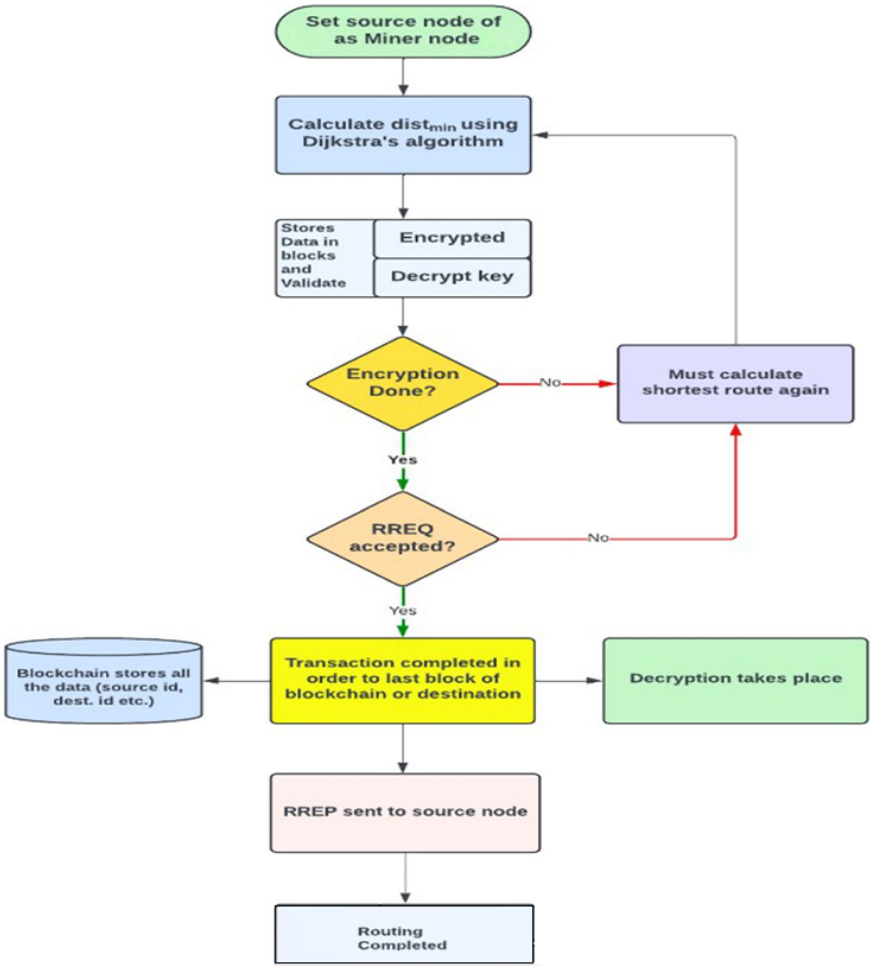

Figure 6.6: Flow process from a route request initiation to the final packet and compute a optimal path.

6.6 Policies and Tactics

This section outlines specific policies and tactics influencing interface and the implementation details of protocol.

6.6.1 Coding and Implementation Policy

- Core modules, including the routing discovery and decisions and the Multi-layer blockchain, will be first implemented in C++.
- A cryptographic library in a high-level language will be utilized for all security primitives, includ-

ing PKI, hashing, and the generation of certificates.

- The project will be managed using Git, with a structured and modular directory layout (/core, /blockchain, /network, /tests) to maintain a clear and organized codebase.

6.6.2 Performance Optimization Tactics

Efficiency and scalability are achieved in order to minimize latency and computational overhead without compromising the protocol's foundation and goal. The multi-layer architecture's primary advantage is realized here. When the routing engine needs to verify a certificate, it makes a non-blocking request. The Bayesian Trust Diaries does not broadcast an update to the Routing Ledger on the blockchain after every single source-to-destination communication. Instead they are being updated on when necessary.

