NAIST-IS-DD1561030

Doctoral Dissertation

Secure Protocols for MANET-Based Commerce System

Babatunde Ojetunde

March 15, 2018

Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Babatunde Ojetunde

Thesis Committee:

Professor Minoru Ito	(Supervisor)
Professor Keiichi Yasumoto	(Co-supervisor)
Associate Professor Naoki Shibata	(Co-supervisor)
Assisstant Professor Gao Juntao	(Co-supervisor)

Secure Protocols for MANET-Based Commerce System*

Babatunde Ojetunde

Abstract

Commerce system in a disaster area has the potential to provide electronic transactions for people purchasing recovery goods like foodstuffs, clothes, and medicine. However, to enable transactions in a disaster area, current payment systems need communication infrastructures (such as wired networks and cellular networks) which may be ruined during such disasters as large-scale earthquakes and flooding and thus cannot be depended on in a disaster area. In such a situation where the communication infrastructure is damaged, it is practically impossible to secure the commerce system or the routing protocol that may be adopted to route transactions against attacks. Furthermore, most existing secure routing protocols adopt a cryptography-based approach, trust-based approach (reputation of nodes), or incentive-based approach to detect and prevent such attacks. However, such protocols still have drawbacks, such as expensive overhead, difficulty in maintaining secure key and session management, or leaving routes unsecured against Byzantine attacks. Therefore, to address the shortcomings of the existing systems, a secure MANET-based commerce system is proposed.

In the first part of this dissertation, we introduce a new mobile payment system utilizing infrastructureless MANETs to enable transactions that permit users to shop in disaster areas. Specifically, we introduce an endorsement-based mechanism to provide payment guarantees for a customer-to-merchant transaction and a multilevel endorsement mechanism with a lightweight scheme based on Bloom filter and Merkle tree to reduce communication overheads. Our mobile payment

*Doctoral Dissertation, Graduate School of Information Science,
Nara Institute of Science and Technology, NAIST-IS-DD1561030, March 15, 2018.

system achieves secure transaction by adopting various schemes such as location-based mutual monitoring scheme and blind signature, while our newly introduced event chain mechanism prevents double spending attacks.

In the second part of this dissertation, we introduce a monitoring-based method in the link state routing protocol to secure the packets' route against Byzantine attacks. The goal of our proposed scheme is to guarantee communication among connected benign nodes in the network. Specifically, each node monitors the action of neighboring nodes and compares the optimal packet route against the packet route history. Nodes in the network create a packet history field which is used to record all activities of an intermediate node when receiving and forwarding packets. Our scheme provides mutual monitoring in which nodes in the network can validate the packet history field of other nodes and report malicious activities. Also, our scheme uses a statistical method to know if a node is dropping packets intentionally by analyzing the packet dropping behavior of each node. The proposed scheme provides protection against colluding attacks and other Byzantine attacks.

Keywords:

Payment system, endorsement, delegation, MANETs, bitcoin, routing protocol, routing attack, byzantine attacks, Link State Routing

Contents

List of Figures

1 Introduction

According to the 2016 World Disaster Report [1] carried out by the Centre for Research on the Epidemiology of Disasters (CRED), in 2015 a total of 108 million people were affected by disasters from 371 natural disasters reported worldwide causing deaths of 22,724 people while 9,826 people were killed by technological disasters. With frequent changes in global warming and climate change, it is even more difficult to predict patterns of disaster easily, which makes the regions that are not prone to disaster before to be experiencing one form of disaster or the other. Therefore, a more critical approach is still needed for disaster relief management.

There are four stages of a disaster [2] – Mitigation, Preparedness, Response and Recovery. The most critical period in the disaster is the response phase, which is the first 72 hours of a disaster, followed by the recovery phase which can last up to 6 months to 1 year or more. In the response phase, the focus is to address the immediate threat, e.g. saving lives, and meeting humanitarian needs. The recovery phase focuses on returning the disaster area economy back to normality. The recovery phase is further divided into two phases - short-term and long-term recovery. In the short-term recovery phase (which can last up to 6 months or 1 year), though the infrastructure is not fully restored, the people in a disaster area gradually begin to resume their normal existence which includes providing immediate services to businesses and being involved in a business transaction. The long-term phase, which can range up to decades, requires more strategic planning and action to address more serious or permanent impacts of a disaster.

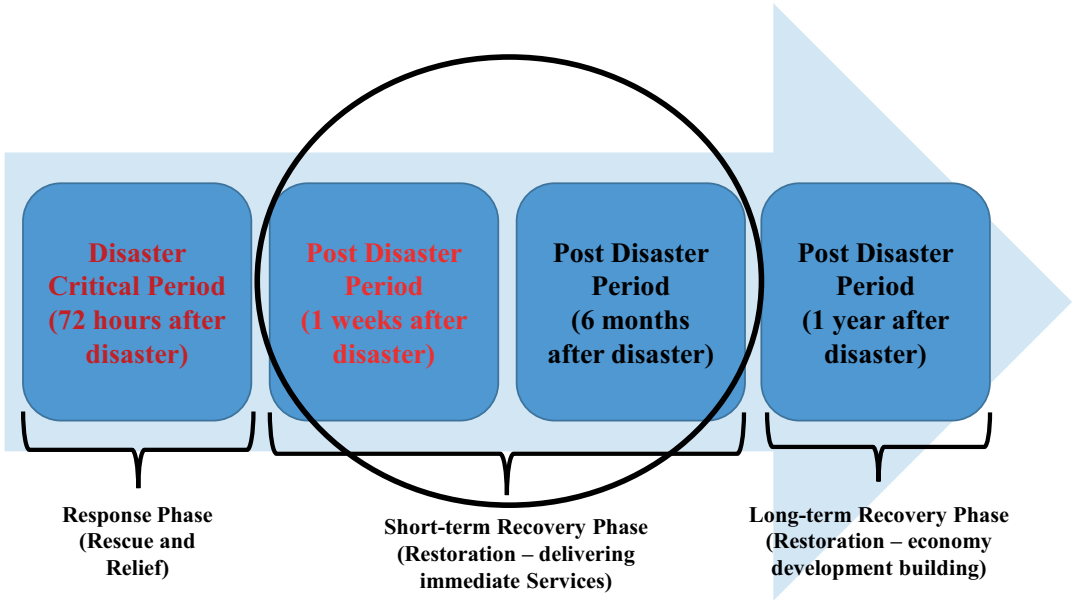

Figure 1.1: Phases of our Mobile Payment System in a disaster area.

Over the years access to financial services has played a major role in improving disaster relief management to ensure that there are enough provisions of evacuation centres, providing relief goods to people, or ensuring there is speedy recovery to social activities. Despite the huge involvement of the government and roles of private organizations in providing relief materials to the affected people, these supplies are often not adequate [3]. There are so many factors that contribute to the inadequate delivery of relief materials to the disaster areas, these include: fuel shortages, manpower shortages, telecommunication disruptions, power outages and little or no real-time information, etc. Also, physical damages and indirect effects of a disaster suffered by the financial sector contributes severely to the inadequacy of relief materials as the distribution of cash to institutions that needed additional cash for a disaster relief duty is also affected. As a result of the failure of the payment and settlement systems and the communication infrastructures during a large-scale disaster, people in a disaster area are not only prevented from making cash deposits or cash withdrawal but also from carrying out an electronic transaction. Therefore, a commerce system is essential for people in disaster areas to purchase recovery goods during the disaster recovery phase.

However, designing and developing such commerce system is difficult without the needed communication infrastructure to support the real-time transaction. Hence, our research proposes a mobile ad-hoc network (MANET) based commerce system. Figure 1.2 shows the phases in which a MANET-based commerce system can be deployed. In order to design and develop a commerce system that can be fully utilized in any environment, it is necessary to have a real-time connection to the bank or the central authority that manages the users' account and authenticates the users. However, this is not possible in a disaster situation where the communication infrastructure may be destroyed and there is no physical access to the bank. The first problem to consider in this research is how to establish communication among the users. As a result of the communication failures between the users and the bank or central authority, it is practically impossible to carry out a transaction in a disaster area. Hence, a MANET communication needs to be established for easy communication between the users.

Also, a delay or disruption-tolerant network (DTN) [20] can be utilized for communication between the users in a disaster area and the bank. In such a situation where network access is impossible, identifying each user is a serious challenge. This means that users can easily impersonate each other and such problem poses a lot of security issues to any payment system or payment transaction. A decentralized approach is needed to allow mutual authentication of users in a disaster area. Another area to consider is how to confirm the account balance of each user of the system. In the aftermath of a large-scale disaster, access to the bank is restricted due to the destruction of bank infrastructure and communication infrastructure. As a result of this restricted access, it is impossible to get physical cash, which a customer needs to pay for the item being purchased. Also, the crime rates in such an area in most cases are on the high side which makes it impossible for people to keep cash at home. Therefore, a mechanism to confirm the account balance of users is required.

In a MANET, a user can only send messages to another user when they are in the same transmission range of each other. The communication between the two users is not always guaranteed even though they are in the same range due to many factors such as power problems, network fading, etc. Therefore, a method of relaying messages through other users can be utilized to improve the message

throughput. Routing protocols that adopt relay of messages have been proposed for MANET, however, Byzantine attacks is still a major challenge in such routing protocols. In our commerce system, the second problem to consider is how to detect and prevent Byzantine attacks where users that are relaying messages deliberately drop the message.

Several studies have been carried out on mobile payment systems which, however, require the support of communication infrastructures to enable secure transactions and are therefore unsuitable for disaster areas without communication infrastructures. Li *et al.* [5] introduce an electronic payment mechanism that permits a payment transaction between a vehicle and a merchant when there is a limited connection, however, this mechanism needs a constant link from the merchant to the bank to complete the transaction, and cannot be used, therefore, to provide the needed services for people in a disaster area. Dai *et al.* [6] proposed an offline payment mechanism, that is used to buy digital goods. Their proposed mobile payment system adopts mechanisms from Dai's previous works, which introduced a debit-based payment protocol. Patil *et al.* [7] introduced an offline electronic coupon micro-payment system. Their scheme is based on credit and allows users to delegate their ability to pay for an item to another person device. The electronic coupon scheme delegation protocol is based on multi-seed payword chains. Their scheme focuses on minimizing the computational cost of mobile devices with limited resources. Similarly, Chen *et al.* [8] proposed a scheme that focuses on e-payment systems with electronic cash. To reduce a merchant's burden of having an account for depositing electronic cash received from customers with multiple banks. Chen's scheme introduced the concept of deposit delegation, which allows a merchant to maintain a single account at its trading bank: the system delegates all deposits from various banks into that account. Kiran *et al.* [9] introduced a payment system that uses a public-key and a cryptographic hash function to provide security for the transaction. In addition, the proposed payment system uses chains of delegates in which a customer can delegate the authorization to transfer money from the customer's account to other clients (to a vendor, for example). The system allows clients to carry out transactions both on-line and off-line.

Hu *et al.* [10], for example, proposed an online micro-payment system where

a customer can purchase goods from the merchant. To do this, a customer need to first send to the merchant a purchase request together with the payment authorization. In addition, the identity of users is confirmed indirectly, hence, customer's privacy is protected. However, the protocol can only handle one payment at a time, and relies on a trusted third party, which sometimes hinder the performance of the system. Wang *et al.* [11] introduced an electronic cash payment system which reduces the computational overhead of transactions. The computational cost reduction is achieved by integrating the trapdoor hash function into the system. Wang's payment system requires only integer multiplication and addition operations for computation, similar to [12, 13].

Chang *et al.* [14] focuses on an e-payment system by introducing a novel electronic check scheme to address the inflexibility of the electronic check proposed in [15, 16]. The scheme adopts cryptographic techniques such as a one-way hash function, a blind signature and RSA cryptosystems to protect the system against attacks. The scheme allows a customer to attach the cost of goods to be purchased and the merchant information to the electronic check during a transaction, thereby achieving mutual authentication by the customer and the merchant. Liaw *et al.* [17] also adopted a similar concept to Chang's electronic check mechanism to introduce an electronic traveler check scheme that is capable of handling an offline/online transaction. However, Liaw's scheme, unlike Chang's electronic check, adopts a one-way hash function which improves performance and reduces the cost of the system. The customer ID is added to the traveler's check to prevent impersonation of the customer by other users. Dahlberg *et al.* [18] survey several existing mobile payment systems and suggests the basis for evaluating the mobile payment study. Furthermore, concerning several gray areas, they propose solutions on which, they suggest, future mobile payment research should be centered.

Nakamoto [19] introduced a distributed e-cash system known as Bitcoin that does not depend on a central authority. In the system, a new transaction is transmitted to the entire network, and each node receives the transaction into a block. Then each node attempts to perform a reverse calculation of a hash function as proof-of-work to verify the transaction in their blocks. (The verification procedure is called mining, and each miner are compensated for each block

verified). This calculation takes a large amount of computation. Nodes receive a block only if the transactions are genuine and if the Bitcoin has not been used in the previous transaction. The hash of a received block is used in the next block to form a block chain, and with this, all users can agree on the sequence in which transactions occurred. However, Bitcoin requires a device with high power, and transactions are computationally irreversible, so that Bitcoins can never be replaced if a user's private key is forgotten or destroyed.

Our approach differs from related work in the following points: We introduce a secure payment system that utilizes infrastructureless mobile ad-hoc networks (MANETs) to permit users to buy recovery goods in disaster areas. Also, we propose a mechanism that ensures that double spending will be detected before a transaction is completed, unlike existing systems that detect double spending only when e-coins are deposited in a bank or deducted from a customer's account. Our proposed system uses an approach comparable to that of Bitcoin in that transactions are stored in the block chain. However, our method differs in its techniques, since users in our system do not need proof of work. Rather, users calculate the hash value of a transaction log, and neighboring nodes append their signatures to the log to form an event chain (similar to a block chain). The event chain can be verified by surrounding neighboring nodes. Unlike most existing payment systems, our proposed mechanism does not depend on a central authority or mint to detect double spending.

Additionally, several methods on routing protocols have been proposed. Geetha *et. al.* [29] classified routing protocols into three distinct types: proactive, reactive, and hybrid protocols. They described proactive protocols as protocols where nodes frequently exchange network topology information and construct routing tables to send packets from the source to the destination. Examples of such protocols include the Optimized Link State Routing protocol, and the Destination-Sequenced Distance-Vector protocol. Reactive protocols are described as protocols that ensure packets are sent from the source to the destination only when needed. Ad-hoc On-Demand Vector (AODV) and Dynamic Source Routing are examples of reactive protocols. Finally, hybrid protocols are produced by combining both proactive and reactive protocols. For example, route discovery makes use of a proactive protocol scheme while a reactive protocol scheme is adopted

for sending packets. The Zone Routing Protocol and Fisheye State Routing are both examples of hybrid protocols.

Harshavardhan [30] surveyed security issues in ad-hoc routing protocols and identified ways to mitigate such security threats. Harshavardhan first defined the properties of an ad-hoc routing protocol as providing distributed operation, loop free, demand-based operation, unidirectional link support, security, quality-of-service support, multiple routes, and power conservation. Then they used findings from related work to summarize different ad-hoc routing protocols before analyzing various security threats and techniques to mitigate them. Some of the security threats they included were: impersonation or spoofing, black-hole attack, sinkhole attack, and wormhole attack. They classified solutions to these attacks into categories including: trust values, wormhole detection method, intrusion detection systems, credibility management and routing test, and multi-factor authentication techniques.

Ali *et. al.* [31] also surveyed security challenges in mobile ad-hoc networks (MANETs). They introduced three important security parameters, and further divided security aspects into two areas, which are security services and attacks. They classified security services into five important services which are used to protect the network before attacks happen, while attacks are the threats to the network. In addition, they analyzed and discussed various mitigating approaches against attacks in MANETs. Mojtaba *et. al.* [32] also investigated routing attacks and various solutions to such attacks. They highlighted security attacks that MANET routing protocols are vulnerable to and identified mechanisms such as cryptography schemes, key management, and special hardware using GPS as some possible solutions to such attacks. Similarly, Kannhavong *et. al.* [33] surveyed routing attacks in MANETs. They investigated various security issues in MANETs and examined routing attacks, such as flooding, black holes, wormholes, replays, link spoofing, and colluding attacks, as well as solutions to such attacks in MANETs. They identified the advantages and drawbacks of the reviewed solutions, then recommended improvement of the effectiveness of the security schemes they had surveyed.

Jhaveri *et. al.* [34] surveyed various DoS attacks that are security concerns in MANETs and some of the proposed solutions to identify and prevent such attacks.

They describe various routing protocols, and DoS attacks such as a wormhole, black hole, gray hole attacks and their operations. Zapata *et. al.* [35] introduced a security mechanism to secure AODV routing information. First, they identified integrity, authentication, confidentiality, and non-repudiation as security goals for routing. Then they proposed two mechanisms to secure AODV packets, hash chains and digital signatures. Specifically, the hash chain is used to verify that the hop count was not decreased by a malicious node, while the digital signature is used to safeguard the integrity of other information in the packets besides the hop count.

Alajeely *et. al.* [36] proposed a new detection scheme for malicious nodes to detect packet faking by a malicious node. In this type of attack, malicious nodes drop one or more packets and inject another packet to replace the dropped packet. They introduced a hash chain technique to detect the attack and trace the malicious nodes. They compared their approach to an acknowledgment-based mechanism and a network coding based mechanism. Baadache *et. al.* [37] proposed a scheme to check if packets are routed correctly in the network. They adopt the acknowledgement of packets at each intermediate node which is used to construct a Merkle tree. Packet dropping is detected if the root of the Merkle tree is not the same with a precalculated value.

Papadimitratos *et. al.* [38] proposed a secure link state protocol (SLSP) for MANETs to secure neighbor discovery and adopted a neighbor lookup protocol to further strengthen their system against DoS attacks. In addition, the proposed SLSP restricted the forwarding of packets within a cluster, and adopted the use of public and private keys to validate that the packets are only forwarded within the cluster. Unlike our proposed monitoring scheme, their protocol only focused on securing the topology discovery and protected the link state update packets, but did not secure the routing of packets. Our proposed scheme addresses routing security using a monitoring mechanism to protect packets and also guarantees the communication of benign nodes. Another main difference found in our work is that our proposed scheme secures the routing protocol against colluding attacks where a group of nodes collaborates to carry out an attack.

To secure the packet route and provide secure message transmission in MANETs, Papadimitratos *et. al.* [39] proposed a different mechanism from their previous

work. Their mechanism is based on four main schemes: secure end-to-end transmission of packets and feedback, dispersion of a packet, multi-path routing of packets, and adaptation to topology changes. In their protocol, the source node will first select several disjointed paths that are valid, referred to as an active path set (APS). Then the node splits the packet into a number of pieces, which are transmitted simultaneously across the selected APS. After receiving a sufficient number of pieces of the divided packet, the destination node will then reconstruct the packet, even when some fraction of the pieces are dropped or invalid. Whenever a piece of the packet is not received by the destination, that route is considered broken or compromised. In addition, their mechanism also introduced path rating based on feedback from the destination node. Paths that fall below a given threshold are discarded from the network. Their secure protocol focused on detecting unsecured routes, unlike our approach in which the actual malicious nodes in a selected route are detected and discarded to prevent further relaying of packets.

Although some of the proposed schemes successfully mitigate routing attacks, they are either too expensive for resource-constrained networks or the solution provided is not applicable to mitigate colluding attacks from malicious nodes. Also, it is possible for malicious nodes to drop packets and attribute the cause to poor communication links. Therefore, we propose a mechanism to analyze the action of all nodes in the network. Specifically, our scheme focuses on mitigating Byzantine attacks in link state routing protocols.

1.1 Research Contribution

In this section we alight the contribution of our dissertation which we developed to solve the issues mentioned above.

The first contribution of our dissertation is the introduction of a secure payment system that adopts infrastructureless mobile ad-hoc networks (MANETs) to allow users to purchase necessities in disaster areas. We introduce an endorsement-based mechanism to provide payment guarantees for a customer-to-merchant transaction and a multilevel endorsement mechanism with a lightweight scheme based on Bloom filter and Merkle tree to reduce communication overheads. Our

proposed secure payment system adopts various schemes such as location-based mutual monitoring scheme, blind signature, and event chain mechanism to prevent double spending attacks. The event chain mechanism ensures that double spending is detected before a transaction is completed and instead of when the e-coin is deposited in the bank or deducted from the customer's account.

The second contribution of our dissertation is the introduction of a monitoring-based method in the link state routing protocol to secure the packets' route against Byzantine attacks. First, we investigate various routing protocols and security challenges. Then proposed a monitoring approach to secure the routing of transactions in commerce system against Byzantine attacks. Our goal is to guarantee communication among the users of the commerce system in a disaster area and this can only be achieved if the routing protocol that allow the relaying of messages from one user to another is secured. We adopts three main methods: (i) Hello message verification, (ii) Packet history field monitoring, and (iii) Statistical hypothesis testing.

1.2 Research Scope and Limitation

The scope of this research includes the design and development of a secure commerce system that can be utilized in a disaster area to purchase recovery goods during the disaster recovery phase. The first part includes the endorsement-based mechanism which guarantees that a merchant can get paid for every transaction carryout by the users of the system. Also, introducing of various schemes such as mutual information-based monitoring, event chain, blind signature, etc. which are adopted to secure the payment system.

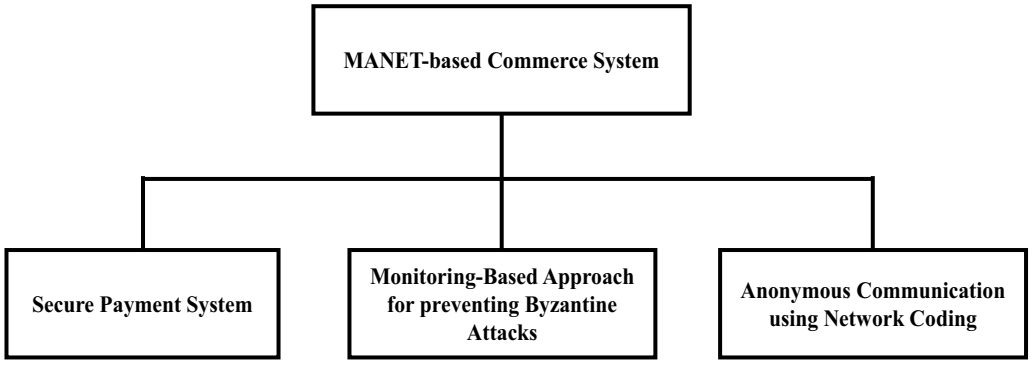

Figure 1.2: Research scope.

The second part includes the introduction of a monitoring-based method to detect Byzantine attacks. Routing protocols is essential to our system as each transaction needs to be relayed from one user to another before reaching the merchant. The monitoring-based approach prevents the transaction from been selectively or completely dropped by an attacker. The simulation results includes transaction completion of the proposed systems. The research is limited and does not include the actual implementation of the payment system on real devices such as iPad.

1.3 Dissertation Layout

The rest of the dissertation is organized as follows:

Chapter 2 introduces the secure payment system. We describes the design and schemes needed to achieve payment transaction in a disaster area. Chapter 3 proposed a monitoring-based method for securing wireless networks against Byzantine attacks. In this chapter, we discussed the overview of Byzantine attacks and various types of Byzantine attacks. Also, we present ways to prevent such attacks with our proposed method. Finally, Chapter 4 concludes the whole dissertation.