6.6.3 Testing and Validation Policy

The correctness, performance, and security of the system are validated through a multi-stage testing policy.

- Each module is tested in isolation. For example, the Trust Diaries is unit-tested with predefined interaction scenarios to verify the correctness of the Bayesian update logic, and the Dijkstra Processor is tested with known network graphs to validate its path finding accuracy in C++ before going to actual simulations.
- The entire protocol will be then simulated in a network environment. Validation is based on measuring Packet Delivery Ratio End-to-End Delay, Routing Overhead and Trust Convergence Latency.
- The simulation environment will also be used to conduct security testing. We will simulate Sybil, wormhole, and blackhole attacks and test the protocol.

6.6.4 Maintenance and Extensibility Policy

The protocol is designed from the ground up to be maintainable over the long term. The modular architecture ensures that each primary component (Routing, Blockchain Interaction, Data Handling) can be upgraded independently. The cryptographic functions are abstracted. This allows the underlying PKI system to be upgraded from current standards in the future by developing a new cryptographic module. As far as ad-hoc networking is concerned, the initial implementation uses DSR, it can be adapted to replace with other protocols like AODV. Lastly, the current Bayesian Trust Diary can be replaced with AI-based trust models to detect complex attacks if appropriate hardware become available.

6.7 Conclusion

This chapter presented the overall design of the proposed secure approach for a routing protocol like DSR. The system combines Dijkstra's algorithm, the DSR protocol, and blockchain to ensure safe and efficient communication between vehicles. Its modular structure allows easy updates and strong protection against common network attacks. The design focuses on trust, reliability, and adaptability.

Chapter 7 Implementation and Test Cases

This chapter contains a detailed setup of the demonstration and implementation of the system to provide a detailed analysis of a security and performance on a single PC. The prototype emphasis on backend algorithms, blockchain based mechanisms, routing, and trust evaluation in a user space.

7.1 Implementation

7.1.1 Tools and Technologies

The implementation of the project requires a careful realistic simulation that demands a custom made simulation setup, the core algorithmic development management, and dependencies.

Table 7.1: Tools and Technologies

Category	Tools and Technology	Description and Purpose
Operating System	Ubuntu 22.04 LTS	OS for development, testing and evaluation
Programming Languages	C 23 and C++ 23	Language use for simulations, blockchain development, testing and routing
Compiler	g++ 11 / gcc 11 / clang 14	Modern and industry standard implementations
Concurrency	mutex, thread, atomic	Multi-threading, synchronization and lock free operations
Networking	POSIX sockets	Node to Node communication over a virtual network
Process Management	fork(), wait()	Each node is an individual and isolated process
Blockchain Storage	PKCertChain and RouteLogChain	Custom Linkedlist containing the hash pointers and timestamps
Cryptography	ECC + ECDSA	Encryption, Decrption, Singing and Certificates
Route Management	Adjacency List / Cache	Stores discover paths and make dynamic decisions
Trust Management	Local Trust Diaries	Manage Reputation of each node
Build System	CMAKE	Compile and Manage dependencies

7.1.2 Simulation Setup

The simulation contains tunable parameters that allow the user to analyze the protocol in various scenarios.

- Number of nodes
- Simulation Duration
- Each node should send a number of packets per second known as Send Rate
- Number of nodes that are simulating Sybil Attacks (Sybil Density)

- Fake ids density
- Number of nodes that are simulating Blackhole Attacks (Blackhole Density)
- Packet dropping density
- Number of nodes that are simulating Wormhole Attacks (Wormhole Density)
- Tunneling Density
- Number of nodes that are simulating Replay Attacks (Replay Density)
- Repetitive Packet Send density
- At a time how many nodes can be inactive is known as Mobility Density

Now each node will be an isolated process and each process will contain following threads:

- Communication Thread that is responsible for packet sending, receiving and maintaining route caches
- PKCertChain Thread that is responsible for new registration and certificates expiration
- RouteLogChain Thread that is responsible for validations, authentications, and routing data storage
- Local Trust Diary Thread, which is responsible for maintaining trust records
- Analytics Thread that will be evaluating all the metrics

The core simulator will be an event driven system, and each thread will be activated whenever an event is triggered, and otherwise it will be inactive.