2 Secure Payment System

2.1 Background

Large scale disasters have a major and lasting social and economic impact on people, causing damage that leads to loss of human life, materials and massive economic loss. One of such impact is leaving people in a disaster area without cash-at-hand to purchase necessities like foodstuffs, clothes, and medicine. Although real cash is considered to be the easiest means for carrying out a transaction, it may be impossible to get cash in a disaster situation since access to a bank is restricted both physically (roads may be blocked or the bank destroyed) and electronically (communication infrastructures, like wired networks and cellular networks, may fail due to an earthquake or flooding). Furthermore, existing payment systems require such communication infrastructures for transactions in a disaster area. To enable people to do transactions even in a disaster area, therefore, of vital importance to people in disaster areas is an infrastructureless mobile payment system which can utilize flexible and robust mobile adhoc networks (MANETs) formed via the widely used smart mobile devices (smart phones, etc.).

Furthermore, several payment systems are developed to provide electronic currency services, but none has been specifically created to solve the payment challenges faced by the people in a disaster area. The proposed system is also capable of providing such services, however, since there is no access to the bank in a disaster area, the use of electronic currency for online transaction is restricted. Therefore, our secure payment system is centered on enabling offline transactions utilizing MANETs. In designing such a MANET-based payment system, the following challenges [4] should be considered:

1. *Frequent network disconnection* - One of the characteristic of MANET is

low-power supply, this can impede a constant connection between users.

2. *Persistent change in topology* - Topology changes quickly in MANET as a result of node's mobility in the network. Thereby leading to a decrease in performance.

3. *Inadequate security* - Secure characteristics of wireless networks are lacking in MANETs; this increases the flaws of MANETs to attacks.

In this chapter, we propose a mobile payment system that utilizes self-organized MANETs to enable people to carry out a transaction in disaster areas. The main contributions of this chapter are summarized as follows.

- First, we propose a new mobile payment system to allow electronic commerce in disaster areas, in a situation where the bank is not accessible.
- Second, we introduce an endorsement-based scheme to provide a merchant payment guarantees for a customer using multilevel-endorser scheme to sufficiently cover transaction amount.
- We introduce a transaction-log-checking scheme (called event chain) to prevent double spending attack before a transaction is completed. In addition, we propose an electronic money scheme (called e-coin) for account balance checking and to prevent a predetermined number of parties (N_c) from colluding.
- We also adopt a light-weight scheme, based on techniques of Bloom filter and Merkle tree, to reduce communication overheads.
- Additionally, we introduce a mutual tracking mechanism that can proof that transaction are valid and reliable.
- A digitally signed photograph is proposed for authentication and to restrict an attacker from carrying out a fraudulent transaction and impersonating others.
- Furthermore, we adopt a blind signature technique to protect user's privacy by ensuring that each user uses different temporary IDs in every transaction.

- Finally, we evaluate the performance of our proposed secure payment system by simulation to test the usability in disaster areas. Our simulation focused on: the ratio of successful transaction completions, merchant communication overhead, the validity ratio (VR) of event chain, the size of an event chain and the effect of various parameters such as endorser density, mobility speed of nodes and density of monitoring nodes on the transaction completion ratio (TCR). Our simulation results showed that the TCR increased significantly by an average of 48%, 28% and 22% using 100, 200 and 500 mobile nodes, respectively.

2.2 Proposed System Overview

Our proposed payment system adopts two operation modes: the first mode is the Internet mode, which functions like every normal online payment system and it is used when there is no disaster. The second mode is the MANET mode, which is used in a disaster situation. When there is a disaster, the system automatically switches the operation mode from the internet mode to MANET mode. Since our goal is to allow people in a disaster area to access essential amenities, we will focus on the disaster mode of our payment system.

In payment systems, successful transaction completion is essential, however, this cannot be achieved if there is no communication between the users, merchant and the bank. This is the case in a disaster area where the communication infrastructure may be destroyed and access to the bank is cut off both physically and electronically. Therefore, the first aspect of our payment system for the MANET mode is to establish a means of communication among users in a disaster area. To achieve this, we adopt infrastructureless MANETs and DTN-based communication (the communication process is explained later in Section 2.6.

Then, the next aspect is to establish a means of identifying users and confirming if there is enough account balance to pay for an item since there is no direct connection to the bank during transaction. Therefore, we introduced various schemes such as digitally signed photograph and e-coin to achieve this. Also, it is impossible to get physical cash, which a customer needs to pay for the item being purchased since access to the bank is restricted due to destruction

of bank infrastructure and communication. Hence, a customer cannot make a direct transaction with the merchant. Detailed implementation of these schemes is explained later.

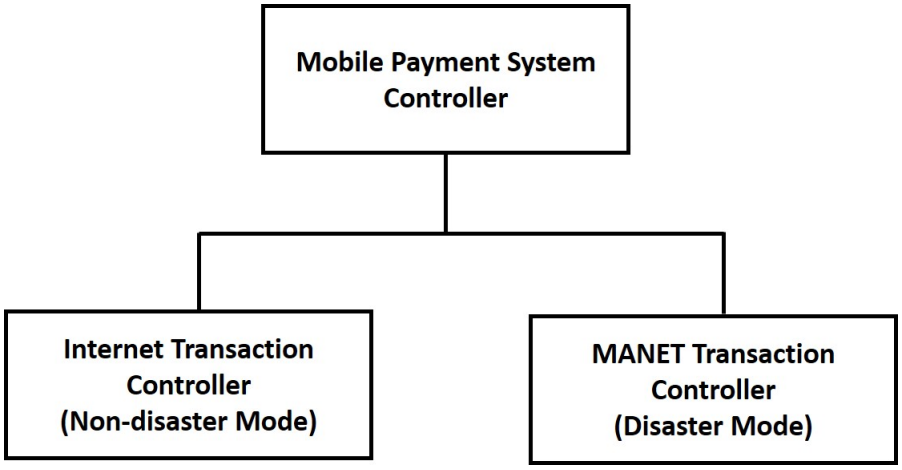

Figure 2.1: Mobile payment system controller.

2.3 Preliminaries

In this section, we introduce entities involved in our payment system, user registration and authentication processes, system assumptions and a purchase example in a common payment system.

2.3.1 Participants

All entities (customer, endorser, merchant, and bank) that join and are involved in the payment system will be referred to as users. All users communicate through MANETs.

- **Merchant** - A user that provides goods.
- **Customer** - A user that buys goods from a merchant.
- **Endorser** - A user that agrees in advance to make payments for the customer, if the customer fails to pay.

Table 2.1: Proposed System Keys

User	**User's Identity**	**Public Key**	**Private Key**	**Digital Signature**
Bank	B	K_B	K_B^{-1}	$S_{K_B^{-1}}$
Merchant	M	K_M	K_M^{-1}	$S_{K_M^{-1}}$
Customer	C	K_C	K_C^{-1}	$S_{K_C^{-1}}$

- **Monitor** - A customer that audits every transaction within the radio range to make sure that each message is valid and reliable.
- **Bank** - An organization that maintains users' accounts.
- **Delivery Truck** - A truck used for delivering items to the customer. Also used to pass messages between the bank and the users (endorsers) in a disaster area every two days.

2.3.2 Registration

To join the system, customers and merchants register with the bank before a disaster occurs. Each user generates its public and private key pair, then sends only the public key to the bank. The bank is the only trusted party among all the entities involved in the payment system, hence, acts as a certification authority and set the key expiration which can be as long as specified by the bank. Introducing a separate third party to carry out this function will introduce a bottleneck in the system as all users will need to communicate with this third party and since the bank is not available in the disaster area, thereby introducing more overhead in the system. Hence, paying the merchant for a transaction will be difficult. The private key is kept secret by each user. The notations for a user's public and private keys are shown in Table 2.1.

The registration process in our system can be divided into three stages: merchant registration, customer registration and endorser selection. This registration process takes place before disaster happens.

Merchant registration

A merchant submits a registration request to the bank to join the mobile payment system. Then the bank accepts the registration request and a public from the merchant.

Customer registration

A customer submits a registration request to the bank to participate in the mobile payment system. Then the bank accepts the registration request and a public key for the customer. The customer selects a photograph and requests the bank to sign the photograph with the bank's digital signature. The bank signs the customer photograph with the bank's digital signature.

Endorser selection

Each customer personally selects his/her endorsers. To select an endorser, the customer submits the list of users that will serve as his/her endorsers in the system before disaster occurs (these endorsers are only used for MANET mode transaction). If a user agrees to endorse other specific users, the user deposits real money in the bank. Since there is no direct connection to the bank (both electronically and physically) in a disaster area, the deposited money need to be converted to electronic coins which is used in a disaster area to confirm if an endorser has sufficient money to endorse other user's transaction when the purchase of an item is initiated. The bank generates electronic coins equivalent to the amount deposited by the user (now as endorser).

2.4 Providing Authentication and Security

In an online payment transaction, the customer identity is verified real-time via the bank, and access to the payment system is allowed providing the verification is successful. A customer cannot be impersonated without an attacker knowing the customer's information. In a disaster area, verifying a customer's identity is currently difficult as a direct link to the bank is not accessible, as a result of the lack of a communication infrastructure.

In our system, each customer chooses a photograph that will be digitally signed by the bank, which is used to verify a customer's identity during a transaction and protects the customer when an attacker stole their phone (Which is similar to checking an individual photograph on an ID card, moreover, in our system the merchant will also check the digital signatures of the bank and the customer which is on the photograph). Another form of biometrics authentication mechanism may also be used.

To further secure transactions, each message is digitally signed and encrypted. Thus achieving nonrepudiation of transactions. In addition, a monitor can audit every transaction and thus detects an attacker in the network.

2.5 Assumptions

We make the following assumptions about our mobile payment system.

- Fewer than a predetermined number of parties (N_c) collude to commit fraud.
- Users are identified by digitally-signed pictures.
- Most of the users do not power off phones very often. This is to discourage users from deliberately switching off their phones in order to carry out an attack.
- Most of the phones owned by legitimate users do not share similar location histories, as their global positioning system (GPS) coordinates are error bound with a 4.9 - 10 meter range of each other.
- Node density is sufficient in most of the locations.
- Users can use GPS in almost every location, i.e., we adopt the use of normal GPS for accessing users GPS position since the A-GPS and other positioning technologies used to improve GPS accuracy cannot be accessed due to the destruction of cell towers when disaster happens.
- An attacker is not quick enough to get the needed information from the system before the event chain is invalidated (a scheme explained later).

- A user can access a bank using the DTN-based communication formed via the delivery truck at least every two days.

2.6 Communication Model

In a disaster area, a delay/disruption tolerant network (DTN) [20] can be used in addition to a MANET formed among user nodes. The DTN communication can be achieved when two nodes in close proximity to each other communicate. Using the store-carry-and-forward technique, a node stores a message temporarily and forwards the message when the node comes across another node. For the DTN in a disaster area, our approach uses smart phones of users and the delivery truck to form such a network.

2.6.1 Our Network Model

Since there is no direct communication to the bank as a result of the destruction of the existing communication infrastructure and users in a disaster area are characterized by limited resources (such as bandwidth), it takes several days for users messages to get to the bank. We assume that customers and endorsers are in close proximity to a merchant. Therefore, we adopt a network with a communication range of 100 m between the users in a disaster area (that is, customers, endorsers and merchant). A minimum of six (6) nodes (i.e. one customer, one endorser, three monitors and a merchant) are required to complete a transaction successfully and the six nodes are present within this communication range. When a user sends a message to the merchant, the message is store-and-carry-forward by the intermediate node between the customer and the merchant. In addition to a MANET formed, we introduced DTN-based data dissemination and collection via the delivery truck to transmit messages to/from the bank for the users and the merchant. Each delivery truck moves from the nearby reservoir and cover regions one after the other. The delivery truck is used to deliver items to the merchant in the core disaster area from the nearby reservoir and data moves with this truck. Therefore, with the DTN formed, multihop data transfer is possible and communication formed by the truck to the bank in a nonaffected area is established.

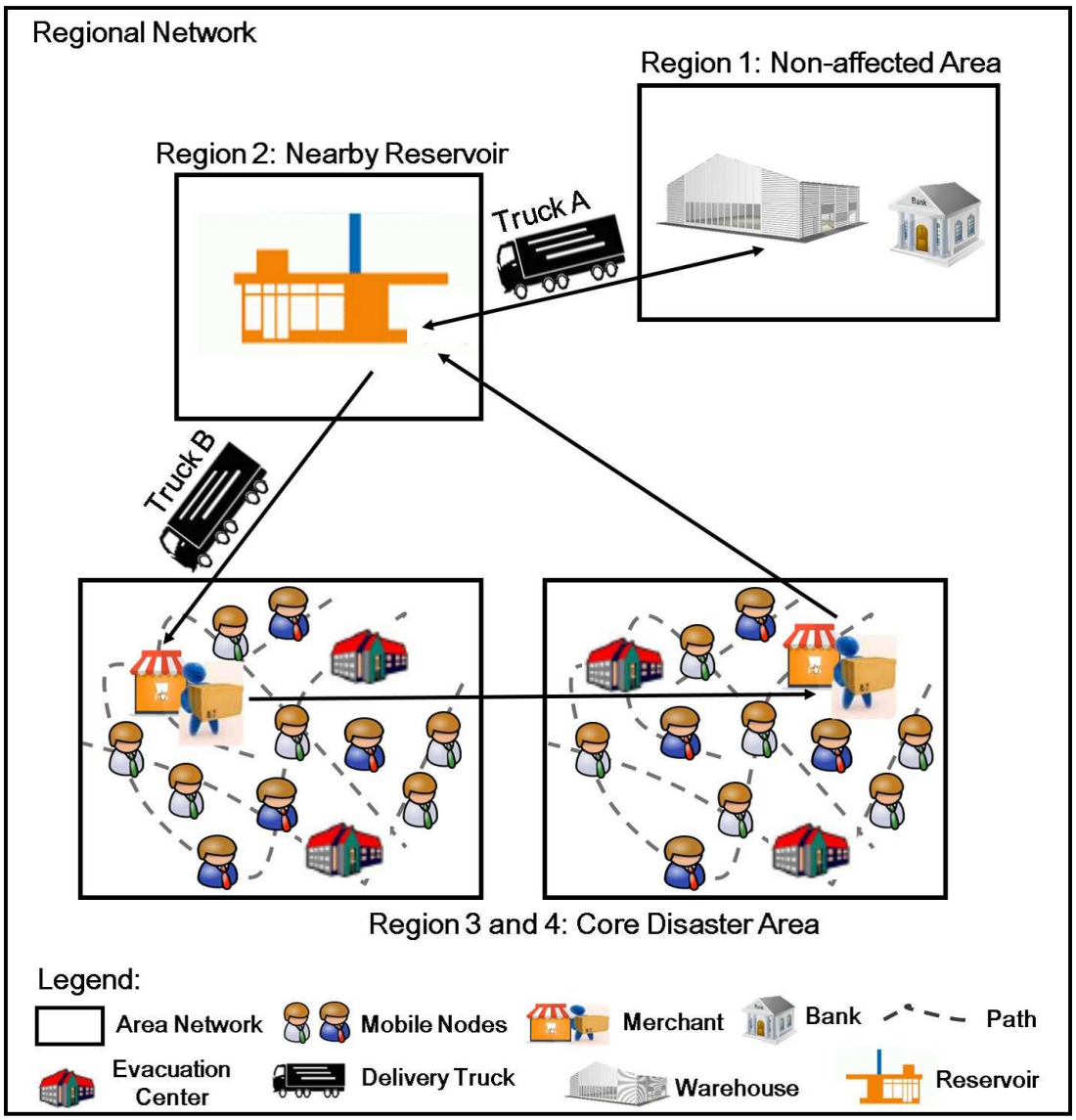

Figure 2.2: Example of regional communication network.

2.7 Payment System in Areas without Disaster

In areas that are not affected by a disaster, the customer and merchant can connect directly to the bank using the wired or wireless networks. The steps to purchase an item in such a payment system is illustrated below:

1. The customer broadcasts a transaction order to purchase an item from the merchant, (for example, an apple that costs $20).

2. The merchant verifies the customer's identity and forwards the billing message to the bank, (for example, customer C requests to purchase an apple that costs $20).

3. The bank confirms the customer's account balance and accepts the transaction if the balance is enough to cover the cost of the transaction. Then withdraw the equivalent cost from the customer's account and inform the merchant to supply the item. However, if the account balance is not enough, the bank rejects the transaction.

4. The merchant delivers the item to the customer.

5. The transaction amount is paid to the merchant, then the bank sends transaction completion notification to the customer.

This approach will be unsuccessful in a disaster area due to the following reasons:

1. **Inaccessible communication infrastructures**.

2. **Inaccessibility of a bank**.

3. **Fraudulent transactions and impersonation**.

4. **Security/Authentication Issues** | Real-time verification of user's identity is impossible in disaster areas due to the lack of a communication infrastructure.

To provide a solution to these problems and ensure that there is a payment system that can function in a disaster area, we propose a secure payment system based on endorsement and adopt various mechanisms to secure the proposed system.

2.8 Secure Mobile Payment System Based on Endorsement

In this section, we first introduce the concept of endorsement and then give detailed explanation of our secure payment system for disaster areas. Our system provides payment guarantees to the merchant in a disaster area where there are no network infrastructures nor direct access to a bank.

2.8.1 Endorsement

In our system, an endorsement is a mechanism by which the endorser agrees in advance to make payments for the customer, if the customer fails to pay. For this, the endorser should have real money deposited in a bank beforehand. An endorser agrees to serve directly as a customer's endorser by signing an endorsement agreement, thereby personally guaranteeing the customer's transaction and pledging to make payment for up to the amount deposited by the endorser for every transaction in which the customer defaults in payment. The endorsement agreement comes with the two conditions that (1) the real money deposited in the endorsement account will be restricted (locked) to endorsing a customer (the locking of the account is effected when the mode of our system is switched from the Internet mode to MANET mode) and (2) the amount endorsed for any transaction has a limit. The endorsement agreement is made during registration prior to a disaster.

In the proposed method, a minimum of one endorser can successfully endorse a transaction as long as that endorser can cover the payment for the transaction amount. However, to avoid a situation where the endorser is not able to pay for the transaction amount which would lead to a shortage of money to pay the merchant. Therefore, we allow a customer to have multiple endorsers to guarantee each transaction so that the endorsement liability for one transaction is shared among all endorsers. In this way, the risk of endorsing is reduced if a customer purchases an item, but then defaults. To motivate endorsers to cooperate and support the mobile payment system, some part of the transaction amount (e.g., 3%) is shared among the endorsers as incentives. The percentage of the transaction amount to be used for the incentives is agreed between the bank and

the merchant when the merchant joins the mobile payment system. In addition, we introduce multilevel endorsement (MLE) where an endorser delegates its endorsement capabilities to its own endorser. Each user indicates if they want to participate in such MLE at registration, which is before disaster happens. In the MLE, when an endorser inherits a transaction from users it normally endorsed, it does so using the exact same endorsement amount agreed to for such user. For example, if user A is an endorser to user B, and user B is an endorser to user C. Using the MLE when user A inherit the user *C's* transaction, for the endorsement to be completed, user A needs to sign its signature to show its intention to guarantee the transaction. User A uses the actual endorsement amount that is agreed for endorsing user B to endorse user A. In the MLE, each user inheriting a transaction still needs to append its signature on each endorsement. Any transaction without endorsement (i.e., there are no primary endorsers or secondary endorsers) is rejected by the merchant.

2.8.2 Starting transaction after disaster

Once a disaster occurs, a customer and a merchant in close proximity agree to begin a transaction; the users and the merchant meet to establish a connection by exchanging IDs and pictures. The customer sends his/her photograph to the merchant for identification. The merchant compares the photograph with the customer's actual appearance. The merchant also confirms the digital signature of the bank on the photograph. When a customer tries to purchase an item, the exchanged picture is used to identify a customer. The merchant verifies the bank digital signature, timestamp and the customer's digital signature on the picture. The same procedure is used by all users in the network to identify each other.

2.8.3 Transaction Process

Through the endorsement mechanism, we realize a mobile payment transaction in a disaster area even when there is no direct access to the bank. For example, let us consider a scenario where an endorser D decides to endorse customer C as shown in Figure 2.3. The minimum node density required to complete a transaction is six nodes (one customer, one endorser, three monitors and a merchant). The

process for customer C for buying an item from a merchant using an endorsement mechanism is illustrated below:

- **STEP 1:** Customer C broadcasts a transaction order message to purchase an item from the merchant, (for example, an apple that cost $20). The transaction order message contains a transaction order form, customer *C's* temporary ID, the merchant's ID, the endorser's ID, the bank's ID, the item number, the item quantity, etc.

- **STEP 2:** The merchant checks customer *C's* ID (through a digitally signed photograph) and generates a billing message. However, since there is no definite process of confirming customer C account balance, the merchant forwards the billing and transaction messages to the endorser, to request that the endorser provides payment security the transaction.

- **STEP 3:** The endorser checks the merchant's ID and customer *C's* ID and generates an endorsement message, signifying that he/she will provide payment security for the transaction by signing the endorsement message with his/her signature. The endorser sends the endorsement message, billing message and transaction order message to the merchant, stating for example, "I agree to provide payment security for customer *C's* transaction of $20".

- **STEP 4(a):** The merchant checks the endorser's ID and customer C's ID and sends all messages to the bank if the IDs are valid. These messages take two days to reach the bank as there is no direct communication to the bank as a result of the destruction of the existing communication infrastructure. The messages are transmitted from the merchant to the bank using the multihop data transfer through the delivery truck DTN-based data collection.

- **STEP 4(b):** After sending all the messages to the bank, the merchant immediately supplies the item to customer C. The merchant will receive the payment as the transaction is endorsed by endorser D.

- **STEP 5(a):** The bank checks the ID of all users and if other information

provided is genuine. A few days later, the bank checks the account balance of customer C and withdraw the transaction amount ($20).

- **STEP 5(b):** The equivalent amount of $20 is paid to the account of the merchant.
- **STEP 5(c):** However, in an instance where customer *C's* account balance is not sufficient to cover the transaction cost, the transaction amount is taken from endorser *D's* account.

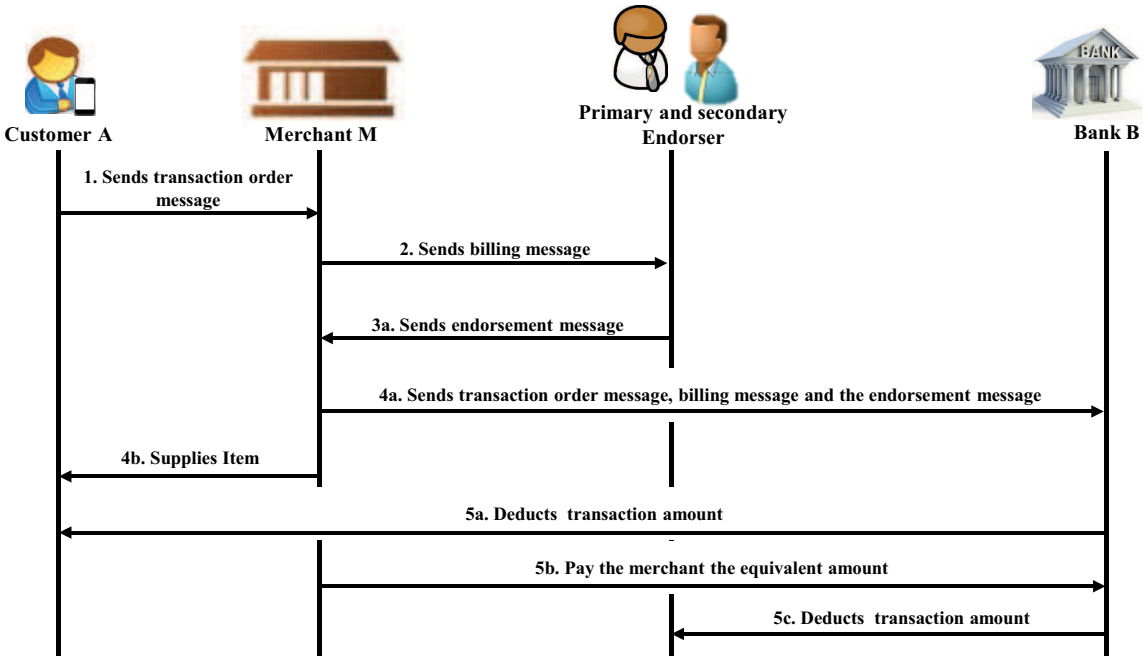

Figure 2.3: Transaction flow.