7.1.3 Normal Nodes

These nodes will be honest and demonstrate the core methodology of the algorithm. These nodes will honestly evaluate trust metrics, mining, and route cache. The analyzer thread of these nodes will help us to examine the core methodology, security and performance under normal and attack scenarios.

7.1.4 Sybil Nodes

Sybil nodes will demonstrate fake identities generation and increase the PKCertChain size. The fake ids density will generate fake nodes per Sybil Node and demonstrate Sybil Attacks. The number of fake ids generation and the overall assessment will let us to evaluate the core algorithm under Sybil Attack scenarios.

7.1.5 Blackhole Nodes

These nodes will drop the packets during protocol and let the packet delivery ratio to be decreased. These nodes are malicious and degrade the network performances.

7.1.6 Wormhole Attacks

These nodes will demonstrate tunneling that will further mitigated by rolling signatures. Analyzer thread will help us to identify the performance overhead caused by detecting and mitigating the Blackhole attacks.

7.1.7 Replay Nodes

These nodes will flood the network by sending multiple same packets.

7.2 Test Metrics

The following metrics form the comprehensive evaluation framework for the proposed secure PANET system. They will be measured first in the single-PC virtual topology (FYP-I) and then validated in a real multi-PC environment (FYP-II). All metrics will also quantify the overhead introduced by the security mechanisms.

Table 7.2: Overall Test Campaign Plan

Metric	Purpose
Number of Test Cases Developed	Ensure complete functional and security coverage
Number of Test Cases Passed	Verify overall system correctness
Test Case Defect Density	Measure reliability of the implementation
Test Case Effectiveness	Assess ability to detect injected faults
Traceability Matrix	Guarantee full requirement-to-test mapping

7.2.1 Security Effectiveness Metrics

Table 7.3: Security Effectiveness Metrics

Metric	Purpose
Sybil Attack Detection & Mitigation Rate	Prevent creation of fake identities
Blackhole Attack Detection & Isolation Rate	Detect and isolate packet-dropping nodes
Wormhole Attack Detection & Mitigation Rate	Prevent out-of-band tunnelling attacks
Replay Attack Resistance	Ensure freshness of routing and data packets
False Positive Rate	Avoid penalising legitimate nodes
Average Attack Mitigation Time	Measure speed of response to malicious behaviour

7.2.2 Computational Overhead Metrics

Table 7.4: Computational Overhead Metrics

Metric	Purpose
Average PKCertChain Block Processing Time	Evaluate PoW, quorum voting and insertion cost
Average RouteLogChain Block Processing Time	Measure RouteLogChain overhead
Bayesian Trust Diary Update Time	Assess real-time trust computation cost
Rolling-Signature Verification Time (per hop)	Evaluate per-hop cryptographic verification cost
Full Packet Processing Time (encrypt+verify+resign)	Measure end-to-end per-hop latency overhead
Memory Overhead per Node	Quantify storage cost of blockchains and diaries
CPU Overhead per Node	Assess processing load introduced by security

7.2.3 Network Performance Metrics

Table 7.5: Network Performance Metrics

Metric	Purpose
Average End-to-End Delay (normal case)	Measure latency under normal operation
End-to-End Delay under Sybil Attack	Evaluate impact when fake nodes are present
End-to-End Delay under Blackhole Attack (post-mitigation)	Assess recovery time after isolation
Network Response under Wormhole Attack	Verify immediate packet drop on tunnel detection
Optimal Secure Path Selection Success Rate	Compare chosen trustworthy path vs theoretical optimum
Packet Delivery Ratio (normal & attack scenarios)	Measure reliability of data delivery
Throughput (single flow and aggregate)	Quantify achievable data rate
Routing + Blockchain Control Overhead	Count extra control packets per second
Total Bandwidth Overhead	Measure percentage increase in traffic volume
Blockchain Synchronization Traffic per Block	Evaluate dissemination cost of new blocks
Convergence Time after Topology/Trust Change	Measure network stabilisation speed

These metrics collectively provide a complete, fair and quantifiable basis for evaluating both the security guarantees and the associated overhead of the proposed framework in FYP-II.

7.2.4 Conclusion

The prototype will demonstrate the functioning of security mechanism of the algorithm in the simulator, integerating communication, security, blockchain route logging, blockchain certificate management, and trust evaluation. The prototype will further deployed in Multi-PC environment in FYP-II.

Chapter 8 Experiment Results and Discussion

The chapter is separated into two sections. One section demonstrates the packets delivered with the modified protocol and the second one analyzes various security and performance metrics.

8.1 Experiment setup

The entire simulation was conducted using C++ simulator. Number of nodes were 20 and among which 7 nodes are malicious. The normal DSR algorithm was applied first. After that, the proposed protocol is applied on 20 nodes in different security threats. The entire simulation consisted of 200 rounds. The nodes distribution is visualized in 8.1.

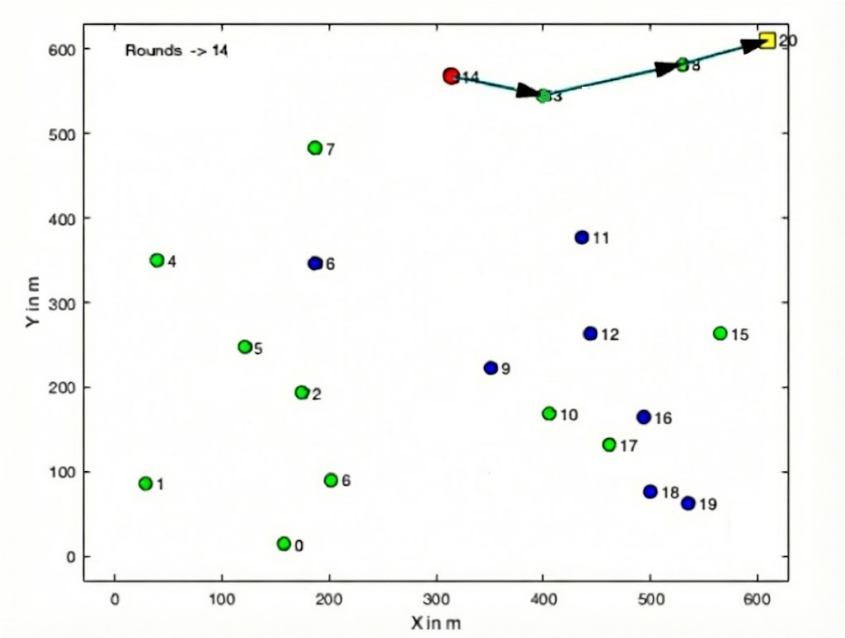

Figure 8.1: Node distribution scenario where blackholes (blue colored) are placed in different locations.

8.1.1 Contrast between normal DSR versus proposed blockchain DSR for attack incidence

Fig. 8.4 depicts the packets delivered in Sybil occurance, which was quite less as compared to standard DSR as the algorithm successfully detected the Sybil attack. Similarly, the modified DSR protocol proved to be a better approach after the wormhole and blackhole occurrence scenario. The rate of packets received was increased in both wormhole and blackhole occurrence as shown in Fig. 8.3 and 8.2 respectively, which is much better compared to normal DSR.