To prevent unfair-exchange, we adopt transaction settlement and dispute settlement process, where the paid transaction amount is set aside for a particular period during which a customer can report a merchant for not delivering the items purchased. The merchant needs to show proof of item delivery to the customer (usually customer signature collected by the merchant when the customer receives the item). If the merchant fails to do so, the paid amount is refunded to the customer. Hence the merchant is paid if the proof is confirmed to be valid or the dispute period elapse without a customer complaint. The merchant does not

need to worry about a customer not paying if the endorsers has guarantee the transaction with valid e-coins.

Our approach enables electronic commerce in a disaster area despite the restricted communication access to a bank. However, we still encounter the challenges presented in Section 2.7. We will discuss solutions for each challenge successively.

2.8.4 Preventing Collusion

In our mobile payment system, endorsers provide financial security to pay a merchant on behalf of their customers. However, since there is no direct connection to the bank during a transaction, there is a possibility that the endorsers and a customer to collude to cheat in the payment system. In addition, there is a possibility that a customer or the endorsers could draw out money from their accounts before the bank deducts money for the item. Therefore, a method is required to check the endorser account balance before the transaction is completed. We adopt the e-coin technique for the endorsers account balance confirmation.

To be able to purchase an e-coin, a certain amount of money needs to be deposited. The deposited money is locked in an endorsement account, thereby preventing the endorser from using the money to buy an item (that is, the endorser can use the money locked only for endorsement). In a situation where an endorser endorses a transaction and attempts to take away all the account balance from his/her endorsement account before the bank confirms the payment, this attempt will fail as the endorsement account is locked during the disaster mode of our payment system.

E-coin: The bank generates unique e-coins for an endorser, identical to the tokens in [21,22] e_{T_1}, e_{T_2}, e_{T_3},... e_{T_n}, for instance, the total amount of the e-coins will be equivalent to the account balance of the endorser. The e-coin contains the endorser's ID, the e-coin ID (signed with the bank digital signature), the e-coin value, and a predefined expiration date.

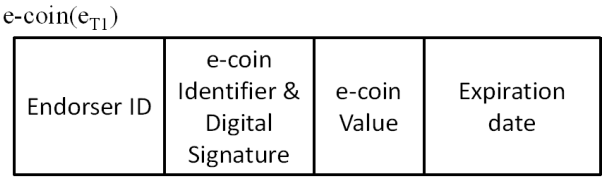

Figure 2.4: Format of an e-coin created by the bank.

The reason the expiration date is attached to an e-coin is to avoid the endorser losing money from their account if an e-coin is lost or corrupted while being delivered to the endorser. The bank sets a predetermined expiration date on the e-coin. The e-coin will be invalid after the predetermined date, if the bank has not received a report from the endorser that the e-coin was received. In the case of invalidity, the bank then issues a new e-coin as a replacement for the lost or corrupted one. If the e-coin is not utilized till it expires, the e-coin turns invalid and cannot be accepted by a merchant. A monitor can prove if an e-coin is still valid or not by confirming the expiration date on the e-coin.

To endorse a transaction, an endorser attaches to an endorsement message, an e-coin equal to the endorsement amount of the transaction. (The e-coin is part of the endorsement message, which is signed by the endorser).

In a situation where the endorsed customer pays for the transaction, the bank will reissue the e-coin to the endorser. Otherwise, the corresponding amount will be deducted from the endorser's account. Thereby, collusion between the customer and the endorser is impeded by checking if there is an e-coin attached to the endorsement message.

When an endorser requests a new e-coin from the bank, the e-coin is either received directly from the bank or transmitted to the endorser through the users available within the radio range. As a result of some communication disruption between the users and the bank, the e-coin may be lost or corrupted while being transmitted. Therefore, we adopt the use of the DTN-based data dissemination and collection via the delivery truck for delivering e-coins to endorsers in a disaster area. The bank delivers new e-coins to endorsers every two days via truck. Additionally, multihop communication can be used to deliver e-coins from the truck to endorsers. Apart from delivering e-coins to the endorsers, the e-coin truck is also used to bring back to the bank such users' messages as merchant

payment and refund of e-coins to endorsers for nondefault transactions.

Number of Colluding Parties

In our system, there might be four colluding parties: 1) customer; 2) endorser; 3) merchant; and 4) monitor. We analyze different possible types of colluding scenario formed among these parties (e.g., customer and endorsers, customer and monitor, customer and merchant, endorsers and monitor etc.) in our system.

- **Two Customers Colluding :** A customer acts as if he/she is an endorser to the other customer.
- **Two Endorsers Colluding :** Such colluding can only happen when one endorser falsely acts as a customer (e.g., has the means of forging the customer credentials) while the other endorser guarantees the transaction.
- **Two Monitors Colluding:** This is the same as endorsers colluding; colluding between two monitors can only happen one falsely disguise as a customer while the other disguise as an endorser.
- **Two Merchants Colluding :** This is conceivable only when the merchants get access to the customer and the endorser credentials (e.g., the customer and the endorser private keys, real IDs, etc.).
- **Customer, and Merchant Colluding :** The goal of this type of colluding is to defraud the endorser. In this colluding, a customer pretends to buy an item, then return the item to the merchant and share the money with the merchant. This form of colluding is difficult to detect as the endorser has genuinely agreed to endorse such transaction.
- **Customer, Endorser and Monitor Colluding :** This type of colluding is possible if the endorser is able to forge an e-coin or reused e-coins already used in previous transactions to defraud the merchant. Also, for this to work the colluding parties needs to have three unique monitor for the endorser. The merchant confirms if the e-coin is been double spent and if the event chain is not broken. Transaction is only allowed if valid e-coins and event chain are used.

- **Merchant and Monitor Colluding :** Similar to other colluding with the monitor, collusion with the monitor by a merchant is hard if the monitor of the transaction is not known beforehand. Hence, this can only happen when there is a limited number of users in the system. Our proposed mechanism dynamically assigns a monitor to check if a transaction is valid before appending their signature on the transaction.

- **Colluding with the Merchant :** Collusion between endorsers and a merchant is not possible if there is no customer. Moreover, it is not possible to forge a customer's digital signature, which is needed for every transaction.

2.8.5 Preventing Double Spending

An endorser may try to spend the same e-coin twice for two different transactions, thereby double spending the e-coin, (i.e., using e-currency twice to pay the same or different people). To prevent double spending in the system and also to ensure that the e-coin is secure, a merchant should be able to check the log for all events in the past associated with the endorser. To do this, the endorser requests other monitoring nodes to sign (with their digital signature) his/her transaction log each time a new event occurs. This will, however, require a lot of communication overhead, since the monitoring node will need to go through the endorser's entire transaction log before signing. Therefore, we propose an event chain with a light weight scheme as a solution to double spending.

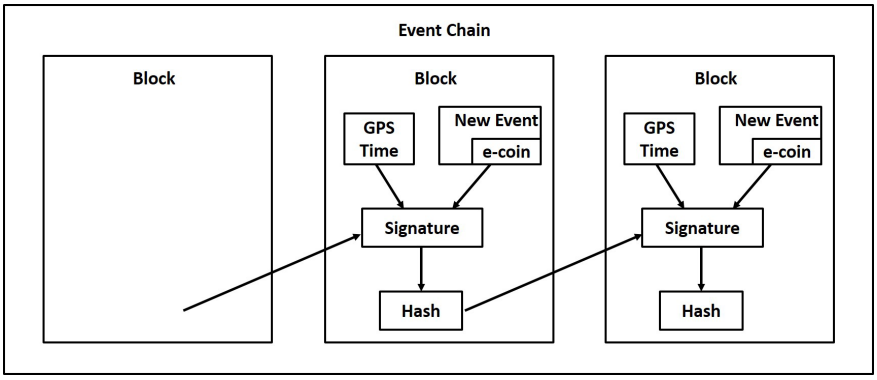

Figure 2.5: Event Chain.

Event Chain

An event chain is a successive application of a cryptographic hash function on a piece of an event log (called a block). Instead of sending and signing on the entire log, the endorser calculates the hash value of the last block, and sends it to a monitor. The monitor signs on the combination of hash value, GPS coordinates, timestamp, and a new event (e.g., spending an e-coin); the monitor then sends the block back to the endorser. In this way, all past events of the endorser are recorded to form an event chain (see Figure 2.5), which can be verified by any user. An endorser exchanges a hello message with neighboring monitor nodes periodically to add a new event to the event chain. In order to prevent colluding of up to N_c nodes, we require $3N_c + 1$ unique monitor nodes to do this operation since the maximum parties that can collude at a time is three and also, we need to prevent users that are serving as an endorser to a customer from acting as a monitor of the same transaction they are endorsing. Hence the $3N_c + 1$ unique monitor nodes will reduce the likelihood of a monitor node from being compromised as other monitors can verify the same event chain. Using less than $3N_c$ monitor nodes may result into the problem identified in [23], where the two monitor nodes may give conflicting information back to the merchant (i.e., one monitor node validates the event chain while the other invalidates the same event chain). If a predetermined length of time passes after the last event and before a new event is added to the event chain, the event chain is invalidated and can no longer be used. In order to ensure that the e-coin has not been double-spent, a user receives and checks the event log which is the entire event chain from the point at which the e-coin was issued by the bank. When a new e-coin is relayed through multihop communication to an endorser, a relay node could possibly duplicate the e-coin before sending it to the endorser. By recording all IDs of e-coins in the event chain, we can prevent the use of a duplicated e-coin.

Each user keeps the event chain as their transaction log. When a new event is created, a new block is linked to the previous event chain, as shown in Figure 2.5. The previous block and the entire log of the present transaction event are signed and forwarded to the monitor. To validate other information in a block, a user requests the entire log. It is possible a user may decide to switch off his/her phone deliberately in order to carry out a reset and recovery attack or to break

an event chain. Here the user backup his/her phone, reset the phone to default settings and restore all previous data to buy an item.

When a phone owned by an endorser is switched off, the event chain is broken as the endorser is not able to exchange hello messages with neighboring monitor nodes. Thereby preventing a new event from being added to the event chain. As a result, the endorser cannot endorse a transaction immediately after turning the phone on but since we assume that there are many endorsers available, the transaction can be guaranteed by other endorsers. The reason we use e-coins only for endorsement is to allow customers to make new transactions immediately after turning off and on the phone since the transaction is guaranteed by his/her endorsers. An endorser on the other hand, first need to exchange hello messages with neighboring monitoring nodes that will verify and update its event chain before such endorser can endorse a new transaction.

By introducing the event chain we can prevent double spending during transaction. However, due to the limited bandwidth of mobile devices in a disaster area, we need to make our mechanism significantly light weight. To achieve this, we adopt the bloom filter mechanism.

2.8.6 Light Weight Scheme

We adopt a Bloom filter [24] to represent all the spent e-coins since the beginning time of the event chain. That is to say, all spent e-coins are mapped into the Bloom filter. Instead of recording all the IDs of the spent e-coins in the event chain, only the hash value of the latest Bloom filter is recorded in the event chain. When a user checks whether a certain e-coin is double spent, the user receives and checks the Bloom filter.

In the case of a false positive of the Bloom filter, the corresponding e-coin is regarded as already spent; this coin cannot be used. In this case, users have to wait until the e-coin expires and is reissued by the bank. The Bloom filter can represent a set of a sufficiently large number of coins with a small amount of data. When 3000 coins are represented in a Bloom filter with a 1% false positive ratio, the size of the filter is 4 kb.

We also incorporate the technique called Merkle Tree [25] for reducing the size of the transaction block stored in the event chain which is to be checked by the

monitor nodes. Each transaction block is hashed and the hash values are then paired together, the resulting paired hash values are further hashed until a Merkle tree root is formed (see Figure 2.6). The Merkle tree root is stored in the event chain, thereby reducing the size of the event chain. During a transaction, only the reduced event chain and the Bloom filter need to be checked by the monitor nodes.

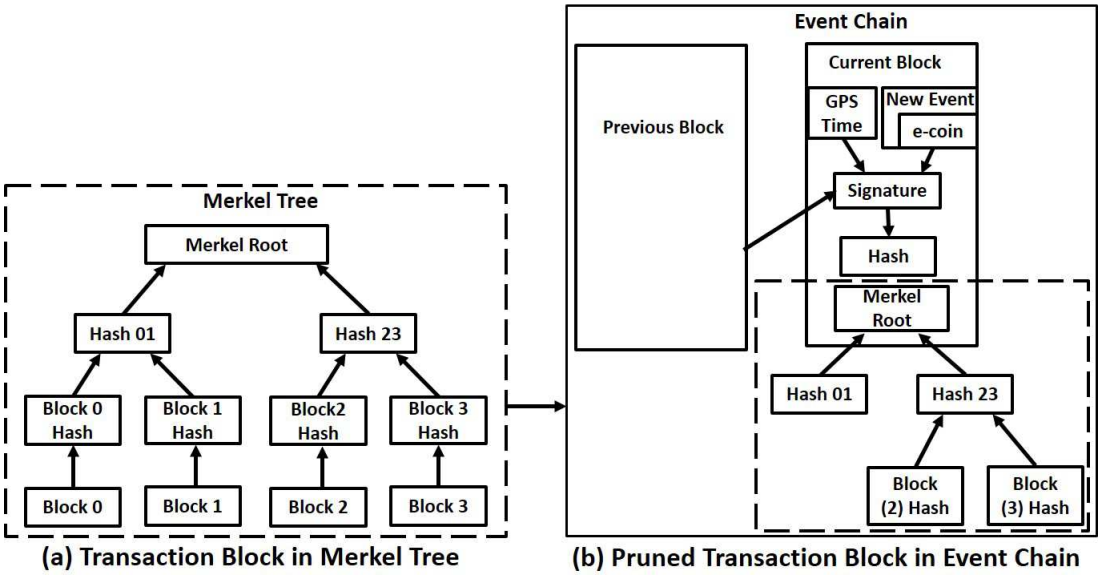

Figure 2.6: Reducing log size using Markel tree.

2.9 Other Schemes for Secure Transaction

In this section, we explain briefly various schemes adopted to secure transactions in our endorsement-based mobile payment system.

2.9.1 Location Information-Based Monitoring

Many phones might be stolen by one party to use those phones at the same time to attack the system. To prevent collusion using stolen phones, we propose a location information-based monitoring scheme to achieve confirmation of transaction location. According to this scheme, each endorser will continuously

exchange HELLO messages with monitoring nodes to prove that the endorser is in a specific location at a specific time. Other users of the system can audit the endorser's transaction location (its coordinates obtained from the GPS of the endorser's phone) by checking the endorser's log of the event chain or the log from the time when the e-coin was received. If an endorser fails to exchange HELLO messages with other users over several time intervals, this would show that the endorser is no more in close proximity of the other users or there is loss of connection. Phones that have similar location histories cannot be used as monitoring nodes.

In addition, if an attacker wants to use a stolen phone, the attacker first needs to find a way to access the customer or endorser's phone which may be protected by a biometric security. Then the attacker will need to break the 1024 encryption key, then get the bank digital signature to forge a new digitally signed picture and the customer digital signatures.

2.9.2 Blind Signature

Monitoring nodes might access another user's message before signing it during a transaction, thereby compromising the user's anonymity in the system. To prevent this and, more widely, as part of the scheme for preventing a user (customer or endorser) from carrying out multiple transactions using already endorsed transaction order message for reset and recovery attacks, we utilize the techniques of the event chain (to prevent users from reusing the same message) and techniques of the blind signature [26] (to protect anonymity).

2.9.3 Chains of Endorsers

It is possible that the number of endorsers accessible is not sufficient to pay for the transaction amount, or the customer does not have enough users to serve as his/her endorser, which would lead to a shortage of money to pay the merchant. To detect if there is a shortage of money, the e-coin attached to each endorsement message is checked, this however, would cause the merchant to reject an endorsement message every time the e-coin is less than the transaction amount. To prevent such and to ensure that the customer can purchase an item, even

when all the endorsers are not fully accessible or when the endorsement amount are not sufficient to cover the transaction amount, we introduce MLE, where each customer has multiple levels of endorsers. When an endorser is not available to endorse a transaction, an endorser of the endorser will be able to endorse such transaction.

According to this method, endorsers have their own endorsers that can inherit transactions to be endorsed. The primary endorser delegates its endorsement capabilities to the secondary endorsers; the secondary endorsers agree to serve as a secondary endorser to a customer beforehand (i.e., during registration, which before disaster happens) and only pays if the primary endorsers do not have enough money to endorse the transaction. The secondary endorsers thereby serve as a proxy to the primary endorsers and are responsible for paying the merchant in a situation where the customer fails to pay for a transaction. The merchant can access the list of the primary and secondary endorsers from the transaction message sent by the customer. The list is created beforehand (during registration) to form an endorsement-chains tree and signed with the bank signature to avoid forgery. The list is updated when primary and secondary endorsers select their own endorsers.

Let us consider a default scenario in which customer C buys an item for \$40 from merchant M, with endorser P_D as the primary endorser and endorser S_D as the secondary endorser for customer C.

Default Scenario 1 :

When a customer defaults, the primary endorser is billed by the bank. The e-coins are collected from the primary endorser P_D and S_D, however, the secondary endorser is only billed if the primary endorser does not have enough money.

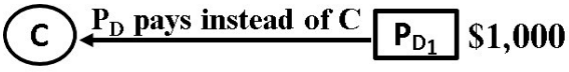

Figure 2.7: Customer default scenario using endorsement with sufficient money from primary endorsers.

Default Scenario 2 :

In a situation when a customer defaults and the primary endorsers do not have

sufficient money to cover the payment or are not available during transaction, the secondary endorsers will be charged for the transaction. Let us consider the same scenario described above, the primary endorsers (direct endorser to customer C) P_{D_1}, P_{D_2} and P_{D_3} do not have enough money. In this case, the secondary endorsers (for example, S_D) are charged. Each secondary endorser is charged according to the amount they agreed to pay for the endorsement.

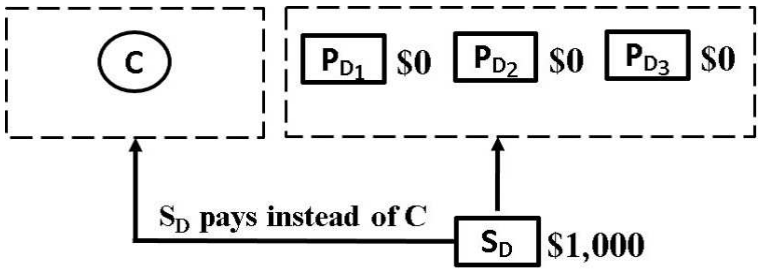

Figure 2.8: Customer default scenario using endorsement delegation with insufficient amount from primary endorsers.

The merchant sends the billing message to both the primary and the secondary endorsers to obtain their signature on the transaction as a payment guarantee. Unlike our previous method where the merchant searches for the secondary endorsers one level after the other if the primary endorsers are not available, this approach allows the merchant to send the billing message to the secondary endorsers whether the primary endorser is available or not, thereby avoiding the excessive communication needed to search for secondary endorsers when there are insufficient endorsers to endorse a transaction. This way merchant overhead is reduced. This will also ensure that there are more endorsers available to endorse a transaction. In a situation where customer C fails to pay for the item purchased, both the primary endorser and the secondary endorser will pay instead.

2.10 Relationship of Event Chain with Blockchain

In our payment system, as described in Section 2.8.5 we adopt the blockchain technology to form an event chain which is used to prevent double spending and to secure the system. This is similar to blockchain technology utilized in Bitcoin which uses a peer-to-peer network and a distributed timestamping to manage the blockchain mechanism and transactions are stored in blocks. In such blockchain, transactions are publicly verified and audited by other participants and to do this a proof-of-work is utilized. Then any transaction that is verified and validated are accepted into the blockchain where each blocks are linked by a hash pointer of a previous block. Unlike in the Bitcoin blockchain where all transactions are accepted into one blockchain which as at 2017 has grew to 100GB in size. In our system, we proposed a lightweight blockchain (event chain mechanism) which allows each user to maintain only their own chains of transaction that can be verified by other monitoring users.

In addition, there are many alternative blockchain-based system commonly referred to as Blockchain 2.0 which also adopts blockchain technology with their own modified properties and implementation of blockchain technology. The blockchain 2.0 was developed to address the challenges and limitations experienced in Bitcoin blockchain. The design of the bitcoin network limits it to handling of 3 to 7 transactions per seconds. The event chain in our system can also be termed as a Blockchain 2.0, as its implementation involves a modified properties of the original blockchain. Also, when a user buys an item with bitcoin, the transaction is broadcast to the entire network irrespective of the amount spent and the transaction is stored by all users.

Similar to Bitcoin blockchain and other blockchain 2.0, the transactions in our system are accepted into a blockchain and are publicly verified by other users. However, our approach allows each user to maintain their blockchain unlike in other blockchain where a single blockchain is adopted which resulted in scalability issues. In addition, our method does not utilized the proof-of-work scheme rather we adopt the use of the hash value of a transaction log and the monitor node's digital signature. The table below shows some of the similarities and differences

Table 2.2: Comparison between Event Chain and Blockchain

Properties	Proposed Event chain	Bitcoin Blockchain
Number of Blockchain	Multiple (users maintain their blockchain)	Single
Proof-of-work	No	Yes
Transaction verification	Public	Public
Computation	Simple	Complex
Block space	Storage Efficient	Limited
Scalable	Yes	No

between the event chain of our system and other blockchain-based systems.

2.11 Security Analysis of the Endorsement-Based Mobile Payment System

The attacks considered were selected as likely given the limitations of a disaster area plus other common MANET security challenges. Other MANET-related attacks will be considered in future work.

2.11.1 Impersonation Attack

To prevent impersonation, customer C attaches a photograph that is digitally signed by the bank before the customer encrypts the message. An attacker cannot impersonate Customer C without obtaining his/her digital signature.

2.11.2 Colluding Attack

In a situation where an endorser and a customer collude to cheat in the payment system (e.g., a dishonest endorser may endorse a dishonest customer while neither has money in their accounts). The e-coin technique is used to confirm

the endorsers' account balance during the transaction. Endorsers attach to an endorsement message, an e-coin equal to the endorsed amount of that transaction.

2.11.3 Double Spending

Suppose endorser D endorsed customer C for a transaction with merchant M_1 with an e-coin (for example, e_{T_3}) and then tried to use the same e-coin to endorse another customer's transaction with M_2. The monitoring user first checks to see if the event chain is broken or valid. If valid, then the monitoring user can hash and sign the event chain. So the event chain prevents an endorser from double spending an e-coin in our system.