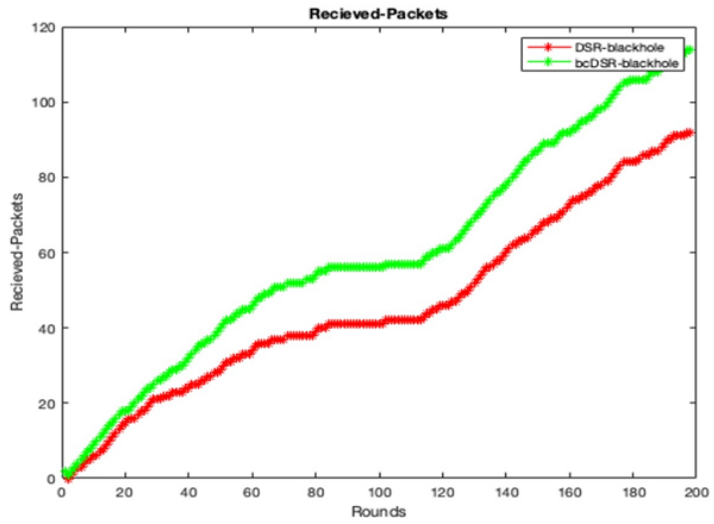

Figure 8.2: More Packets are delivered during blackhole as green curve represents the proposed algorithm and red is the normal DSR.

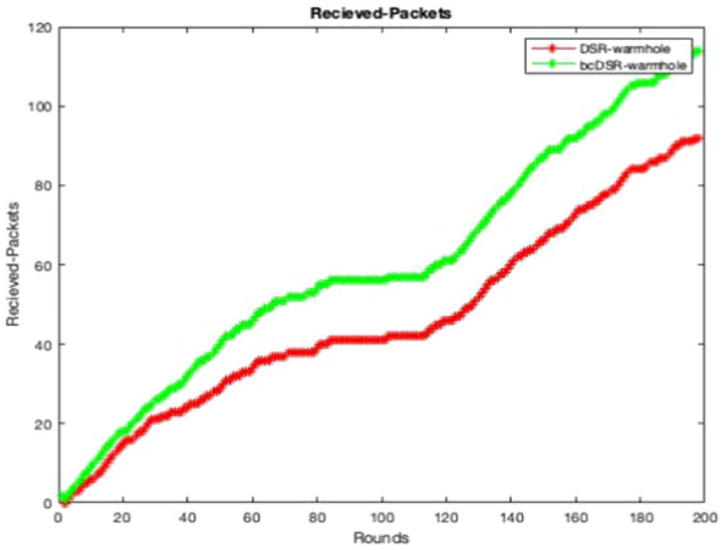

Figure 8.3: More Packets are delivered to host during wormhole occurrence v/s rounds.

8.2 Discussion

8.2.1 Impact of Recalculation on Performance Metrics

Frequent recalculation impact performance metrics such as end-to-end delay, packet delivery ratio (PDR), and scalability. 8.1 presents the analysis.

8.2.2 Comparison with DSR

The comparison in 8.2 analyze the security considerations, performance impacts, and associated computational costs observed under simulation.

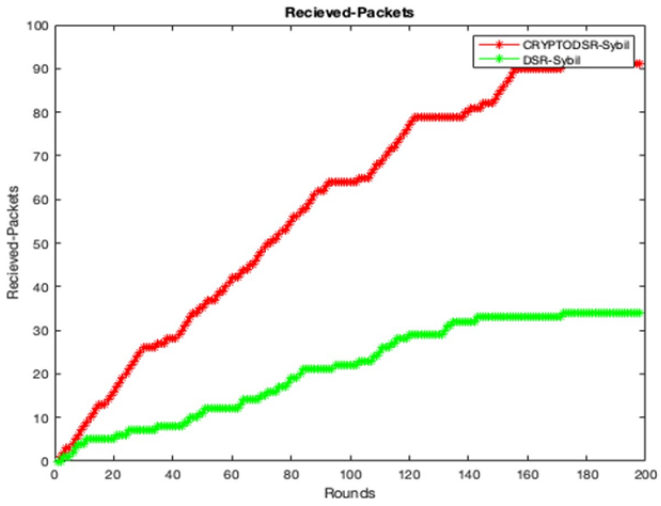

Figure 8.4: Less Packets are delivered during Sybil occurrence as shown in green curve.

Table 8.1: Impact of Frequency Recalculation on Performance Metrics

Metric	Impact of Frequent Recalculation
End-to-End Delay	• Route recalculations increase end-to-end delay due to rediscovering new optimal routes. • Blockchain validation contributes to processing overhead for route verification during each recalculation. • Delay would be more under mobility since frequent route failures and recovery attempts.
Packet Delivery Ratio	• Frequent recalculations can reduce PDR if the route discovery process cannot adjust with network changes. • Overhead from recalculation can lead to dropped packets if bandwidth or processing resources are constrained.

Table 8.2: Performance and Security of Standard DSR versus modified DSR protocol.

DSR protocol	Modified DSR
Throughput is a lot lower	Throughput is higher than ordinary DSR
The bundles shipped off get proportion is less	The parcels shipped off get proportion is higher
Energy utilization is low	Energy utilization is higher
Dead hub discovery is poor	Dead hub recognition is better
Security is poor	Security is given using Blockchain

8.2.3 Processing delay

Table 8.3 shows the average time overhead measured in various modules of the algorithm during the simulation.

Table 8.3: Average processing delays in various modules calculated during simulation

Metric	Operation	Value	Description
Processing delay	Block Creation	8ms	Avg. Delay incurred during block formation for routing updates.
	Encryption	6ms	Avg. delay per packet for encryption.
	Block verification	5ms	Lightweight consensus verification avg. delay per block
	Data transmission	3ms	Delay per packet due to blockchain metadata encapsulation.
	Data Reception	2ms	Delay per packet for decryption and processing of metadata.

8.3 Conclusion

The use of modified DSR handled dead nodes quite effectively. In today's infrastructure-less environments, the use of encrypted data packets has become vital. Moreover, during blackhole incidents, the proposed modified DSR has shown impressive performance, further reinforcing its role in maintaining the security. Additionally, the algorithm has modular structure, which means that in future, the algorithm has room for more enhancements and scalablity.

Chapter 9 Conclusion and Future Work

The chapter provides overall conclusion of the project including conclusion, core contributions, findings and achievements. Moreover, it clearly states the limitations of the project and future work of Final Year Project and industry-ready applications. The intention is to give final reflections for system enhancements and future improvements.

9.1 Overall Project Summary

Traditional protocols are efficiently transferring data but, vulnerable to different network security attacks like Sybil, Wormhole, Blackhole and Replay Attacks. To mitigate these attacks in a reactive protocol, a proactive-reactive hybrid approach is devised that provides security with optimized overhead on the existing DSR protocol. Three components of Security PKCertChain, RouteLogChain and Local Trust diaries provide separable, maintainable mitigation. The findings of our mechanism guarantees security and maintianing the health of network.

9.2 Achievements of Objectives

The core objectives of the ad-hoc are achieved as shown in the evaluation framework. Following are the list of achievements:

- Sybil Attacks are controlled by adding complexity in node registration mechanism and new certificates are generated based on security performance of the old nodes.
- Wormhole Attacks are controlled using path selection, path logging and rolling signature methods making it difficult to threat the network
- Blackhole Attacks are controlled by Local Trust diaries mainly by reducing the trust on the node and then isolate it from the network.
- Replay Attacks are controlled by Route logging and storing DATA packet in the blockchain and avoids network congestion
- A partial lightweight autonomic security mechanism that controls the synchronization and trust models using Bayesian inferences and Beta-Bernoulli Equations
- An event driven system to fulfill the proposed methodology, beneficial for resource constrained environment.