2.11.4 NonRepudiation of Transaction Location Source

Suppose many phones are stolen by an attacker, collusion among those phones is possible. Also, a customer or an endorser current transaction may be carried out from a different geographical location which differs from the location of previous transactions, and then repudiate having made such a transaction. Regarding such cases, other users of the system can detect if any transaction has been carried out away from an endorser's usual location by monitoring the transaction location. The usual location is the geographic location where the user's phone has been used for a few days. The endorser's entire log of the event chain or the log since an e-coin was received is compared to the event chain at the end of the previous HELLO message exchanged by the endorser. This makes it impossible for an attacker, a customer or an endorser to carry out a transaction in a location other than the usual location.

2.11.5 Reset and Recovery Attack

Suppose a dishonest customer buys an item from a merchant, then resets the phone to the default settings. Then the customer recovers the backup data and uses the same data to buy an item from a different merchant. To reuse a message (a transaction order message or an endorsement message) or an already endorsed transaction message, the user needs to change the event chain of all

previous transactions in order to modify the hash values, GPS coordinates and the timestamp in the previous transaction. The user cannot modify the previous transaction message without changing the hash values. The merchant or the monitor will detect that the message has already been used. They do this by checking the entire event chain to see if the predefined time has passed before a new event was added to the event chain.

2.12 Results and Discussion

2.12.1 Simulation Configuration

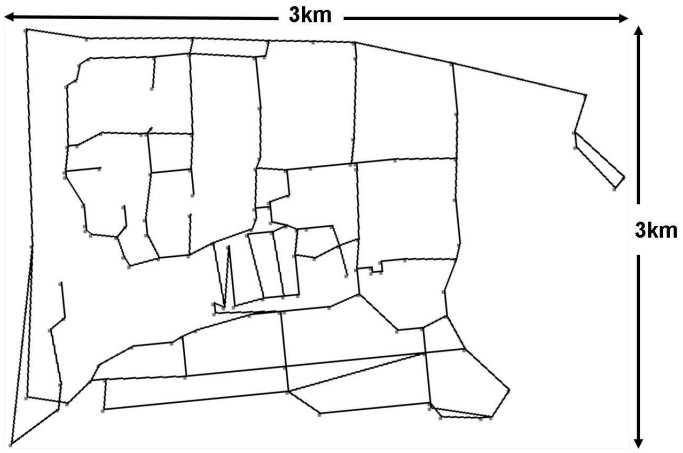

Figure 2.9: Map for Simulation.

The main objectives of our simulation are to validate: (i) usability of our proposed system in a disaster area and (ii) reduction of communication cost in order to provide excellent service for people in a disaster area.

We conducted our simulation using a customized simulator. The simulated scenario is implemented to enable nodes to connect with each other easily within the transmission range, given that mutual and location monitoring is an important mechanism in our protocol. Mobile nodes are first evenly placed in a 3km x 3km area. This is based on an actual map of the area around Nara Institute of

Science and Technology in Nara, Japan, as shown in Fig 2.9. The skeleton map represents the road network there. Each node moves according to the Random waypoint mobility model [27] at a uniform speed of 1 to 1.4m/s and a pause time of 10 seconds (Nodes serving as endorsers also move according to the same mobility). The route is based on Dijkstra's shortest path algorithm. In our simulation, transaction message broadcast time interval was set wherein during this time, the nodes move according to its mobility model and actively perform an action depending on their role at that particular time, (that is, either customer, endorser or monitor). All nodes have the same buffer size and transmission range. We assume 802.11g wireless WiFi (802.11g comes with ad-hoc mode) is used for communication. The summary of the default values used in our simulation is shown in Table 2.3. The network bandwidth of our simulation is set to 1 Mbps, and our message size is set to 5KB.

Table 2.3: Typical simulation parameters value in a disaster area

Parameter	**Value**
Propagation Model	Unit-disc model
Bandwidth	1 Mbps
Buffer Size	100-500KB
Transmission range	100m
Disaster area map size	3km x 3km
Number of mobile nodes	100–500
Node speed	1 - 1.4m/s
Node pause time	10s
Mobility Model	Random Waypoint
Message size	5 KB
Hello Message Size	5 bytes
Hello message Interval	10s
Bloom filter size	256bits
Proportion of endorser to customer	4%
Number of monitoring nodes	3
Transaction amount ($)	2
Endorsement amount ($)	2
Total e-coin per endorser ($)	3000

The following metrics will be measured in our simulation.

2.12.2 Transaction Completion Ratio

- **Transaction Completion Ratio (TCR)**: The transaction completion ratio is defined as follows:

$$TCR = \frac{\text{No. of successful transactions}}{\text{No. of transaction messages Rec'd by merchant}}$$

We evaluated the transaction completion ratio to determine the usability of our system in a disaster area. Specifically, we considered two scenarios, the first being the single-level endorsement (SLE), where transactions are endorsed by primary

endorsers only. The second scenario considered is the MLE where transactions are endorsed by primary and secondary endorsers. All simulated results in the figures below are the averages from 20 simulation runs (see [28]).

2.12.3 Transaction Completion Ratio of Single-level Endorsement

As shown in Figure 2.10, the SLE achieved an average of 42%, 51% and 43% of transaction completion ratios for 100, 200 and 500 nodes, respectively. The transaction completion ratio increases as time increases at the early stage of the simulation and decreases as the simulation reach a steady stage. This is due to limited number of endorsers for guaranteeing customers transactions.

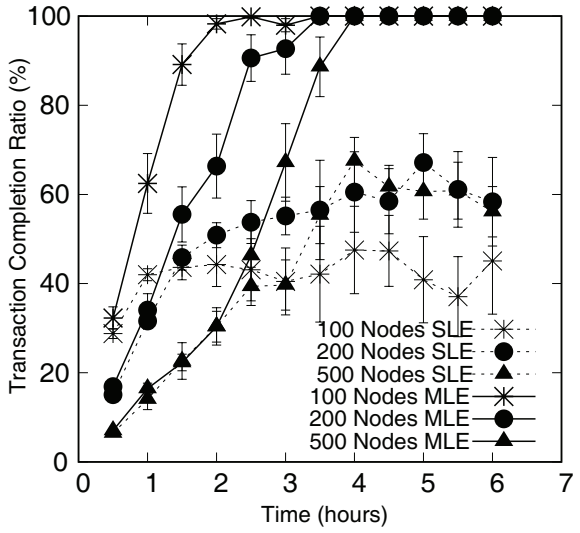

Figure 2.10: Transaction completion ratio. (SLE : Single-Level Endorsement, MLE : Multilevel Endorsement, endorser ratio = 4%, merchant No.= 1 and monitoring node No. = 3).

2.12.4 Transaction Completion Ratio of Multilevel Endorsement

The transaction completion ratio increases significantly with the MLE mechanism, averaging 90%, 80% and 65%, respectively, for the three cases above. The

significant increase is due to having more endorsers for guaranteeing customers' transactions. Although the transaction completion ratio decreases as the number of mobile nodes increases, the proposed MLE achieves better performance when compared with the SLE, showing an increase from 22% to 48%. The significant increase is as a result of having more endorsers to guarantee customer transactions. We achieved this improved performance with the introduction of the MLE. We can also observe that the transaction completion ratio increases as time increases, this is because simulations are in a transient stage from 0.5 hours to 3.5 hours, and beyond 3.5 hours simulations reach a steady stage.

2.12.5 Communication Overhead

- **Merchant message size**: The size of the message needed by the merchant to check the validity of an event chain and to contact secondary endorsers in a successful transaction.

Our goal in introducing the MLE is to increase the transaction completion ratio in our system. However, the merchant should not incur additional communication overhead when MLE is used. Therefore, we evaluated the merchant communication overhead of our previous event chain as against the merchant overhead of our proposed MLE. As shown in Figure 2.11, when compared to the merchant message size in our previous MLE where merchant searches for secondary endorsers, one level after the other, there is a 49%, 52%, 60% decrease in merchant message in our mobile payment system with our newly proposed MLE for different scenarios with 100, 200 and 500 mobile nodes respectively. In all scenarios, the simulation results show that the overall merchant message size of our system is 7MB on average, with an average of 54% decrease of that of our previous event chain, indicating that our system with MLE is storage-efficient for mobile devices, which have limited resources in disaster areas.

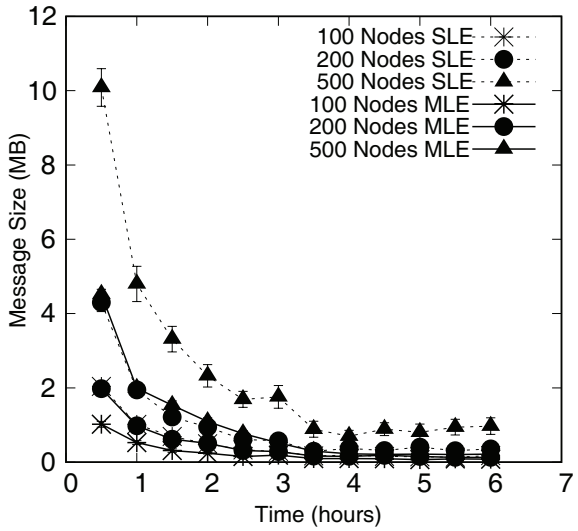

Figure 2.11: Merchant message size. (SLE : Single-Level Endorsement, MLE : Multilevel Endorsement, endorser ratio = 4%, merchant No.= 1 and monitoring node No. = 3).

2.12.6 Event Chain Validity

- **Validity Ratio of Event Chain (VR):** The ratio at which the event chain is valid in our system, which is computed with the following formula:

$$VR = 1 - \frac{\text{No. of rejected transaction}}{\text{No. of rejected endorsement messages}}$$

Another metric we measured is the validity ratio of event chain. In our mechanism, we introduced event chains to prevent double spending. However, an event chain may be invalidated if dishonest users in the network double-spend e-coins, complete a transaction without e-coins or try to complete a transaction without a monitoring node's signature; or, if too many nodes share a similar location history. The simulation results of the validity ratio of an event chain are shown in Figure 2.12. The results indicate that the validity of an event chain in our system is very high for different scenarios with 100, 200 and 500 mobile nodes with an average validity of 98%, 99% and 99%, respectively, when the proposed MLE mechanism is used. The validity ratio increases as the number of mobile nodes

increases, which is a result of having more endorsers with valid event chains and having sufficient monitoring nodes available to monitor transactions. The slight decrease observed in event chain validity from 0.5 hours to 3 hours is due to an insufficient number of monitoring nodes when the simulation is in a transient stage.

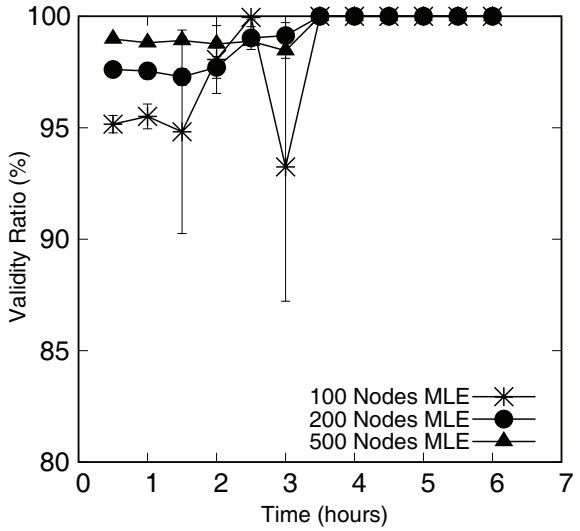

Figure 2.12: Event chain validity. (SLE : Single-Level Endorsement, MLE : Multilevel Endorsement, endorser ratio = 4%, merchant No.= 1 and monitoring node No. = 3).

2.12.7 Transaction completion time

- **Transaction completion time:** The time interval from the time a customer initiates a transaction to the time the merchant accepts the transaction and supplies the items.

We also evaluated the transaction completion time of our system in our simulation. First, we explain each process and analyze the simulation time. The customer creates a transaction message, appends its digitally signed picture and its signature to the message. The average computation time for creating the transaction message and generating a signature by a customer is 0.006s. The merchant first verifies the customer information and signature, then verify the

digitally signed picture. If the event chain is valid, the merchant generates the billing message and forward it to the endorser, the computational time for this is 0.07s. Similarly the endorser verifies the merchant signature, generate the endorsement message and creates a new block for the event chain with an average computational time of 0.5s. The endorsement message is forwarded to other users for monitoring, a monitoring node validates the event chain and append its signature. The average computational time for monitoring an endorsement message is 0.1s. Finally, the validated endorsement message is forwarded to the merchant and the merchant also validate the endorsement message to avoid collusion between a monitoring node and an endorser, this takes an average of 0.03s. The advantage of our proposed system is that the average transaction completion time is 1.2s, which is the reason for the faster execution of transactions in our endorsement-based mobile payment system.

2.12.8 Event chain size

- **Event Chain Size:** The event chain size with light weight mechanism as against the normal event chain size.

We also evaluate by calculation the size of an event chain scheme when the light-weight mechanism is used. First, we analyze each component that form a block such as new event, timestamp, GPS coordinate, signature and hash value and calculate each components size to get the total size of a block. Each block contains information of 10 e-coins while an event chain using 30 blocks stores information of up to 300 e-coins. Then we calculate the size of event chain. The result shows that the size of the event chain decreased from 3.6KB to 1.7KB with the light weight mechanism, which shows that our light weight mechanism brought a 54% reduction in event chain size. Similarly, the size of the event chain checked by a monitor decreased from 0.24KB to 0.14KB, with 41% reduction when the light weight mechanism is applied.

2.12.9 Effect of Various Parameters on Transaction Completion Ratio

To clarify if our system can achieve better performance than our newly introduced MLE mechanism, we examined other scenarios in our simulation by varying different parameters (endorser density, mobility speed of nodes, and density of monitoring nodes) to check how these parameters impact the performance of our system when the SLE is used.

2.12.10 Endorser Density

Figure 2.13 shows that the endorser's density has an impact on the transaction completion ratio. First, we varied the proportion of endorsers from 2% to 12%. The transaction completion ratio increases as the number of endorsers increases, confirming the effectiveness of our MLE mechanism. We also observe that there is a slight decrease in the transaction completion ratio for 500 nodes. This decrease is as a result of an insufficient number of monitoring nodes in spite of there being more endorsers in the system, e.g., 40 endorsers, giving an endorser proportion of 8%.

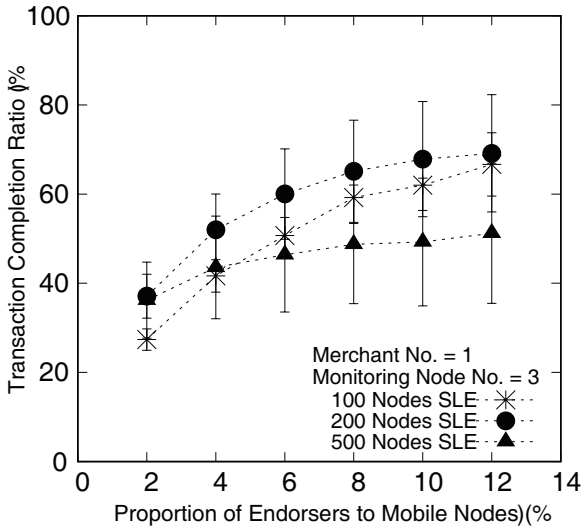

Figure 2.13: Effect of endorser density on transaction completion ratio (SLE : Single-Level Endorsement).

2.12.11 Mobility Speed of Nodes

Since the contact times of nodes are essential for a successful transaction, we evaluate the impact of a node's mobility speed on the transaction completion ratio. The result is shown in Figure 2.14 with almost constant transaction completion ratios. According to this result, a node's mobility speed has no significant effect on the transaction completion ratio while the mobility speed increases.

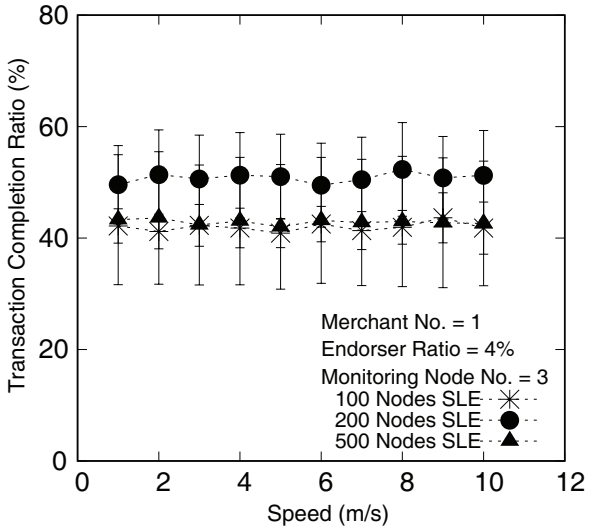

Figure 2.14: Effect of mobility speed of nodes on transaction completion ratio (SLE : Single-Level Endorsement).

2.12.12 Density of Monitoring Nodes

As shown in Figure 2.15, the transaction completion ratio decreases when the number of monitoring nodes needed to complete a transaction successfully increases. The highest transaction completion ratio achieved is found when the monitoring node proportion is set to 4%. This confirmed the effectiveness of our proposed system setting, i.e., 3 monitoring nodes for validating each message to avoid collusion.

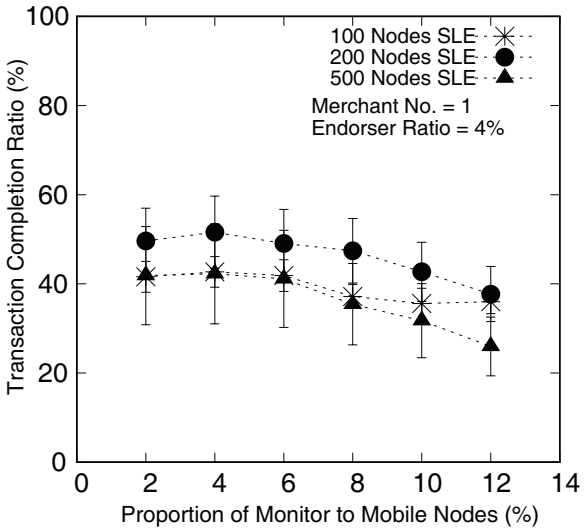

Figure 2.15: Effect of density of monitoring nodes on transaction completion ratio (SLE : Single-Level Endorsement).

2.13 Conclusion

In this paper, we proposed a new mobile payment system which utilizes infrastructureless MANETs to enable users to buy recovery goods in a disaster area. According to the endorsement mechanism, endorsers provide absolute payment security for every transaction between a customer and a merchant, therefore permitting mobile transactions in disaster areas even without direct access to the bank. Moreover, by adopting various schemes like the Bloom filter, the blind signature, the event chain, plus location information-based monitoring, the proposed mobile payment system is capable of providing secure transactions, while preventing a fraudulent transaction, collusion, reset and recovery attacks, impersonation of users, double spending. The system also reduces merchant overhead and transaction completion time.

Simulations confirmed that our endorsement based mobile payment system is useful in disaster areas. Specifically, we evaluated the transaction completion ratio, the merchant communication overhead, the validity ratio of event chain, transaction completion time, and the event-chain size of our system. The MLE mechanism in our mobile payment system achieved a better transaction comple-

tion ratio, showing an increase of 22% to 48% when compared with SLE. Also, the results show that our system is storage-efficient for mobile devices with limited resources in disaster areas, with an overall average merchant message size of 7MB for all network scenarios tested, which is an average decrease of 54% compared to our previous mobile payment system. Also, the validity of the event chain mechanism is significantly higher, with an average of 98% - 99% for all network scenarios.

3 Monitoring-Based Method for Securing Link State Routing against Byzantine Attacks in Wireless Networks

3.1 Background

The need to ensure routing security in wired and wireless networks has kept growing over the years with the development of various secure routing protocols. Secure routing allows easy packet transmission between nodes without any fear of compromise. However, routing protocols are still vulnerable to security challenges. One of the foremost challenges to routing protocol is Byzantine attacks. In such attacks, a node can interrupt route discovery, impersonate a destination node, corrupt routing information, completely drop packets, or inject fake packets into the network. These attacks prevent timely delivery of packets from the source to the destination. These types of attacks can be carried out by a malicious node either outside the network or within the network. Even though these types of attacks can be easily detected in a wired network, ad-hoc networks are still very vulnerable to such threats.

Most work already carried out on route security has adopted one of three main approaches: the cryptography-based approach, the trust-based approach, or the incentive-based approach [29]. In the cryptography-based approach, various cryptography mechanisms such as private and public key encryption schemes, digital signatures, hash functions, and/or end-to-end authentication are adapted to secure the packet in the routing protocols. The major drawbacks of this approach

are the high computation overhead, and the difficulties of maintaining secure key management and session management. In the trust-based approach, nodes participating in the routing of packets are assumed to be trustworthy as the assigned trust value is used to determine each node's reputation, so the security mechanism provided focuses more on the information being exchanged among nodes. In this approach, lost packets are often attributed to poor link quality which may not be the case when malicious nodes may drop packets. In the incentive-based approach, nodes participating in routing are given some form of incentive to report malicious nodes. However, this approach is often combined with other approach such as a trust-based approach to be successful. In addition, tamper resistant hardware is added in this approach as a security measure and this is not generally applicable in all scenarios.

In this chapter, we proposed a monitoring approach to secure the link state routing protocol against Byzantine attacks and give detailed explanations about the implementation of our monitoring-based method, then show the results as validated by simulation. Our monitoring-based method secures the link state routing protocol against Byzantine attacks except Denial of Services (DoS) attacks. Here, a DoS attack is an attack where one or more malicious nodes transmit an overwhelming number of packets or jamming signal to clog some links. The goal of our proposed scheme is to guarantee communication among connected benign nodes in the network. Our approach focuses on using the link state routing protocol to analyze and record the actions of each node within the network. Specifically, each node monitors the actions of neighboring nodes and compare the optimal packet route against the route history. This allows monitoring nodes in the network to track the past events of packets sent. Our monitoring scheme adopts three main methods:

1. Hello message verification - where a node collects hello messages and digital signatures of neighboring nodes, which are used to verify the validity of hello messages and to identify inconsistent information when a malicious node tries to corrupt routing table information.

2. Packet history field monitoring - here the source node calculates the optimal path and stores it in each packet (like Dynamic Source Routing), and then

neighboring nodes check whether packets are forwarded correctly according to the stored optimal path. Also, the event history is recorded in each packet at each intermediate node.

3. Statistical hypothesis testing - while some packets may be dropped due to poor link quality, we need to know if a node is intentionally dropping packets. To determine this, we adopt a statistical measure in which monitoring nodes observe the packet-dropping behavior of other nodes, and then calculate the probability (*P-value*) of an intermediate node dropping a packet.

To detect malicious nodes, the *P-value* is compared to a significance level value (reflecting the number of dropped packets that can be tolerated), while packet history field monitoring is used to identify at which node a malicious action is carried out.

3.2 Overview of Byzantine Attacks

In this section, we describe Byzantine attacks.

3.2.1 Byzantine Attacks

Byzantine attacks can be described as attacks in which malicious nodes take control of one or more network nodes and disrupt the network functions [29]. Malicious nodes can selectively drop packets, corrupt routing information, or send packets on non-optimal paths. When carried out by a fully authenticated node in the network, these types of attacks are difficult to detect. Some of the Byzantine attacks are described below.

3.2.2 Corruption of Routing Table Attacks

In these attacks, the goal of a malicious node is to corrupt the routing table, either by falsifying neighbor information, or by capturing and modifying the neighbors' link information broadcast by a benign node. Doing this can cause the routing protocols to maintain the wrong information in the routing tables, which now

include the malicious nodes in almost all routes to destinations. Figure 3.1 shows an example of corruption of routing table attack.