9.3 System Limitations

As of FYP-1, the following limitations of the system are provided:

- Single-PC simulations
- Synchronization are still costly need to be optimized
- Concurrency of Modules in the project.
- Blockchain overhead in terms of storage and performance in the network.
- The project is dependent on tunable parameters so careful tuning is required. The security can be fully autonomic and self-optimizing security prototype.

9.4 Goals for FYP-II

Now the next goal is to turn simulations into reality and for that we will be providing a comprehensive details for it. First of all our major goal is to shape-up as a deployable product and that can be use for any data exchange system with reliability and security. As we are using Linux as a host OS, we are designing an app and a visualizer for the FYP-2. Following are the key details for it split into two parts demonstration and application.

9.4.1 Demonstration

The core routing logic is further deployed into real world ad-hoc environment such that:

- 3-5 ad-hoc nodes
- Low mobility (pedestrian speed) demonstration for the justification of security mechanism
- Live attack and defense demonstrations
- live monitoring of the test metrics

9.4.2 Application

We are developing a proper Linux application named as DataExSys that will be deployable in the linux environment dynamically discover network and no need to configure. The application will integrate following modules:

- Secure Data exchange
- Real time monitoring

- Custom routing and trust mechanism as per Chapter 5.
- Honest and Malicious Roles assignments
- Blockchain Integerations

FYP-2 is breakdown into layers and layer-specific details for leveraging the potential of the Secure Blockchain enabled DSR security mechanism.

9.4.2.1 Frontend

The frontend of the application is a GUI layer made in Qt (C++). Qt is a cross platform GUI with C++ native integrations, supports async communications, and can render real-time packet visualizations and charts. Moreover, Qt is a mature framework widely used in research and industry for networking. Functional Overview for the frontend is given below:

- Login and Register Modules
- Sending and Receiving Data
- Sending and receiving user-specified data.
- Real time visualization of the security mechanism (Blockchain states, packets logging etc)
- Assigning own role as a Honest, Sybil, Wormhole, Blackhole and Replay roles for demonstration only so that the node itself behave according to scenario.

9.4.2.2 Middleware

The middleware is the orchestrator that binds the networking layer with the frontend for the app. We will be using Boost.Asio library for the orchestration and event loop managing networking layer. It will act as a dispatcher for the networking and blockchain modules. Boost.Asio is lightweight mature library for integeration. It provides access to kernel interface and raw sockets. It supports timer, I/O operations, threads, timers and co-operative multi-tasking via async handlers.

9.4.2.3 Networking Layer

This layer is currently implemented in the simulator in FYP-1. For practical development we need to expose it to kernel interfaces using AF_PACKET for raw sockets and it will be controlled by the middleware. It will capture custom packets, blockchain management, route table manipulation and DSR protocol.

9.5 Future Works

Following are the future works related to it:

- High mobility support
- DataExSys deployed into UAV, drones, VANETs and MANETs
- Scalability analysis and extension
- Machine Learning and AI integration for efficient routing and dynamics of network
- Extend this mechanism for the battery aware resource constrained devices i.e. Battery aware trust model for power-efficient secure routing.
- Extend it for advanced security threats like Colluding attacks, ML attacks and quantum attacks.
- Real-world heterogeneous nodes deployment.

9.6 Conclusion

The project shows the secure, modular and adaptive security system for low mobility PANETs under 50 nodes. Currently implementing the custom system effectively mitigates Blackhole, Wormhole, Sybil and Replay attacks. The project achieved its objectives by robust attack mitigation and asynchronous event driven autonomic operations and layered design for maintainability, modularity and scalability. FYP-2 aims for the linux-ready application that is practical and deployable real time monitoring data exchange system. The platform also provides support practical product and research testbed for future extensions.

Bibliography

[1] S. Majumder, D. Bhattacharyya, and S. Chowdhuri, "Abcd: Advanced blockchain dsr algorithm for manet to mitigate the different security threats," *EURASIP Journal on Wireless Communications and Networking*, vol. 2025, Feb 2025.