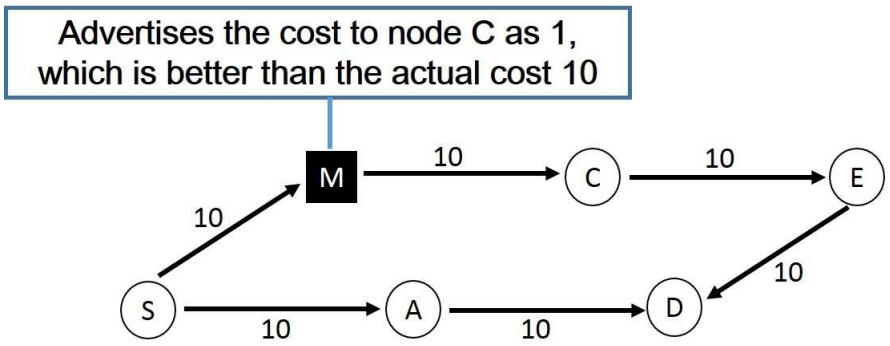

Figure 3.1: Corruption of routing table attack.

3.2.3 Falsifying Location Information Attacks

In this attack, a malicious node forges a position in the network which is completely different from its actual position, and reports the forged position to other nodes in the network. This causes benign nodes in the network to calculate the wrong status and cost of the malicious node's links, which leads to invalid information in the routing table and packet loss. Figure 3.2 shows an example of a falsifying location information attack.

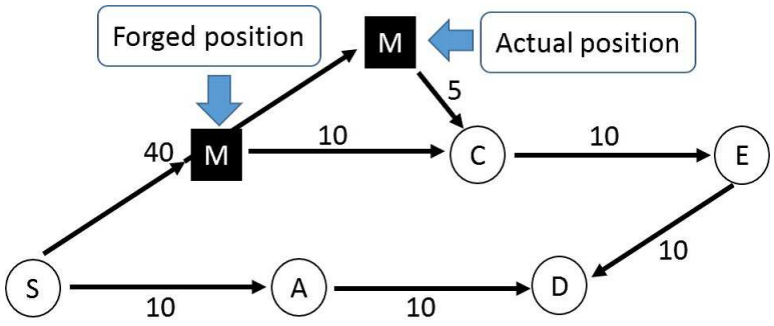

Figure 3.2: Falsifying location information attack.

3.2.4 Black Hole Attacks

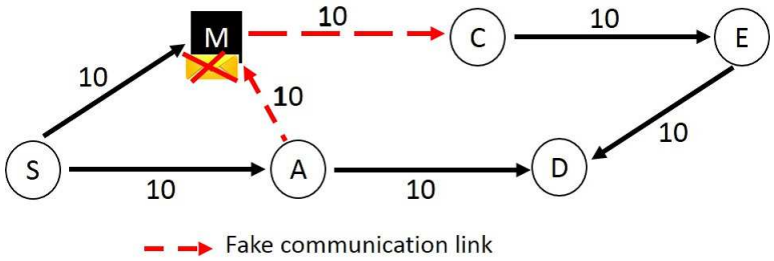

Figure 3.3: Black hole attack.

In this form of attack, a malicious node injects fake routing information to attract all packets to itself, and then either drops all of the packets, modifies some packets, or selectively drops packets. To avoid detection, such malicious nodes sometimes actively participate in routing packets to the destination in a normal way. This makes it difficult for other nodes in the network to detect such malicious node action. Figure 3.3 shows an example of a black hole attack.

3.2.5 Sink Hole Attacks

Similar to the black hole attack is a sink hole attack, in this attack a malicious node attracts all packets to itself by claiming to have shortest path to all destinations in the network. Other intermediate nodes then relay their packets through the malicious node. The malicious node can then either modify, fabricate, or eavesdrop on the packets.

3.2.6 Wormhole Attacks

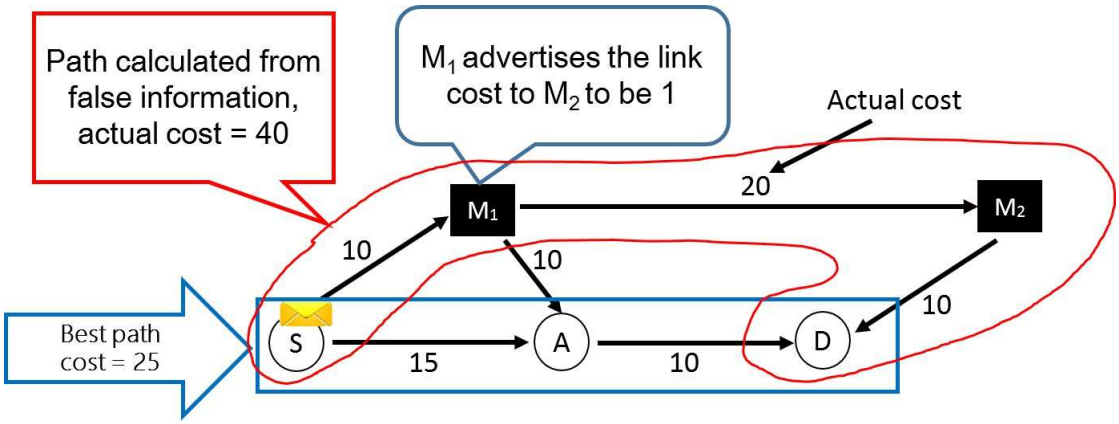

Figure 3.4: Wormhole attack.

In this form of attack, a malicious node advertises an artificial route as the best path to the destination node, and tunnels the packets to another malicious node, thereby causing the source node to ignore the genuine route. Such malicious nodes can either drop all packets, or selectively drop packets, preventing timely delivery of packets and causing packet loss in the network. This is also a form of colluding attack. Figure 3.4 illustrates an example of a wormhole attack.

3.2.7 Colluding Attacks

In a colluding attack [40], a group of nodes collaborates to carry out an attack by dropping or modifying packets. One of the nodes will advertise itself as having the shortest path to the destination. The shortest path may or may not include other collaborating nodes to complete the attack. This form of attack is hard to detect, especially when the nodes align with each other as neighbors. For example, Figure **??** illustrates two example scenarios for colluding attacks. As shown in Figure 3.5, the first colluding scenario is when a malicious node M_1 is part of the selected packet route, but decides to forward the packets to another colluding malicious node M_2 on a non-optimal path.

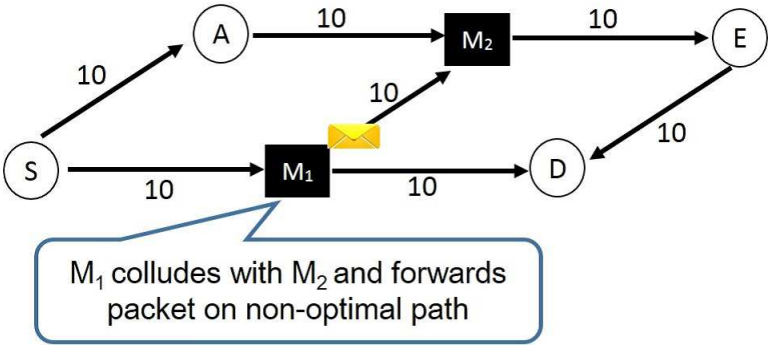

Figure 3.5: Colluding by forwarding packets on non-optimal path.

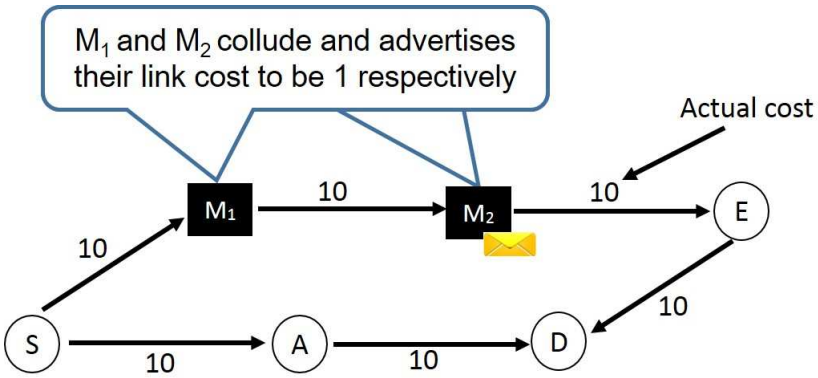

Figure 3.6: Colluding by delaying packets.

The second colluding scenario, shown in Figure 3.6, is when the source node S decides to send a packet to destination D. The best shortest path is S - A - D. However, since node M_1 is malicious and colluding with another malicious node M_2, both malicious nodes advertise the wrong link costs, e.g. 1 and 2 respectively, so the best route appears to be S - M_1 - M_2 - E - D. Then the malicious nodes M_1 and M_2 forward the packet from the source node S at the actual link cost, which causes packet delays to node E. The colluding malicious nodes can also drop packets.

These types of Byzantine attacks are difficult to detect or prevent, especially when carried out by an insider attacker. Therefore, as described in the next section, we adopt a monitoring scheme to secure routing in the LSR protocol.

3.3 Proposed Secure Routing Protocol with a Monitoring Scheme

In this section, we first describe the link state routing protocols and its features. Then we explain our secure routing protocol and monitoring scheme designed to protect a network against Byzantine attacks. We explain how a valid routing table is formed, then we describe the statistical method used to detect malicious nodes and the monitoring scheme used to secure the LSR protocols. Finally, we explain how our proposed scheme mitigates Byzantine attacks.

3.3.1 Link State Routing Protocols (LSR)

Link state routing (LSR) protocols [41] are proactive protocols in which a node exchanges Hello messages with other surrounding nodes to know the entire network information. Based on the information acquired from other nodes, the node first creates a topology of the network and positions itself at the root of the spanning tree, then uses a Shortest Path First algorithm, such as Dijkstra's Algorithm, to find the best path to a destination. Examples of LSR protocols are open shortest path first (OSPF) and intermediate system to intermediate system (IS-IS).

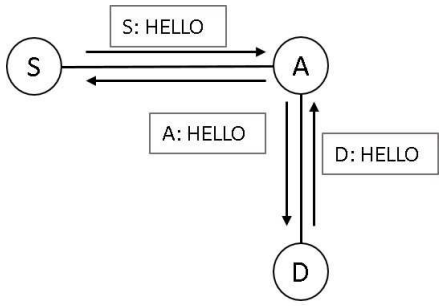

Figure 3.7: Exchange of Hello messages by nodes

Each node needs to discover neighboring nodes. In order to do this, a node sends a Hello message periodically (e.g. every 10 seconds). The Hello message contains the node's unique ID. By receiving Hello messages from other nodes, a node can determine which nodes it is directly connected to. The Hello messages are only sent to directly connected neighbors, for example, as shown in Figure

3.7. Hello message from node S is only sent to node A, while Hello message from node A is sent to both nodes (node S and node D) which are its direct neighbors. Also, Hello messages are used to detect link failures and node availability in the network. A link failure is detected if a Hello message is not received from a particular neighbor within a time interval (e.g. 30 seconds). The time interval in which a node is not able to send/receive a Hello message to/from its neighbor is also called a dead interval.

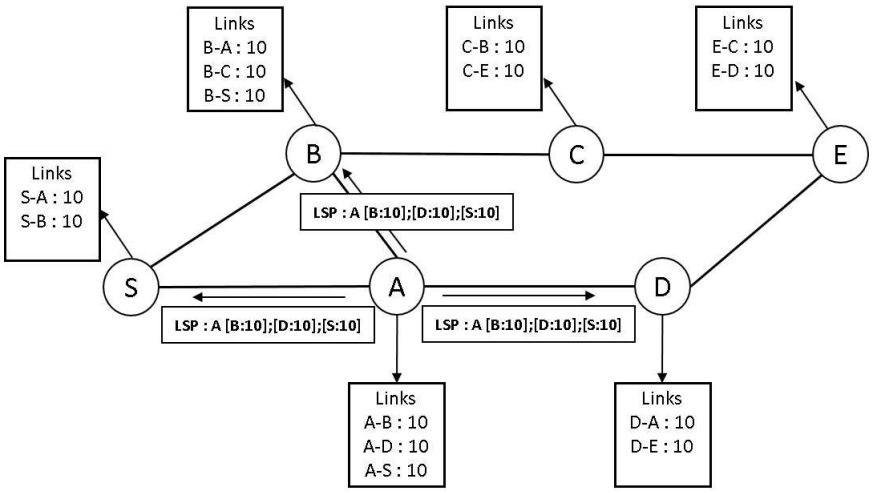

Figure 3.8: An example of link-state packet flooding

After a node has learned all the information about the network topology, the node can now distribute its local link information (i.e. its view of the network) to other nodes in the network. This will assist other nodes in forming their own view of the whole network. To distribute its local view, a node needs to build a link-state packet (LSP) which contains: node's ID, LSP age, LSP sequence number, LSP links (i.e. links advertised by another neighbor node which include the neighbor's ID and link cost). A flooding algorithm is adopted for the distribution of the LSP to all nodes in the network. Also, each node maintains a link-state database (LSDB) to store the recent LSP received from other nodes. Once a new LSP is received, each node verifies it against the one that is stored in the LSDB. If the LSP did not match any of the LSP stored in the LSDB, then the node will forward it to all nodes except for the node it receives it from. Figure 3.8 highlights an example of LSP flooding.

In addition, acknowledgments and retransmission are used to ensure that all LSPs are received by all nodes in a situation where there is a link failure. It is possible that a link failure is detected by one of the nodes directly connected to each other. In that case a new LSP is generated and the failed link is removed from the LSP. The new LSP is then flooded to all nodes in the network. Each node can then replace the previous LSP with the new one. After receiving all LSPs in the network, each node can compute the complete network topology. Then, using Dijkstra shortest path algorithm, each node computes a spanning tree and positions itself as the root of the tree. The routing table is automatically formed from the spanning tree. Packet from a particular node in the network is routed to the specified destination based on its routing table. Each node consumes energy while broadcasting and receiving Hello messages from neighbor nodes. Also, as the number of transmitting packet increases, the power consumption of a node will also increase.

3.3.2 Preliminaries

Any node can join the network without pre-registration. We adopt the use of symmetric and asymmetric keys for the encryption/decryption of packets. The asymmetric key scheme is used to generate/verify a digital signature, while a session key, generated using symmetric key algorithms, is used to encrypt/decrypt the data of a packet. A unique key pair can be safely created from random numbers by any node. The public and private keys are unique to each benign node and the private key is kept secret by each node. Benign nodes create and exchange public keys beforehand. In addition, nodes authenticate each other with a digital signature. A node will sign its signature on the ID which can be verified by other nodes. Our method adopts the digital signature algorithm (DSA) described in [42].

To generate a digital signature, a node applies a one-way secure hash function to a message (e.g. ID or packet), then encrypts the hash value with its private key to form a signature (i.e., the encrypted hashed ID is the signature of the node). The calculation of encrypting and decrypting the hash value of a message by a public key cipher is faster than the calculation of encrypting and decrypting the message directly by a public key cipher. Other nodes can verify a node signature

by first decrypting the signature using the signing node's public key to reveal the hash values (i.e., hashed message), then applying the same one-way secure hash function to the signing node's original message to generate a new hash value. Finally, the verifying node compares the two hash values to validate if it matches. The signature is valid if the two hash values are the same.

Moreover, to encrypt/decrypt a packet, a source node creates a session key, then encrypts the data of the packet with the session key. The session key is encrypted with the destination node's public key using the asymmetric key algorithm such as RSA. The encrypted packet and the encrypted session key are sent through the selected route to the destination node. After receiving the packet, the destination node decrypts the encrypted session key with its private key, then decrypts the data with the session key to reveal the data sent from the source node.

3.3.3 Assumptions

In this section, we make the following assumptions.

- All benign nodes are connected in the network topology. i.e., there is always a route only consisting of benign nodes between any pair of benign nodes in the network.
- Each node in the network maintains low mobility.
- All nodes can generate pairs of public and private keys.
- A key pair is kept secret by a benign node.
- A benign node only generates one pair of public/private keys.
- Links are not stable, i.e., not all packets are received by neighboring nodes.
- All benign nodes know the link states of all neighboring nodes.
- Due to wireless channel fading during transmission between two nodes, a packet may be dropped with probability q. We assume a benign node can estimate the probability q of packet dropping between itself and a neighbor node.

- All packets are forwarded in First-In-First-Out order.
- There is time synchronization between benign nodes.
- Each node knows the upper limit of the time synchronization error.

3.3.4 Routing Table Formation

A malicious node might possibly corrupt routing table information by sending inconsistent information to other nodes in the network. To prevent this and ensure that each node can verify the validity of a Hello message, in our proposed method we make a slight modification to the Hello message of standard LSR protocol by introducing security parameters in the Hello message as shown in Fig. 4.

In addition to the information in the standard LSR hello message, the hello message of our protocol includes the node's ID, digital signature, number of packets dropped, number of packets sent, number of packets received, number of packets forwarded, a timestamp, and a list of neighbors. Also, each node appends to their own hello message information in the collected hello messages from their neighbors which includes the neighbor's ID, neighbor's link cost, timestamp, and the neighbor's digital signature. Figure 3.9 shows an example of typical information in the hello message of an LSR protocol, which is 48 bytes when a node is connected to one neighbor (with a 24-bytes header and a 24-bytes hello message) and the information in our protocol which we specifically introduced to achieve routing security with additional information of 272 bytes when RSA signature is adopted. The size of the hello message of a node varies depending on the number of neighbors.

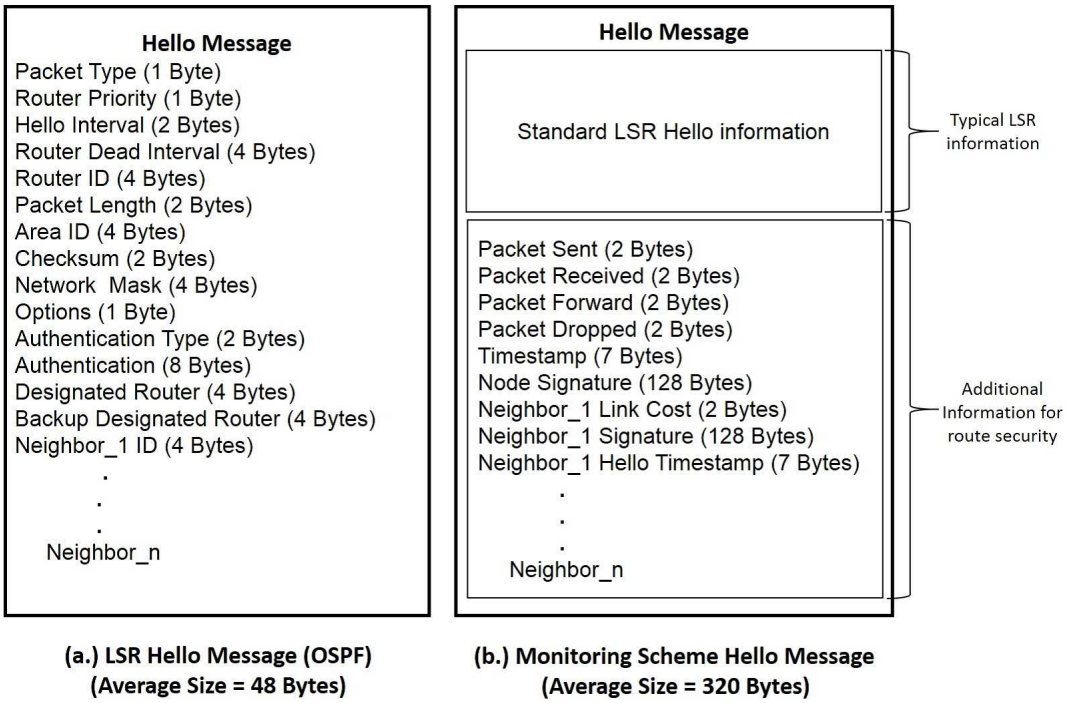

Figure 3.9: An example of hello message

Each node floods the link state information of its neighbor to other nodes in the network. As part of this flooding, we use acknowledgment and retransmission because links are not reliable. Each node maintains its routing table using the neighbor information in the hello message. Since benign nodes are connected, all neighbor information of benign nodes reaches all benign nodes. For each link, the quality of that link is reported twice from two nodes. If the information from two nodes is different, we adopt the worse one. After a node collects all topology information, each benign node calculates the best logical path to every possible destination with the information collected from the hello messages. Then it uses the best paths to each destination to form its routing table.

After neighbor nodes receive a hello message, each neighbor node responds to the hello message by sending an acknowledgment to confirm receiving the hello message. Within the replies, each neighbor node identifies itself with its node ID and digital signature. The node that initiates a hello message can use the information from the neighbors to confirm that the hello messages were received. Also, when a node receives a new hello message from its neighbor, after authenti-

cating the neighboring node with its signature and node's ID, the node will then check the timestamp to confirm that an old hello message has not been replayed.

3.3.5 Monitoring Scheme for LSR Protocol

In a LSR protocol, to send a packet from a source to a destination, the routing protocol finds the shortest path to the destination using the information in the source node's routing table. However, a malicious node that is included in the route to the destination may attack the route. To prevent such attacks, we introduce a statistical method and a mutual monitoring scheme.

Let's consider a situation such as that in Figure 3.10, in which node S is sending a packet to destination D with S - A - D as the shortest path to the destination.

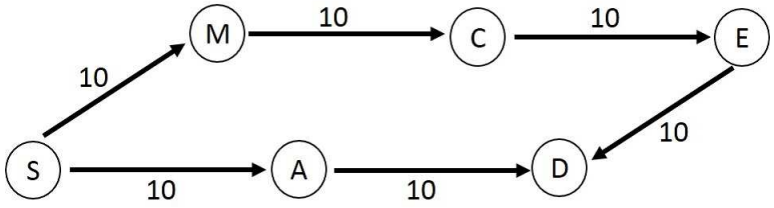

Figure 3.10: Packet route in a network

In our method, we only ensure communication among benign nodes. The first thing we should avoid is forwarding packets along a wrong route, or dropped by malicious nodes. In order to prevent this, first, surrounding nodes compare the optimal packet route against the route history. The optimal path is calculated at the source node and stored in each packet in our protocol. Previous nodes and other neighbor nodes in the network overhear when the packet is forwarded by each node. Then, the nodes checks if the packet is forwarded on a wrong route or dropped. The monitoring node checks the packet history to verify if it is correctly signed by the forwarding node, the packet signing and verification uses the same process explained in 3.3.2.

If the node fails to correctly sign the packet, the results of the monitoring are reported to other nodes in the network. Recording the packet route history of packets allows other nodes to track the past events of packets sent. In order to confirm that the packet is delivered to the destination, the destination node sends

an acknowledgement packet through the reverse route. If the source node does not receive this acknowledgement packet after a certain amount of time elapses, the source node asks the nodes along the route to show the signature from the next node.

In addition, it is possible that some packets may be dropped due to poor link quality. A malicious node may also drop the packet, and state poor link quality as the reason for the packet loss, as a result of this we need to know if a node is intentionally dropping packets. Therefore, we use a statistical method explained in the next section to determine if a node is intentionally dropping packets.

When some node reports another node to be malicious, we need to handle the cases where a malicious node is reporting a benign node to be malicious. Our goal, that is to maintain communication among benign nodes, can be achieved by separating malicious nodes from benign nodes. When some node reports one of its neighboring node to be malicious, we can be sure that at least one of them is malicious. Thus, we separate those two. In our protocol, the link between two nodes is advertised to the whole network, and it will not be used in the future.

In a situation when a malicious node decides to keep rejoining the network with a new address after being excluded from the network, then such malicious node is not immediately included in the routing of packets. We wait for some time after a new node joins the network, during this period this node is not used as a part of a route. In addition, a malicious node might intentionally delay packets, expecting that the packet delay would be hidden by delays due to transmission conflicts with other nodes. Also, if one of the neighboring nodes is communicating with other nodes, that node cannot start sending out packets. This cannot be observed by other nodes because of the hidden/exposed terminal problem. We introduce a signed Request to Send/Clear to Send (RTS/CTS) mechanism (explained in 3.3.8) to detect if a node intentionally delays packet forwarding and to solve the hidden/exposed terminal problem in this case. Before sending a data packet, each node first sends an RTS packet to the next hop node and only transmits the data packet after a CTS packet has been received from the next hop node. Other nodes, overhearing the RTS/CTS, refrain from sending any packets to the node until an acknowledgment packet is overheard. Then a node that is suspected of intentionally delaying packets can show the RTS/CTS packets as a proof that

there is no packet delay.

Since our protocol allows any node to create a pair of keys, a malicious node can pretend there are many nodes around it. Even some of the links are advertised to be invalid, there are still many links usable for malicious node. In order to handle cases like this, a node retransmits its packet using a 2-hop reactive mode. Using this reactive mode, a node will create a new packet history field indicating that the packet is being retransmitted with a reactive mode scheme and broadcast its packet to 2-hop neighbors. On receiving the packet, any node that is neighbor to both the source and the 2-hop destination node can forward the packet to the destination node. If a malicious node is trying not to forward the packet by pretending there are many nodes around it, all these links can be invalidated at the same time. The 2-hop reactive mode is only used when a packet has been dropped and the malicious node has been reported to other benign nodes in the network.

3.3.6 Monitoring packet dropping

A monitoring node observes the packet dropping behavior of a monitored node and adopts the approach of statistical hypothesis testing to determine if the monitored node is a malicious node.

The statistical hypothesis testing approach: First, the monitoring node makes a hypothesis H_0 that the node being monitored is a benign node and sets the value of significance level α (as a common practice $\alpha = 5\%$). Second, the monitoring node observes the monitored node for N packets and counts the number n_d of packets dropped by the monitored node. Third, the monitoring node calculates the *P-value* p using the following formula

$$p = \sum_{i=n_d}^{N} \binom{N}{i} q^i \Big(1-q\Big)^{N-i}. \tag{3.1}$$

where p is the right-tailed *P-value*, and n_d denotes the number of packets dropped among N packets where N is the number of packets being monitored at each link. Since some packets may be dropped due to poor network connection between nodes, we need to identify packets that are dropped in this manner as against packets that are dropped intentionally by malicious nodes. Therefore, we set a

probability that a packet will drop due to channel fading and denote it by q.

If $p \leq \alpha$, the monitoring node rejects the hypothesis H_0, meaning that the monitored node is identified as a malicious node. Otherwise, the monitoring node accepts the hypothesis H_0. The whole process is summarized in Algorithm 1.

Algorithm 1 Monitoring packet dropping

Input: q : the probability of a packet being dropped
α : level of significance
N : sample size of observed packets

Variables: n_d : the number of dropped packets
p : *P-value*
j : counter

Output: Reject H_0 or Accept H_0

```
1: n_d ← 0;
2: j ← 1;
3: while j ≤ N do
4:    The monitoring node observes how the monitored node handles a received
      packet not destined for himself;
5:    j ← j + 1;
6:    if The monitored node drops the received packet then
7:       n_d ← n_d + 1;
8:    end if
9: end while
10: Calculate p according to (3.1);
11: if p ≤ α then
12:    return Reject H_0;
13: else
14:    return Accept H_0;
15: end if
```

All nodes are first identified as benign nodes, which means no packet is expected to be dropped intentionally by a benign node. In such case, the *P-value* is calculated such that $p > \alpha > 0$. Contrarily, a malicious node is expected to drop

more packets to reduce network performance. Therefore, the more the number of packets dropped, the more $p \to 0$. With this approach intentional dropping of packets can be easily detected on every link in the network using the calculated *P-value p*.

3.3.7 Packet History Field Monitoring

Each data packet contains a route history in the packet history field of the packet header, which records all events occurring to the packet, such as receiving or forwarding of a packet. To achieve this, a node creates a packet history field which is added to the packet header. The packet history field consists of the packet route and node signature. Intermediate nodes on the route to the destination append their signatures to the packet history field when they receive the packet. The signature serves as a confirmation for accepting a packet. Similarly, the destination node appends its signature on the packet route history field and replies to the source node with an acknowledgment packet which is used to confirm end-to-end transmission delivery.

The packet history field also contains the source node signature. Each field used for signatures of the intermediate nodes is time stamped. This allows the neighboring nodes to determine the delays at each node and to prevent modifications. The following information are stored in the packet history: time stamp, packet route and node signature.

3.3.8 Detecting intentionally delayed packets

When receiving a packet, a benign node will insert the packet at the end of a packet queue that is served in First-In-First-Out manner (FIFO). However, a malicious node may intentionally delay inserting or removing the received packet into/from the queue, resulting in additional packet delay at that node. Packet delay at a node is defined to be the time interval from the time a packet is received by the node to the time that packet is transmitted.

Nodes that overhear packets can determine the packet queue order of their neighbors by checking the timestamp each time a packet is forwarded by a neighboring node to another node. If the packet is not delayed, the order of the packets

will not change. However, if a node intentionally delays a packet, the order of packets in the queue changes. This can easily be detected by a neighboring node that is overhearing packets. Our method also ensures that there is time synchronization between benign nodes. Neighbors with unsynchronized time are treated as malicious.

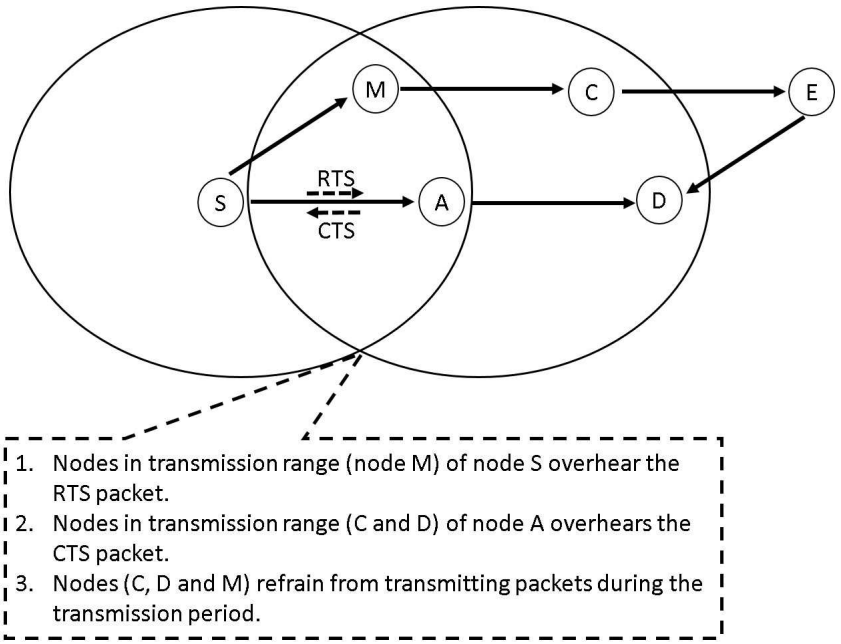

Figure 3.11: An example of RTS/CTS transmission process

To detect a node that is intentionally delaying packets, the RTS/CTS mechanism is used. The RTS/CTS packets contain information about the transmission duration of the data packet. This is used by other nodes to determine the estimated completion time of data transmission. If a CTS packet is overheard, other nodes wait until data packet, and ACK transmission are completed. Also, even if some links are busy and a node cannot send out packets, it can still receive a packet from another node.

In a situation where a node deliberately holds onto a packet after indicating its availability to receive and forward the packet by responding with a CTS packet. The previous node will not overhear the packet and then report such node as malicious. Hence, transmission collisions can be avoided and malicious nodes intentionally delaying packets are detected. Figure 3.11 shows an example of

RTS/CTS transmission.

In addition, a malicious node delay responding to an RTS packet might force the sending node to hold onto a packet, thereby delaying transmission. The sending node after the specified time elapses for receiving a CTS packet will select another route to send its packet.

3.4 Preventing Various Kinds of Attacks

In this section we explain how our monitoring scheme prevents Byzantine attacks. Specifically, we focus on preventing corruption of the routing table, wormhole attacks, colluding attacks, blackhole attacks, and delaying packets.

3.4.1 Corruption of Routing Table

A malicious node may try to corrupt the routing table information by advertising the wrong link delay to its neighbor or not adding a node as a neighbor in its hello message. In a situation where a node advertises a link delay that is better than the actual situation as described in Figure 3.1, where malicious nodes M advertises the link delay to node C to be 1, while node C advertises its link delay to the malicious node M as 10. Other benign nodes in the network will get conflicting information from the nodes connected to such a malicious node. In such a situation, nodes in the network will adopt the worse link delay. Similarly, if a node advertises a link delay that is worse than the actual situation while a neighbor node to such node advertise the actual delay cost, other nodes in the network will still adopt the worse link delay. This is not a problem since the link between these two nodes is a link that should be invalidated.

3.4.2 Wormhole Attack

As shown in Figure 3.4, if a node selected to take part in the routing of packets from the source to the destination decides to carry out a wormhole attack, it will do this by tunneling packets to another malicious node in the network which eventually drops the packets or selectively drops some packets. To prevent this, the neighboring node S overhears the packets, and detects the malicious action

by observing how malicious node M_1 handles the received packet. Neighboring nodes such as node S also check the packet history field signed by the malicious node M_1 to determine the past activities of the packet, and check if node M_2 is part of the packet route by comparing the sending node address to the packet route information stored in the packet history field (e.g. node ID or MAC address in the packet header to the one stored in the packet history field). If node M_2 is not stored as part of the packet route information, the neighboring node reports that node M_1 is a malicious node to other nodes in the network. So recording and checking the route information prevents packet tunneling and wormhole attacks.

When this occurs, Node S will report that node M_1 is malicious to other benign nodes in the network, and the link between node S and malicious node M_1 will be excluded. Again, node S will afterwards select another path and retransmit its packets using the two hop reactive mode.

3.4.3 Black Hole Attack

As shown in Figure 3.3, if a node decides to drop or ignore packets, thereby carrying out a black hole attack, the source node S and node A will not overhear the packet. In this case, after a predetermined time interval without node S and node A overhearing the packet, and the statistical method described earlier detects that node M is malicious. Then the links between node S and node A to node M are excluded from the network, and the nodes report that node M is malicious to other nodes in the network. Afterwards, source node S selects another path for its packets and retransmits its packets using the two hop reactive mode.

3.4.4 Preventing Colluding Attacks

In our scheme, there are two main types of colluding attacks we address. The first colluding attack scenario is when only one node M_1 of the colluding nodes is part of the selected path and it forwards the packet to the second colluding node M_2 on a non-optimal path. This type of colluding attack is similar to the wormhole attacks discussed in subsection 3.4.2 and can be prevented in a similar way as described above, that is neighboring nodes such as the source node S can

check if the colluding node M_2 is part of the packet route by comparing node M_2 address (e.g. node ID or MAC address) to the packet route information stored in the packet history field. If node M_2 is not stored as a part of the packet route information, the neighboring node S reports that node M_1 is a malicious node to other nodes in the network and the link between node S and node M_1 is excluded from the network. Thereby this form of colluding attack is prevented.

In the second colluding attack scenario, suppose node S selected a route S - A - M_1 - M_2 - D which includes two malicious nodes M_1 and M_2 and the packet is dropped at node M_2 but node M_1 fails to report such malicious action. After a predefined time for receiving the ACK from the destination node D by node S has passed and the ACK is not received, node S requests from node A the overheard packet which includes the signed packet history field that confirms that packet from node S is forwarded to the next hop by node M_1. In this situation, if node M_1 fails to show the overheard packet from node M_2, then node M_1 is reported as malicious and the link between node A and node M_1 will be excluded from the network.

Additionally, to prevent a situation in which node M_1 colludes with M_2 such that node M_1 gets node M_2's private key and uses it to make a fake overheard packet (i.e., node M_1 signs the packet history field as node M_2) and shows the fake overheard packet to node S as proof that the packet was forwarded from node M_2. Each node showing the overheard packet must also show the CTS packet received from the next hop node, which can be compared to the previous overheard RTS/CTS packets. After receiving the fake overheard packet, the source node S will request the destination node D to resend the ACK to confirm that the destination node D receives the packet from node M_2, then the destination node D can report not receiving the packet (see Figure 3.12). Also, monitoring nodes can detect such packet dropping behavior by node M_2 by using the statistical hypothesis testing method. If the statistical hypothesis testing method confirms that node M_2 is malicious and there is no ACK received by the destination node D, then the link between node M_2 and the destination node D will be excluded from the network.

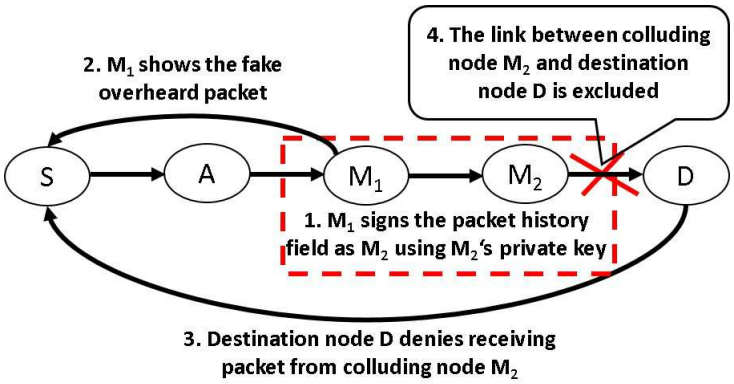

Figure 3.12: Colluding attacks using fake overheard detection

3.4.5 Preventing Intentional Packet Delay Attacks

In this situation, a malicious node M deliberately delays packets in the network, e.g. using the network in Figure 3.3. When this happens, source node S will overhear the packet, and check the order of packets in the packet queue for node M. Node S also checks whether it has previously overheard RTS/CTS packets from node M. If the queue order of the packets has changed and node M has not been sending and receiving RTS/CTS packets, then node S determines that node M is maliciously delaying packets. Node S reports node M as a malicious node to other nodes in the network.

3.5 Security Goals

When our proposed monitoring scheme is run successfully, it achieves the following security goals.

3.5.1 Authentication

In our scheme, there is no certificate authority, therefore each node creates its unique public and private keys in advance. The unique key pair is safely created from random numbers by any node. The unique key pair is used to generate and verify a digital signature which is used for authentication in our schemes. When a node sends/forward packets, the node will sign its ID which can be verified by

other nodes to authenticate the sender of such packets. Also, the packet history field used to store the route history information is digitally signed with source node *S's* private key.

3.5.2 Confidentiality

All data in the packet are encrypted and digitally signed by users. When the source node S sends a packet to destination node D, the data message in the packet is encrypted with a session key created using the symmetric key algorithm and the session key is encrypted with the destination node *D's* public key so none of the intermediate nodes on the route to the destination in the network can decrypt the message. Hence, confidentiality of the message between the source node S and the destination node D is maintained.

3.5.3 Non-repudiation

When a packet is sent by a source node S to a destination node D, each intermediate node appends their signature to the packet history field as a confirmation of receiving the packets and records a timestamp when the node forwards the packet to confirm forwarding the packet. To achieve non-repudiation in the network, each intermediate node digital signature and timestamp is verified whenever the previous node overhears packet forwarding by the intermediate nodes. An intermediate node cannot deny appending its signature to the packet history field.

3.5.4 Integrity

To ensure that the packet and the packet history field are not tampered with by a malicious node or an intermediate node, the source node S signs its signature on the packet history field which can be verified wherever the packets are overheard. The digital signature and timestamp from the source node S ensure the integrity of the packets and the information in the packet history field. Also, an attacker cannot modify the data packet, as it is encrypted with a session key and the session key is further encrypted with the destination node's public key.

3.6 Results and Discussion

In this section, we evaluate the performance of our proposed monitoring scheme for securing Link State Routing against Byzantine attacks using a custom simulator. First, we evaluate our method using a static network. Then we further show the evaluation of our proposed system using a network with a mobility scenario. The performance of our proposed monitoring scheme is evaluated using a custom simulator. In our simulation, we implement all parts of our proposed protocol and focus our evaluation on the packet dropping attacks such as black hole attacks, wormhole attacks, and colluding attacks, where a malicious node selectively or completely drops received packets.

3.6.1 Simulation Configuration for a Static Network

The main objectives of our simulation are to validate (i) that our monitoring scheme guarantees communication among benign nodes, and (ii) detect if a node is dropping packets deliberately.

The simulated scenario is implemented to enable benign nodes to connect with each other easily within the transmission range, given that mutual monitoring is an important aspect of our protocol. Nodes are first evenly placed in a 3km x 3km area as shown in Fig 3.13 with malicious nodes randomly selected to maintain a uniform position in the network. We varied the number of malicious nodes from 2 to 12 out of the 20 network nodes. The skeleton map represents the network topology. Each node maintains low mobility in the network. The route is based on Dijkstra's shortest path algorithm. Each node randomly selected a destination node and calculates the shortest path to the destination node. Then each node generates a packet every second, the packet is routed to the selected destination node using our secured link state routing protocol. The total number of packets generated in the network is 50000 packets. In our simulation, we eliminate sending packets on selected paths with just 1 or 2 hops. We used paths with 3 hops and above to route packets. All nodes have the same buffer size and transmission range. We assume 802.11g wireless WiFi (802.11g comes with ad-hoc mode) is used for communication. The network bandwidth of our simulation is set to 1 Mbps. We set the logical packet size to 5KB, a logical packet is a combination of

several physical frames.

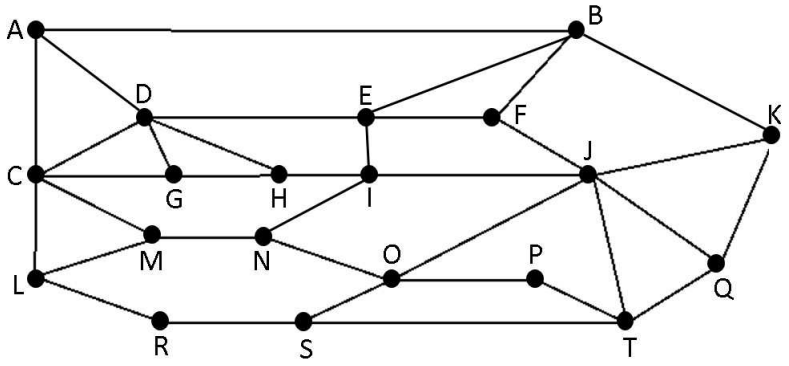

Figure 3.13: Network topology for simulation

The probability that a packet is being dropped due to network fading is set to 0.001 while the probability that malicious nodes dropped a packet is set to 0.5. Each link in the network is monitored for 1000 packets. Links between nodes are identified as a malicious link if the calculated P-value p is less than the level of significance α which is set to 0.0001. All simulated results in the figures below are the averages of 30 simulation runs. The summary of the default values used in our simulation is shown in Table 3.1.

The following metrics will be measured in our simulation.

Packet delivery ratio: This is the ratio between the number of packets successfully delivered to destinations and the number of packets generated by sources. Not all packets can be successfully delivered to destinations due to reasons like malicious node dropping packets, buffer overflow, etc.

False positive ratio: This is the ratio between the number of links between benign nodes that are falsely removed as malicious and the number of all network links. It is desirable to have a low false positive ratio.

Malicious link detection ratio: This is the ratio between the number of successfully detected malicious links and the number of all malicious links in the

Table 3.1: Simulation Parameters

Parameter	**Value**
Simulation time	3600 seconds
Bandwidth	1 Mbps
Buffer Size	100-500KB
Transmission range	250m
Network size	3km x 3km
Mobility	Static
Number of nodes	20
Number of malicious nodes	2 – 12
Level of significance (α)	0.0001
Probability of packet dropped due to fading (q)	0.001
Probability of packet dropped by malicious nodes	0.5
Sample size of observed packets (N)	1000
Total number of packet generated	50000

network.

Malicious node packet dropping ratio: This is the ratio between the number of packets dropped by malicious nodes and the number of packets generated by the sources.

3.6.2 Packet delivery ratio

In our simulation, to show that our system guarantees communication among benign nodes in the network we estimate the packet delivery ratio. We set the probability that a packet is dropped due to channel fading to 0.1% and the probability that a packet is dropped deliberately by a malicious node is set at 50%. Each monitoring node observes the link between two nodes for 1000 packets and counts the number of packets that is being dropped before calculating the *P-value* that such node is malicious or not. If the P value is greater than the level of significance, the link between the two nodes is further monitored for another 1000

packets until a malicious node on the path is detected and the link is excluded. In our method, malicious links require 1000 packets to be detected. We evaluate the number of packets that are successfully delivered to the destination node. As shown in Figure 3.14, our proposed monitoring scheme achieved an average of 89% to 96% packet delivery ratio. The packet delivery ratio reduces as the number of malicious nodes in the network increases. The reason for the higher packet delivery ratio when the number of malicious nodes is set between 2 - 6 as compared to 12 malicious nodes is that the benign nodes have more alternative routes to send their packets to the destination nodes. Hence, more packets are delivered. The more the number of malicious nodes, the more the number of packets that are dropped and vice versa. In addition, our proposed scheme still achieved 89% packet delivery ratio even when more than half of the network nodes (i.e., 12 out of the 20 nodes in the network) are malicious.

Moreover, the proposed scheme is compared to a case without the proposed scheme. In such a situation, malicious nodes drop packets without being detected. As shown in the figure, the packet delivery decreases as the number of malicious nodes increases when there is no secure scheme adopted in the network. The proposed scheme achieves a better performance in packet delivery compared to when no secure scheme is used. Using the proposed scheme, the packet delivery ratio is increased by 17%, 32%, 42%, 52%, 64% and 62% compared to when no secure scheme is used with 2, 4, 6, 8, 10, and 12 malicious nodes respectively.

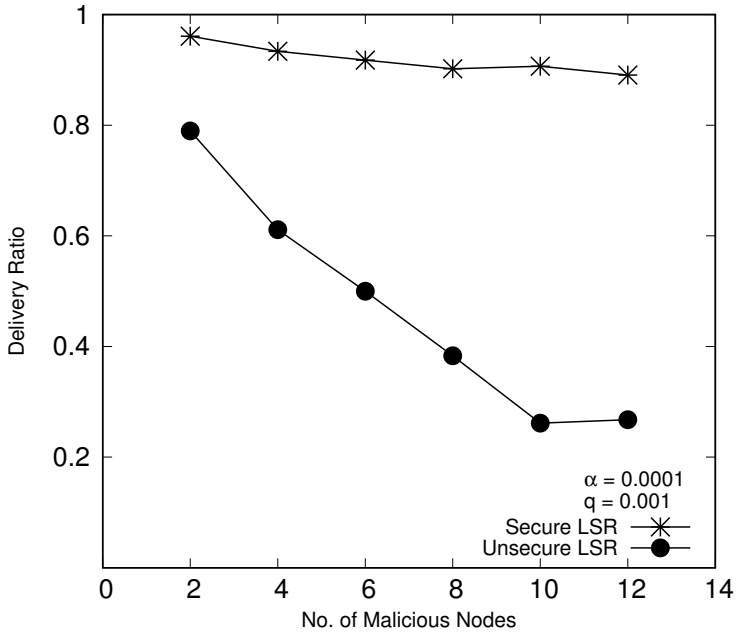

Figure 3.14: Packet delivery ratio.