[2] S. Kudva, S. Badsha, S. Sengupta, H. La, I. Khalil, and M. Atiquzzaman, "A scalable blockchain-based trust management in vanet routing protocol," *Journal of Parallel and Distributed Computing*, vol. 152, 2021.

[3] C. Swamynathan, R. Shanmugam, K. Kumar, and B. Subbiyan, "Traffic prevention and security enhancement in vanet using deep learning with trusted routing aided blockchain technology," *Transactions on Emerging Telecommunications Technologies*, vol. 35, 2024.

[4] L. Morin and B. Princy, "Lightweight cryptography and ids for edge networks," in *Proc. Int. Conf. on Electronics, Communication and Aerospace Technology (ICOECA)*, pp. 107–112, 2025.

[5] V. Sugumaran, E. Dinesh, R. Ramya, and E. Muniyandy, "Distributed blockchain-assisted secure data aggregation scheme for risk-aware zone-based manet," *Scientific Reports*, vol. 15, 2025.

[6] B. Hou, Y. Xin, H. Zhu, Y. Yang, and J. Yang, "Vanet secure reputation evaluation and management model based on double layer blockchain," *Applied Sciences*, vol. 13, no. 9, 2023.

[7] H. Kamel and A. Abed, "Blockchain-enhanced secure routing protocols for vehicular ad hoc networks: A comprehensive review," in *Proc. Int. Conf. on Automation and Electrical Engineering (AUTEEE)*, pp. 624–630, 2024.

[8] N. Mouchfiq, C. Benjbara, and A. Habbani, "Security in manets: The blockchain issue," in *Advanced Communication Systems and Information Security*, pp. 219–232, Springer, 2020.

[9] J. Guo, H. Gao, Z. Liu, F. Huang, J. Zhang, X. Li, and J. Ma, "Icra: An intelligent clustering routing approach for uav ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 2447–2460, 2023.

[10] A. Mohindra and C. Gandhi, "A secure cryptography-based clustering mechanism for improving data transmission in manet," *Walailak Journal of Science and Technology*, vol. 18, 2021.

[11] X. Feng, K. Cui, L. Wang, Z. Liu, and J. Ma, "Pbag: A privacy-preserving blockchain-based authentication protocol with global-updated commitment in iovs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 10, pp. 13524–13545, 2024.

[12] R. Agrawal, R. Tripathi, and S. Tiwari, "Cluster based manet security with n-th degree truncated polynomial ring (ntru) public key cryptosystem," in *Proc. Int. Conf. on Computational Intelligence and Knowledge Economy (ICCIKE)*, pp. 126–133, 2023.

[13] Y. Chen, L. Meng, and J. Zhang, "Graph neural lasso for dynamic network regression," *arXiv preprint arXiv:1907.11114*, 2019.

[14] H. Jari, A. Alzahrani, and N. Thomas, "A novel indirect trust mechanism for addressing black hole attacks in manet," in *Proc. Int. Conf. on Computer Science and Engineering*, pp. 27–34, 2021.

[15] S. Patel and H. Pathak, "A regression-based technique for link failure time prediction in manet," *International Journal of High Performance Computing and Networking*, vol. 16, pp. 95–104, 2020.

[16] G. Prisco, "Filament develops ad-hoc mesh networks of smart sensors operating on the blockchain." https://bitcoinmagazine.com/business/filament-develops-ad-hoc-mesh-networks-smart-sensors-operating-blockchain-1435352121, June 2015. Bitcoin Magazine.

[17] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Lsb: A lightweight scalable blockchain for iot security and privacy," *arXiv preprint arXiv:1712.02969*, 2017.

[18] M. T. Lwin, J. Yim, and Y.-B. Ko, "Blockchain-based lightweight trust management in mobile ad-hoc networks," *Sensors*, vol. 20, no. 3, p. 698, 2020.

[19] B. Project, "Briar — secure messaging anywhere." https://briarproject.org/, 2025. Accessed on Briar website.