3.6.3 False positive ratio

In our scheme, each node monitors how the received packet is being handled by the intermediate node. When a monitoring node overhears a packet, the monitoring node calculates the P value p and compares it with the level of significance α. If the P value is less than the level of significance, the node being monitored is detected as malicious and reported to other nodes in the network. Therefore, the link between the malicious and the benign node is excluded from the network. However, it is possible that a malicious node reports other benign nodes as being malicious, thus we set the frequency of malicious nodes reporting other benign nodes as being malicious in the simulation to 0. Any malicious node with a frequency greater than 0 is excluded in the network. Therefore, such malicious node behavior is quickly detected and further influence of such malicious behavior is prevented.

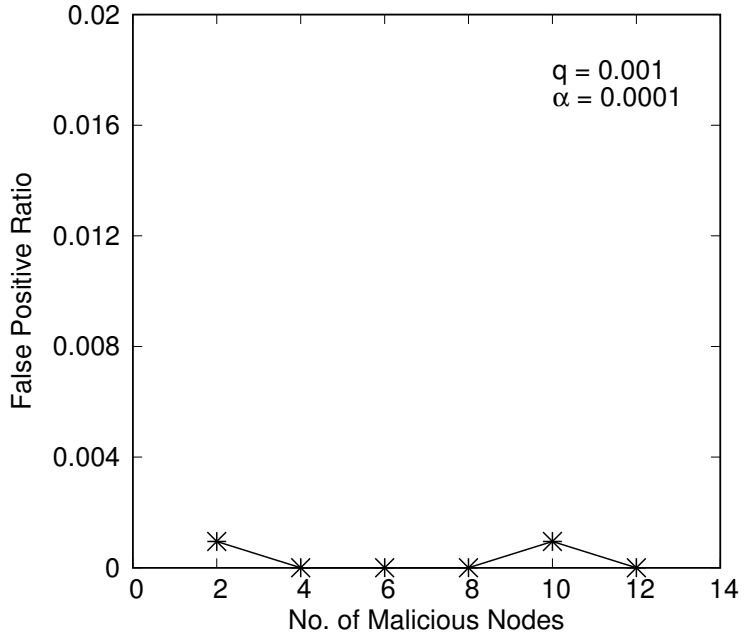

Figure 3.15: Link falsely detected.

In our simulation, we evaluate the false positive ratio of links among benign nodes that are falsely detected as being malicious links. Figure 3.15 shows that the false positive in our scheme is significantly lower with an average of 0.1% false positive ratio when 2 and 10 of the network nodes are malicious.

3.6.4 Malicious link detection ratio

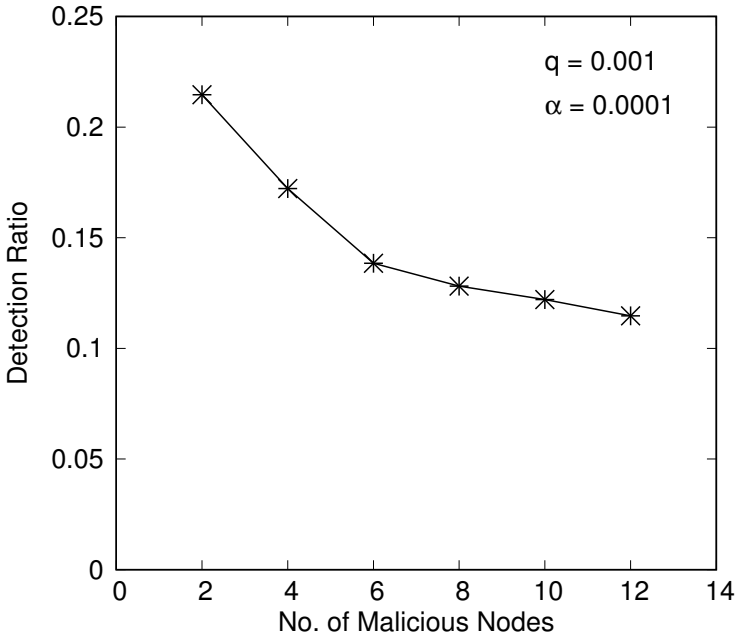

Figure 3.16: Link successfully detected.

Figure 3.16 shows the detection ratio of malicious links that are excluded from the network after being detected as malicious links. The overall average malicious links in our network is 16 links. Our proposed monitoring scheme achieved an average of 11% to 21% malicious link detection ratio. The malicious link detection ratio decreases as the number of malicious nodes increases. The goal of our method is to detect malicious links that are utilized in routing a packet from a source node to a destination node and guarantee secure communication among benign nodes. By excluding 11% to 21% of the active malicious links in the network, our method achieves higher packet delivery ratio as shown in Figure 3.14. In addition excluding all malicious links all at once in the network may result in network partition.

3.6.5 Malicious nodes packet dropping ratio

As shown in Figure 3.17, the packet dropping ratio of malicious nodes increases as the number of malicious nodes increases. In our simulation, 10% of the packets

are dropped when 8 out of the 20 nodes in the network are malicious while 11% of packets are dropped even when more than half of the nodes in the network are malicious (12 out of the 20 nodes). The reason for the lower packet drop rate in the network is due to the early exclusion of some malicious links which further isolates other malicious nodes from being selected in packet routing without causing a network partition among benign nodes.

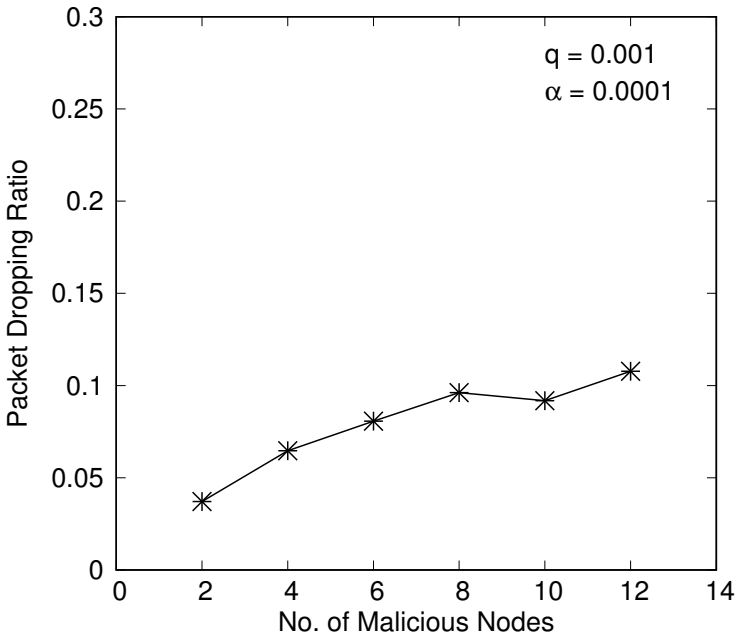

Figure 3.17: Malicious nodes packet dropping ratio.

3.6.6 Communication Overhead

In this section we discuss the overhead cost introduced by our monitoring scheme using the speed benchmark for cryptographic algorithm: River Shamir Adelman (RSA) with 1024 bit key size and ECDSA 192 bit key size digital signature algorithm [43], run on an Intel Core 2 1.83 GHz processor under Windows Vista in 32-bit mode x86/MMX/SSE2.

The packet overhead of our scheme is divided into three parts: the first is the packet history field size required to send a packet from a source node to a destination node. The second is the hello message size a node exchanged with

its neighbor nodes and the last part is the computational cost of signing and verifying the digital signature of intermediate nodes by the monitoring nodes.

3.6.7 Packet History Fields Size

In our proposed scheme, for a node to transmit a packet to a destination, the node needs to create packet history fields which are used for monitoring if the packet is forwarded on the right route. Each intermediate node appends its signature on the packet history field, therefore, we need to calculate the additional overhead introduced by our monitoring scheme to transmit a packet from a source node to a destination node. First, we find the number of signatures appended to the packet history field between the source node to the destination node *(N_S)*, and then the size of the signature appended by the intermediate node *(s)*.

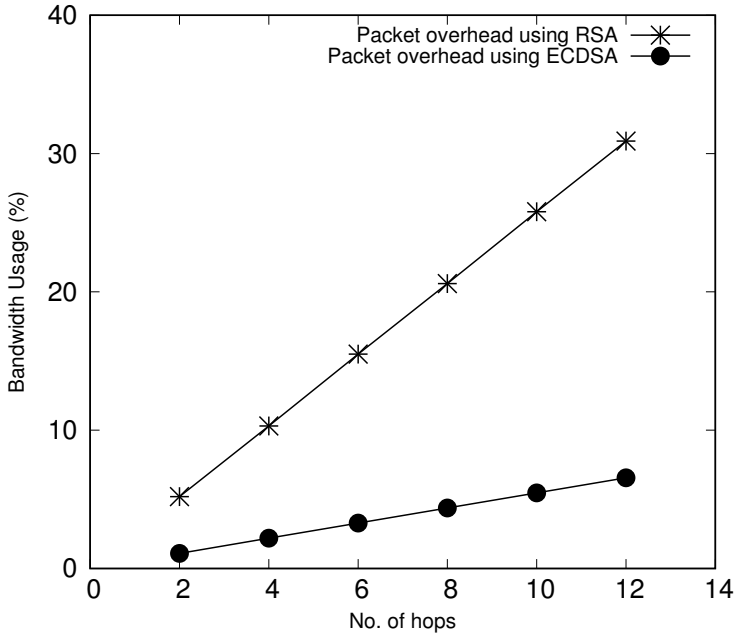

Figure 3.18: Packet overhead.

In addition, each source node stored the optimal path for sending its packet to the destination node in the packet route fields of the packet history. We also calculate the overhead introduced by this route specification *(RS)* in our protocol. The size of each next hop ID stored in the packet route fields from the

source node to the destination node is 4 bytes. Therefore, the total overhead cost needed to route a packet to destination in our scheme is calculated as: $T_{overhead} = (N_S \times s) + RS$. The size of the packet history field increases as the number of nodes increases. We compare the size of the packet history field using the RSA (128 bytes) and ECDSA (24 bytes) signature. The average total overhead in term of size of our packet history field is 1320 bytes and 380 bytes respectively, when the number of nodes in the route is 10.

Each packet needs to store extra information about intermediary nodes, which are the node ID of 4 bytes and signature of 24 bytes for ECDSA which form a total overhead of 28 bytes per hop. Let's say there are 10 hops between a source and destination. We assume that the packet size is 5KB, therefore, the additional traffic generated will utilize 5.5% ($\frac{28\times10}{5\times1024}$) of the bandwidth. Figure 3.18 shows the overhead of our secure monitoring scheme when the number of hops from source to destination is varied from 2 to 12. The scheme achieves better performance with an average of 7% maximum bandwidth utilization. There is a 4%, 8%, 12%, 17%, 21% and 25% decrease in bandwidth utilization using the ECDSA signature algorithm as compared to the RSA signature with hop numbers of 2, 4, 6, 8, 10, and 12, respectively.

3.6.8 Hello Message Size

In our protocol, each node exchanges hello messages with its neighbor nodes and appends to its hello message the information from its neighbor hello message such as neighbor ID, link cost, neighbor signature and timestamp of the hello message. Therefore, we evaluate the size of the hello messages in our protocol as compared to a typical hello message size of LSR protocol. The size of the hello message depends on the number of nodes in the network, each node distance, the number of hello message broadcast by neighbor nodes and their link status. To determine the size of the hello message at each node we estimate the size of the hello message in our scheme and compare it with that of the standard LSR without any security measure. The average size of a standard LSR hello message is 48 bytes when only one node is advertised as a neighbor while the hello message in our scheme is 320 bytes and 112 bytes using RSA and ECDSA signatures. Figure 3.19 shows that the LSR hello message size increases from 52 bytes to 92 bytes as the number

of nodes advertised as neighbor increases. Comparatively, the hello message size of our scheme increases from 461 bytes to 1871 bytes when the RSA signature is used. However, by adopting the ECDSA signature we reduce the hello message size of our scheme by 67% to 72% when we varied the advertised neighbors from 2 to 12 nodes.

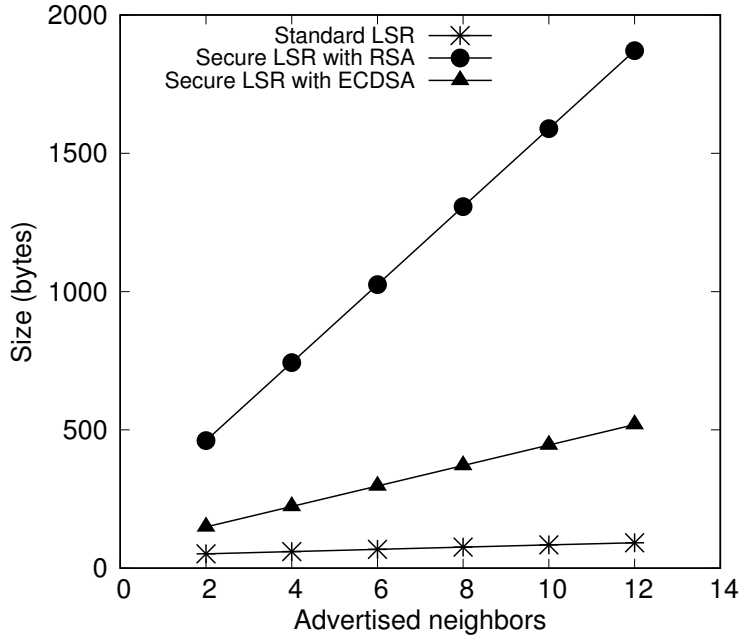

Figure 3.19: Hello packet size.

In addition, there are four additional packets (i.e. database descriptor, link state request, link state update and link state acknowledgment) in LSR protocol, that use a common 24-bytes header as the hello message. We adopt the implicit acknowledgment where a neighbor that received a packet makes a duplicate and encode it to its ACK, then send the ACK back to the sending node. Multiple neighbors can be acknowledged in a single multicast ACK packet.

3.6.9 Computation Overhead

Each node on the selected route generates its signature, which is appended to the packet history field and verified by monitoring nodes in the network. The average computation time for generating a signature by each node on the route is 0.002

seconds while the verification time by monitoring nodes that overhears the packet is less than 0.0001 seconds. Using the network in Figure 3.10, with 5 nodes on the packet route, the total processing time used for generating signatures by each node on the route is 0.01 seconds while a total verification time used by nodes to verify the packet history field is 0.0003 seconds. Hence, the computational overhead of our scheme is adequately low.

3.6.10 Simulation Configuration of a Network with Mobility Scenario

In our system, we adopt MANET, as a result of the dynamic nature of MANET the packets' routes and the routing information changes frequently changes due to nodes' mobility. Therefore, we need to further evaluate our method effectiveness in a network with a mobility scenario. The main objectives of our simulation are to validate that (i) our method can still achieve better performance when MANET is utilized, and (ii) achieve high packet delivery ratios, and (iii) guarantee communication among benign nodes and ensure that better paths are utilized in the network.

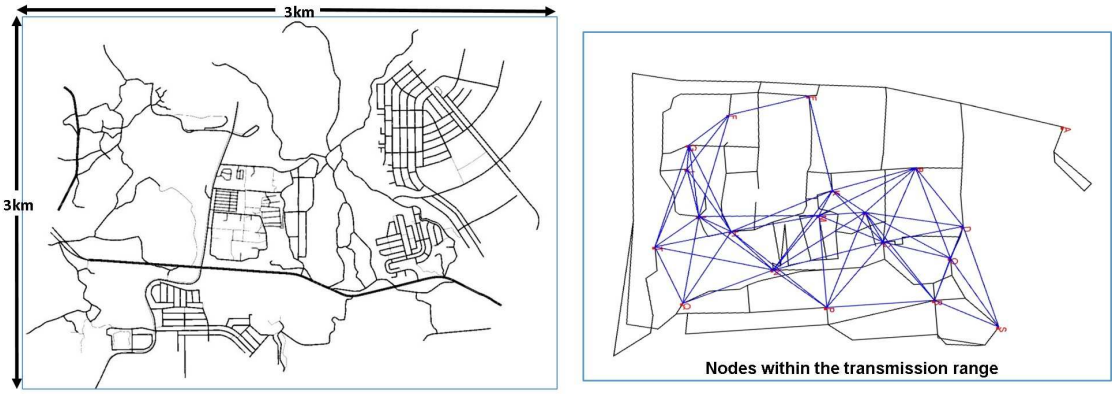

Figure 3.20: Map for Simulation.

The simulated scenario is implemented to enable nodes to connect with each other easily within the transmission range. Mobile nodes are first evenly placed in a 3km x 3km area. This is based on an actual map of the area around Nara Institute of Science and Technology in Nara, Japan, as shown in Fig 3.20. The

Table 3.2: Simulation Parameters

Parameter	**Value**
Simulation time	50000 seconds
Bandwidth	1 Mbps
Buffer Size	100-500KB
Transmission range	200m
Network size	3km x 3km
Mobility	Random Waypoint
Node speed	1 - 1.4m/s
Number of nodes	20
Number of malicious nodes	2 – 12
Level of significance (α)	0.0001
Probability of packet dropped due to fading (q)	0.001
Probability of packet dropped by malicious nodes	0.5
Sample size of observed packets (N)	100
Total number of packet generated	50000

skeleton map represents the road network. Each node moves according to the Random waypoint mobility model [27] at a uniform speed of 1 to 1.4m/s. We varied the number of malicious nodes from 2 to 12 out of the 20 network nodes with 190 links connecting all nodes. We maintained similar simulation settings as the static network setting, however, as a result of nodes' mobility malicious nodes can be connected to all benign nodes when in the transmission range. Hence, we reduced the number of packets monitored on each link from 1000 to 100 in this simulation. Also, we measure the same simulation metrics in the simulation. The summary of the default values used in the simulation is shown in Table 3.2.

3.6.11 Packet delivery ratio

To test the performance of our method using a network with mobility scenario, we further evaluate the delivery ratio in such network. Similar to the settings of the previous simulation, we set the probability that a packet is dropped due to channel fading to 0.001% and the probability that a packet is dropped deliberately by a

malicious node is set at 50%. Due to the node's mobility in the network, malicious nodes are connected to all nodes, therefore, each monitoring node observes the link between two nodes for 100 packets and counts the number of packets that are being dropped before calculating the *P-value* to detect that such a node is malicious or not. In this simulation, malicious links require 100 packets to be detected. We evaluate the number of packets that are successfully delivered to the destination node.

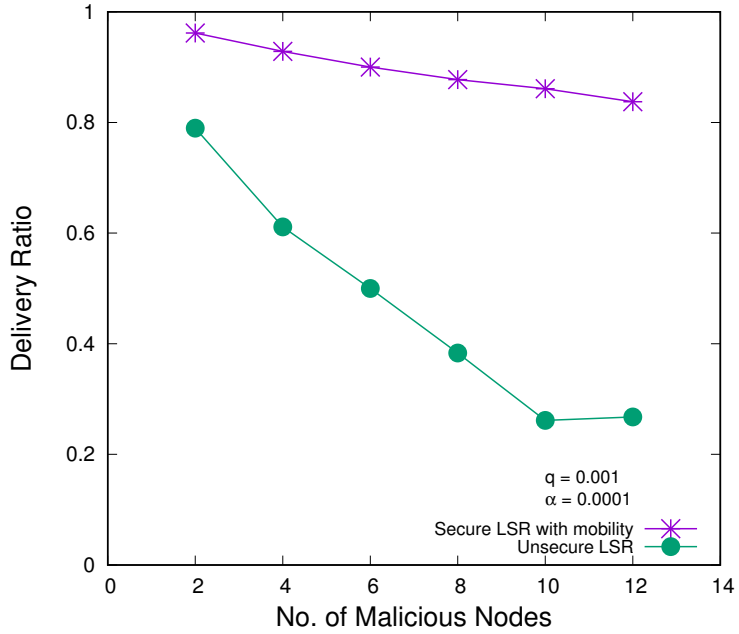

Figure 3.21: Packet delivery ratio.

As shown in Figure 3.21, our proposed monitoring scheme achieved an average of 84% to 96% packet delivery ratio. The packet delivery ratio decreases as the number of malicious nodes in the network increases. The reason for the higher packet delivery ratio when the number of malicious nodes is set between 2 - 6 as compared to 12 malicious nodes is that the benign nodes have more alternative routes to send their packets to the destination nodes. Hence, more packets are delivered. The more the number of malicious nodes, the more the number of packets that are dropped and vice versa. There is a slight decrease in the delivery ratio when compared with the static network. The proposed method

also achieved an average increase of 17% to 58% when compared to the unsecured LSR packet delivery ratio.

3.6.12 Malicious link detection ratio

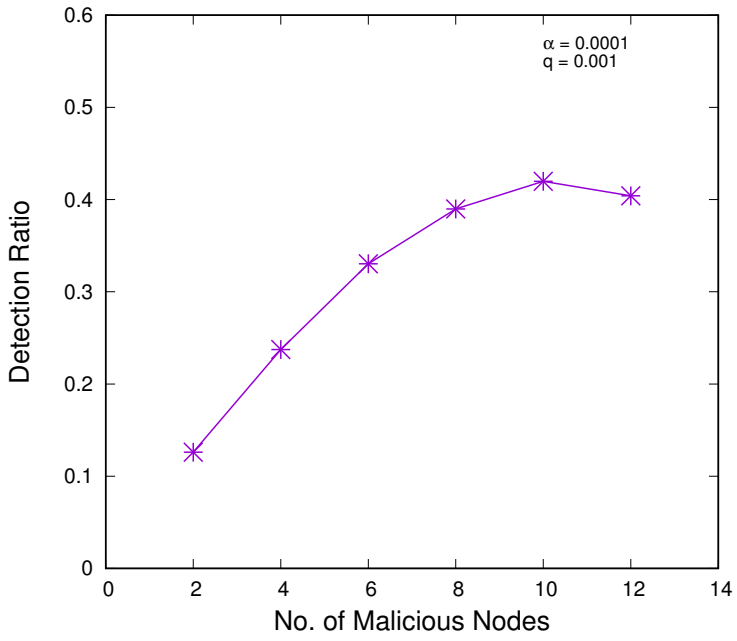

Figure 3.22: Packet delivery ratio.

Figure 3.22 shows the detection ratio of malicious links that are excluded from the network after being detected as malicious links. The overall average malicious links in our network are 106 links. As a result of the frequent changes in routes more malicious links are utilized to route packets. Therefore, to effectively detect such malicious links, we reduced the number of the monitored packets on each link from 1000 to 100. The simulation results indicate that our proposed monitoring scheme achieved an average of 13% to 42% malicious link detection ratio. The malicious link detection ratio increases as the number of malicious nodes increase unlike in the static method where the malicious link detection decreases as the number of malicious nodes increases.

3.6.13 False positive ratio

Another simulation metric that we measure for the network with a mobility scenario is the ratio of links that are falsely detected as being malicious. Therefore, we evaluate the false positive ratio of links among benign nodes that are falsely detected as being malicious links. Similar to the results of the static network, Figure 3.23 shows that the false positive ratio in our scheme is significantly lower with an average of 0.04% false positive ratio when 2 of the network nodes are malicious. This further validates the effectiveness of our method in a network with nodes' mobility.

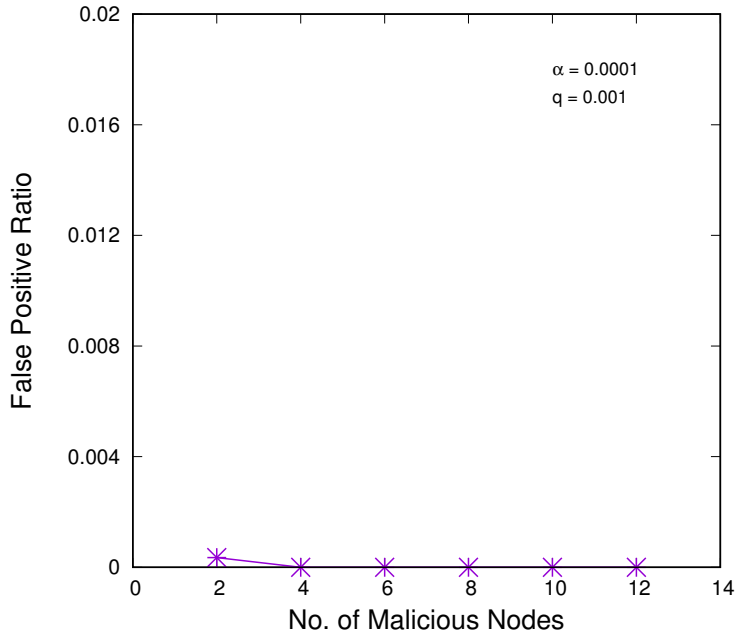

Figure 3.23: Link falsely detected.

3.6.14 Malicious nodes packet dropping ratio

In addition, we evaluate the malicious nodes packet dropping rate and compare it to the theoretically calculated packet dropping rate. We calculate the theoretical packet dropping as follow: $\frac{No.maliciousnodes \times No.of expecteddroppedpackets}{Totalno.of Packets}$. For example, if the number of malicious nodes is 12, then the calculated packet dropping ratio is 0.12 ($\frac{12 \times 500}{50000}$). As shown in Figure 3.24, the packet dropping ratio of malicious

nodes increases as the number of malicious node increases. There is an average of 4% to 16% malicious packet dropping ratio when the number of malicious nodes varies from 2 - 12 in our simulation. 10% of the packets are dropped when 8 out of the 20 nodes in the network are malicious, while 11% of packets are dropped even when more than half of the nodes in the network are malicious (i.e., 12 out of the 20 nodes). In comparison with the theoretical evaluation where there is an average of 2% to 12% malicious packet dropping ratio, the malicious packet dropping ratio of our method is slightly higher than the theoretically calculated packet dropping ratio with an average increase of 2% to 4%. However, this is still significantly lower than the settings of packet dropping ratio in our simulation. The reason for the lower packet drop rate in the network is due to the early exclusion of some malicious links which further isolates other malicious nodes from being selected in packet routing without causing a network partition among benign nodes.

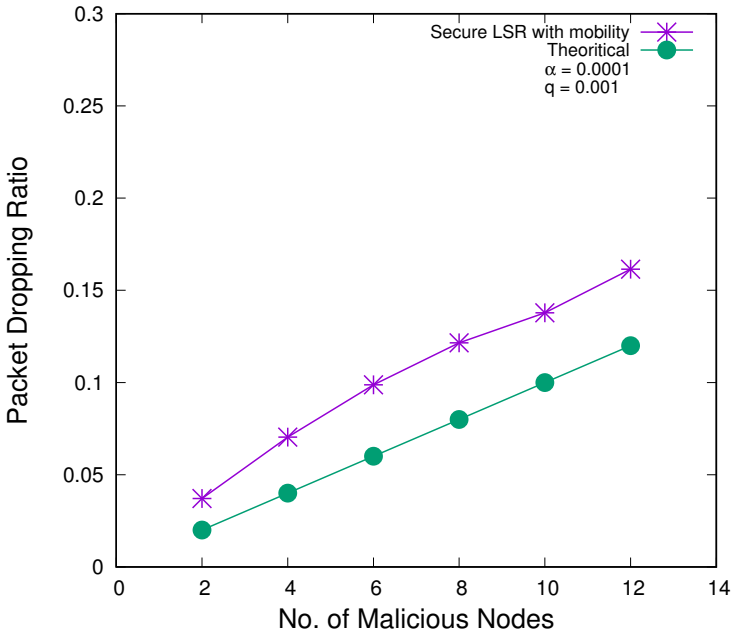

Figure 3.24: Malicious nodes packet dropping ratio.

3.7 Conclusion

In this chapter, we propose a monitoring scheme to secure link state routing against Byzantine attacks. We adopt the statistical hypothesis testing to determine if a node is intentionally dropping packets. In addition, our monitoring scheme also uses hello message verification to validate the hello messages and to identify inconsistent information when a malicious node tries to corrupt routing table information, and packet history field. The monitoring node checks whether packets are forwarded correctly according to the stored optimal path. Also, our monitoring scheme uses RTS/CTS to identify when a node is intentionally delaying packets in the network.

Our approach guarantees communication among benign nodes as validated by simulation with an average of 89% packet delivery ratio even when 12 out of the 20 nodes in the network are malicious. Also, simulation results show a significantly lower false positive ratio with an average of 0.1% while the malicious link detection ratio is between 11% to 21%. Active malicious links to benign nodes are detected and excluded from the network, thereby guaranteeing communication among benign nodes. In addition, our monitoring scheme achieves relatively low communication overhead with an average of 299 bytes packet history field size and 519 bytes hello message size. Also, the communication overhead for generating and verifying signature by each node is less than 1 second (0.002 seconds and 0.0001 seconds respectively).

We further evaluate our proposed method using a network with a mobility scenario. The simulation results show that our method is effective in a network with a mobility scenario. There is an average of 84% to 96% packet delivery ratio and a significantly lower false positive ratio an average of 0.04% while there is an increase in the malicious link detection ratio with an average of 13% to 42%. Therefore, our method achieves better performance when static network and MANET are utilized.

4 Conclusion and Future Work

In this dissertation, we addressed the problem of a secure commerce system in a disaster area where resources to support such systems are limited. In Chapter 1, we first highlighted the importance of financial services in disaster relief management and how important having access to a form of payment or cash is for people in a disaster area. However, such financial services cannot function without the support of communication infrastructures which may be damaged after a large-scale disaster. Then we state the problems that need to be addressed in our research before a secure commerce system can be implemented and the scope of our research.

In Chapter 2, a new mobile payment system is proposed to enable electronic transactions in disaster areas. An endorsement-based scheme is introduced to provide a payment guarantee to a merchant for a customer's transaction. We adopt various protocols such as event chain, e-coin, Bloom filter, Merkle tree and mutual tracking mechanism to prevent double spending, colluding, reduce overhead and secure our mobile payment system. Also, a digitally signed photo is used for authentication and to restrict an attacker while a blind signature technique is adopted to protect user's privacy. Finally, we evaluate the performance of our proposed secure payment system by simulation to test the usability in disaster areas. As validated by simulations, the proposed mobile payment system is useful in a disaster area, achieving a high transaction completion ratio, 65% - 90% for all scenarios tested, and is storage-efficient for mobile devices with an overall average of 7MB merchant message size.

In Chapter 3, a monitoring-based approach to secure the link state routing protocol against Byzantine attacks is proposed. We highlighted the security goals our proposed monitoring approach achieves. Each node monitors the actions of neighbouring nodes and compares the optimal packet route against the route his-

tory. This allows monitoring nodes in the network to track the past events of packets. In our method, we adopt three main schemes to identify inconsistent information, check whether packets are forwarded correctly according to the stored optimal path and to detect if a node is intentionally dropping packets. The proposed monitoring-based method achieves an average of 89% to 96% packet delivery ratio when 11% to 21% active malicious links are excluded from the network. Also, our method achieves better performance when MANET is utilized with an average of 84% to 96% packet delivery ratio and an average of 13% to 42% malicious link detection ratio.

In addition, our mobile payment system is not limited to disaster areas alone, it can be used in developing countries where access to a network is not constantly available. In this dissertation we proposed a MANET-based commerce system and focused on how to enable transaction for disaster areas. Then extend the scope of the work to securing and preventing various Byzantine attacks which may affects the commerce system transaction. In the context of future work of our research, we will focus on the development of a prototype application which can be implemented on real devices such as iPad and can be deployed to disaster areas. Also, we will consider how our monitoring approach can detect and prevent DoS attacks such as an overwhelming amount of traffic. The other future work includes the extension of our method to other MANETs and wireless sensor constraint such as the effect of power consumption, changes in node bandwidth and so on.

Acknowledgements

I give all the glory and honor to the Almighty God for giving me the opportunity to be alive to complete my course.

My greatest gratitude goes to Professor Minoru Ito for his support, assistance and encouragement in making my research possible. His invaluable comments and suggestions have been a great contribution to the success of my research.

My gratitude also goes to Professor Keiichi Yasumoto for his vital and helpful comments and discussion which has contributed greatly to the success of my research. I am sincerely grateful to Associate Professor Naoki Shibata, for his support, suggestions, advices and teachings without which this research would be impossible. I learned a lot from discussions during research meetings, which has helped my research and my personal life. I am most grateful to Professor David Sell and Professor Michael Barker for their assistance in proofreading my papers despite their busy schedules.

I also appreciate Assistant Professor Juntao Gao and Assistant Professor Tomoya Kawakami for their support, comments and contribution to my research. My deep appreciation goes to Mrs. Eri Ogawa, Mrs. Miki Ikeda, current and past students of Mobile Computing Laboratory, all African students in NAIST and other international students that have made staying in Japan enjoyable.

My greatest and deepest gratitude go to the Otsuka Toshimi Scholarship Foundation for the financial support. I really enjoyed all events that were organized by the foundation. I hope they continue to provide this great opportunity to other students. I would also like to thank the administration of Nara Institute of Science and Technology for selecting me as one of the NAIST Students to benefit from the Tuition Fee exemption program. I was able to fulfill my goal of pursuing my doctoral degree as a result of this. Also, to the staff of the International Student Affairs Section and the Graduate School of Information Science Office

for their help and assistance with all the important documents that made life as a student easier.

Finally, I would like to say a big thank you to my lovely wife - Adebola Ojetunde for her encouragement and support during my PhD degree. And to my parents, brothers, sisters, my in-laws and friends, thank you for your words of encouragement, prayers, and support. Finally, to my little princess Oluwasemilore Joy Ojetunde, your birth has brought so much joy, happiness and motivation to achieve more during and after this doctoral course.

References

[1] International Federation of Red Cross and Red Crescent Societies, "World Disasters Report 2016 - Resilience: saving lives today, investing for tomorrow," pp. 224 - 262. http://www.ifrc.org/Global/Documents/Secretariat/201610/WDR%202016-FINAL_web.pdf.

[2] R. Shaw (ed.), "Disaster Recovery: Used or Misused Development Opportunity, Disaster Risk Reduction," DOI: 10.1007/978\T1\textemdash4-431-54255-1_2, pp. 38 – 55. Springer Japan 2014.

[3] R. Federica; I. Mikio, "Learning from Megadisasters : Lessons from the Great East Japan Earthquake," ch. 9, pp. 84, Washington, DC: World Bank, 2014. © World Bank. https://openknowledge.worldbank.org/handle/10986/18864 License: CC BY 3.0 IGO.

[4] A. Mishra and K. M. Nadkarni, "Security in Wireless Ad Hoc Networks," in The Handbook of Ad Hoc Wireless Networks. Boca Raton, FL, USA: CRC Press, 2003, ch. 30, pp 499-549.

[5] W. Li, Q. Wen, Q. Su, and Z. Jin, "An efficient and secure mobile payment protocol for restricted connectivity scenarios in vehicular ad hoc network," Comput. Commun., vol. 35, no. 2, pp. 188-195, Jan. 2012.

[6] X. Dai, O. Ayoade, and J. Grundy, "Offline micro-payment protocol for multiple vendors in mobile commerce," in Proc. 7th Int. Conf. Parallel and Distrib. Comput. Appl. Technol. (PDCAT), Taipei, Taiwan, 2006, pp. 197-202.

[7] V. Patil, and R. K. Shyamasundar, "An efficient, secure and delegable micro-payment system," in Proc. IEEE Int. Conf. e-Technol. e-Commerce e-Service (EEE), Taipei, Taiwan, Mar. 2004, pp. 394-404.

[8] Y. Y. Chen, J. K. Jan, and C. L. Chen, "A Novel Proxy Deposit Protocol for E-cash Systems," Appl. Math. and Comput., vol. 163, no. 2, pp. 869-877, 2005.

[9] N. C. Kiran, and G. N. Kumar, "Implication of secure micropayment system using process oriented structural design by hash chain in mobile network," Int. J. Comput. Sci. Issues, vol. 9, no. 1, pp. 329-339, Jan. 2012.

[10] Z.-Y. Hu, Y.-W. Liu, X. Hu, and J. Li, "Anonymous micropayments authentication (AMA) in mobile data network," in Proc. 23rd Annu. Joint Conf. IEEE Comput. and Commun. Soc. (INFOCOM), vol. 1, Hong Kong, Mar. 2004, pp. 53.

[11] J.-S. Wang, F.-Y. Yang, and I. Paik, "A novel E-cash payment protocol using trapdoor hash function on smart mobile devices," Int. J. Comput. Sci. Netw. Security, vol. 11, no. 6, pp. 12-19, June 2011.

[12] F.-Y. Yang, "Efficient Trapdoor Hash Function for Digital Signatures," Chaoyang J., vol. 12, pp. 351-357, Sep. 2007.

[13] F.-Y. Yang, "Improvement on a Trapdoor Hash Function," Int. J. Netw. Security, vol. 9, no. 1, pp. 17-21, Jul. 2009.

[14] C.-C. Chang, S.-C. Chang, and J.-S. Lee, "An on-line electronic check system with mutual authentication," Comput. and Elect. Eng., vol. 35, no. 5, pp. 757-763, 2009.

[15] D. Chaum, B. Den Boer, E. Van Heyst, S. Mjølsnes, and A. Steenbeek, "Efficient offline electronic checks," in Proc. Workshop Theory Appl. Cryptograph. Tech. Adv. Cryptol. (EUROCRYPT), Aarhus, Denmark, 1990, pp. 294-301.

[16] W. Chen, "Efficient on-line electronic checks," Appl. Math. Comput., vol. 162, no. 3, pp. 1259-1263, Mar. 2005.

[17] H.-T. Liaw, J.-F. Lin, and W.-C. Wu, "A new electronic traveler's check scheme based on one-way hash function," Electron. Commerce Res. Appl., vol. 6, no. 4, pp. 499-508, 2007.

[18] T. Dahlberg, and N. Mallat, and J. Ondrus, and A. Zmijewska, "Past, Present and Future of Mobile Payments Research: A Literature Review," Electron. Commerce Res. Appl., vol 7, no. 2, pp. 165-181, Jul. 2008.

[19] S. Nakamoto, "Bitcoin: A peer-to-peer electronic system," [Online]. Available: `http://bitcoin.org/bitcoin.pdf`.

[20] Y. Ishimaru, W. Sun, K. Yasumoto and M. Ito, "DTN-based Delivery of Word-of-Mouth Information with Priority and Deadline," in Proc. 5th Int. Conf. Mobile Comput. Ubiquitous Netw., 2010, pp. 179-185.

[21] P. Lin, H.-Y. Chen, Y. Fang, J.-Y. Jeng, and F.-S. Lu, "A secure mobile electronic payment architecture platform for wireless mobile networks," IEEE Trans. Wireless Commun., vol. 7, no. 7, pp. 2705-2713, Jul. 2008.

[22] H. Tewari, D. O'Mahony, and M. Peirce, "Reusable off-line electronic cash using secret splitting," Dept. Comput. Sci., Trinity College, at Dublin, Dublin, Ireland, Tech. Rep. TCD-CS-1998-27, Dec. 1998.

[23] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," ACM Trans. Program. Lang. Syst., vol. 4, no 3, pp. 382-401, Jul. 1982.

[24] M. Mitzenmacher, "Compressed Bloom Filters," IEEE/ACM Trans. on Netw., vol. 10, no 5, pp. 604-612, Oct. 2002.

[25] R. C. Merkle, "A Digital Signature Based on a Conventional Encryption Function," in Advances in Cryptology — CRYPTO '87., Lecture Notes in Computer Science, vol. 293, Berlin, Germany: Springer, 1988, pp. 369-378.

[26] D. Chaum, "Blind signatures for untraceable payments," in Proc. of Adv. Cryptol. (Crypto), Santa Barbara, CA, USA, pp. 1983, 199-203.

[27] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," Wireless Commun. Mobile Comput., vol.2, no. 5, pp. 483-502, 2002.

[28] Simulation Data. Accessed: Mar. 8, 2017. [Online]. Available: `https://goo.gl/TLpbSX`

[29] A. Geetha, and N. Sreenath, "Byzantine Attacks and its Security Measures in Mobile Adhoc Networks," (IJCCIE 2016), Int'l Journal of Computing, Communications and Instrumentation Engineering, Vol. 3, Issue 1, 2016.

[30] K. Harshavardhan, "A Survey on Security Issues in Ad Hoc Routing Protocols and their Mitigation Techniques," International Journal of Advanced Networking and Application, Vol. 03, Issue 05, pp. 1338-1351, March-April, 2012.

[31] D. Ali, K. Seyed, and K. Esmaeil, "Security Challenges in Mobile Ad hoc Networks: A Survey," (IJCSES 2015) International Journal of Computer Science and Engineering, Vol. 6, No. 1, February, (2015).

[32] S. Mojtaba, G. Imran, H. Aida, and J. Seung, "Routing Attacks in Mobile Adhoc Networks: An Overview," Science International (Lahore), Vol. 25, No. 4, pp. 1031–1034, 2013.

[33] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato, and A. Jamalipour, "A survey of routing attacks in mobile ad hoc networks," in IEEE Wireless Communications, Vol. 14, no. 5, pp. 85–91, October 2007.

[34] R. H. Jhaveri, S. J. Patel, and D. C. Jinwala, "DoS Attacks in Mobile Ad Hoc Networks: A Survey," in Proceedings of 2012 Second International Conference on Advanced Computing and Communication Technologies, Rohtak, Haryana, 2012, pp. 535-541.

[35] M. G. Zapata, and N. Asokan, "Securing ad hoc routing protocols," in Proceedings of the 1st ACM workshop on Wireless security (WiSE '02). ACM, New York, NY, USA, 1–10, 2002.

[36] M. Alajeely, A. Ahmad, and R. Doss, "Malicious Node Detection in OppNets using Hash Chain Technique," 4th International Conference on Computer Science and Network Technology (ICCSNT 2015), Vol. 1. IEEE, 2015.

[37] A. Baadache, and A. Belmehdi, "Fighting against packet dropping misbehavior in multi-hop wireless ad hoc networks," Journal of Netw. Comput. Appl. Vol. 35, No.3, May 2012, pp. 1130-1139.

[38] P. Papadimitratos, and Z. J. Haas, "Secure link state routing for mobile ad hoc networks," in Proceedings of the IEEE Workshop on Security and Assurance in Ad hoc Networks, in conjunction with the 2003 Symposium on Applications and the Internet, pp. 379–383, January, 2003.

[39] P. Papadimitratos, and Z. J. Haas, "Secure data transmission in mobile ad hoc networks," in Proceedings of the 2nd ACM workshop on Wireless security (WiSe '03), ACM, New York, NY, USA, 41–50, 2003.

[40] M. Z. A. Bhuiyan, and J. Wu, "Collusion Attack Detection in Networked Systems," in Proceedings of 2016 IEEE 14th Intl. Conf. on Dependable, Autonomic and Secure Computing, 14th Intl. Conf. on Pervasive Intelligence and Computing, 2nd Intl. Conf. on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech), Auckland, 2016, pp. 286-293.

[41] O. Bonaventure, "Computer Networking : Principles, Protocols and Practice," 2011, pp. 41-45. `http://inl.info.ucl.ac.be/cnp3`

[42] D. Johnson, A. Menezes, and S. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)," International Journal of Information Security, Vol. 1, No. 1, pp. 36 -63, August 2001.

[43] A. Al Imem, "Comparison and evaluation of digital signature schemes employed in NDN network," in International Journal of Embedded system and Applications (IJESA), Vol. 5, No. 2, June 2015.

Publication List

Journal

J1. Babatunde Ojetunde, Naoki Shibata, and Juntao Gao, "Monitoring-Based Method for Securing Link State Routing against Byzantine Attacks in Wireless Networks," *Journal of Information Processing*, Vol. 26, pp. 98-110, 2018. DOI: 10.2197/ipsjjip.26.98 [Corresponds to Chapter 3]

J2. Babatunde Ojetunde, Naoki Shibata, and Juntao Gao, "Secure Payment System Utilizing MANET for Disaster Areas," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1-13, September 2017. DOI: 10.1109/TSMC.2017.2752203 [Corresponds to Chapter 2]

International Conferences (Peer Review)

I1. Babatunde Ojetunde, Naoki Shibata, and Juntao Gao, "Securing Link State Routing for Wireless Networks against Byzantine Attacks: A Monitoring Approach," in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, pp. 596-601, Turin, Italy, Jul. 2017. [Corresponds to Chapter 3]

I2. Babatunde Ojetunde, Naoki Shibata, and Juntao Gao, "An Endorsement-Based Mobile Payment System for a Disaster Area," in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 482-489, Gwangiu, South Korea, Mar. 2015. [Corresponds to Chapter 2]

Domestic Conferences (Non Peer Review)

D1. Babatunde Ojetunde, Naoki Shibata, and Juntao Gao, "Ideas for Realizing Secure Low-latency Anonymity Network using Network Coding," in *2017 25th Multimedia Communication and Distributed Processing Workshop (DPSWS)*, vol. 2017, pp. 203-207, Hokkaido, Japan, Oct. 2017.

D2. Babatunde Ojetunde, Naoki Shibata, and Juntao Gao, "A Proposed Monitoring Scheme to Prevent Byzantine Attacks on Link State Routing in MANETs," in *IEICE technical report*, vol. 171, no. 71, Vol.2017-DPS-171No.31, pp. 217-224, Okinawa, Japan, Jun. 2017. [Corresponds to Chapter 3]

D3. Babatunde Ojetunde, Naoki Shibata, Juntao Gao, and Minoru Ito, "Consideration on Monitoring Scheme to Secure Link State Routing against Byzantine Attacks," in *2016 24th Multimedia Communication and Distributed Processing Workshop (DPSWS)*, vol. 2016, pp. 214 - 220, Akita, Japan, Oct. 2016. [Corresponds to Chapter 3]

D4. Babatunde Ojetunde, Naoki Shibata, Juntao Gao, and Minoru Ito, "An Enhanced Endorsement Chain using Endorsement Delegation on MANETs Based Mobile Payment System in a Disaster Area," in *SIG technical report (2016-DPS-166)*, vol. 166, no. 11, pp. 1-10, Tokyo, Japan, Mar. 2016. [Corresponds to Chapter 2]

D5. Babatunde Ojetunde, Naoki Shibata, Juntao Gao, and Minoru Ito, "Simulation-Based Evaluation of a Mobile Payment System Utilizing MANETs for a Disaster Area," in *2015 Multimedia, Distributed, Cooperative and Mobile (DICOMO)* , pp. 757-766, Iwate, Japan, Jul. 2015. [Corresponds to Chapter 2]