2012

# ISSUES AND SOLUTIONS OF APPLYING IDENTITY-BASED CRYPTOGRAPHY TO MOBILE AD-HOC NETWORKS

Shushan Zhao
*University of Windsor*

# ISSUES AND SOLUTIONS OF APPLYING IDENTITY-BASED CRYPTOGRAPHY TO MOBILE AD-HOC NETWORKS

by

**Shushan Zhao**

A Dissertation
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy at the
University of Windsor

Windsor, Ontario, Canada
2011

Issues and Solutions of Applying Identity-based Cryptography to Mobile Ad-hoc Networks

by

Shushan Zhao

APROVED BY:

---

Dr. I. Stojmenovic, External Examiner
University of Ottawa

---

Dr. H. Wu
Department of Electrical & Computer Eng.

---

Dr. A. Jaekel
School of Computer Science

---

Dr. Z. Kobti
School of Computer Science

---

Dr. A. Aggarwal, Advisor
School of Computer Science

---

Dr. R. Kent, Co-Advisor
School of Computer Science

---

Dr. M. Hlynka, Chair of Defense
Department of Mathematics & Statistics

09 December 2011

# Declaration of Originality

**I. Co-Authorship Declaration**

I hereby declare that this thesis incorporates material that is the result of joint research undertaken by me under the supervision of Dr R. D. Kent and Dr. A. K. Aggarwal. The collaboration is covered in Chapter 2, 3, 4, 5, 6, 7 of the thesis. In all cases, the key ideas, primary contributions, experimental designs, data analysis and interpretation, were performed by the author, and the contribution of coauthors was primarily through the provision of valuable suggestions and helping in comprehensive analysis of the experimental results submitted for publication.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my dissertation, and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis.

I certify that, with the above qualification, this dissertation, and the research to which it refers, is the product of my own work.

**II. Declaration of Previous Publication**

This thesis includes eight original papers that have been previously published/submitted for publication in peer reviewed journals/conferences, as in following table.

I certify that I have obtained written permissions from the copyright owners to include the above published materials in my dissertation. I certify that the above materials describe work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or

| Thesis Chapter | Publication title/full citation | Publication status |
|---|---|---|
| Chapter 2 | Shushan Zhao, Akshai Aggarwal and Robert D. Kent "A Framework for Revocation of Proxy Certificates in a Grid", In Proc. 8th International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (2007), IEEE, pp. 532–537 | Published. ©2007 IEEE. |
| Chapter 2 | Shushan Zhao, Akshai Aggarwal and Robert D. Kent "PKI-Based Authentication Mechanisms in Grid Systems", In Proc. International Conterence on Networking, Architecture, and Storage (2007), IEEE, pp. 83–90 | Published. ©2007 IEEE. |
| Chapter 3 | Shushan Zhao, Akshai Aggarwal, Richard Frost and Xiaole Bai "A Survey of Applications of Identity-Based Cryptography in Mobile Ad-Hoc Networks", J. Communications Surveys & Tutorials, Volume PP Issue 99, March 2011, IEEE, pp. 1–21 | Published. ©2011 IEEE. |
| Chapters 4,5 | Shushan Zhao and Akshai Aggarwal "PAPA-UIC: A Design Approach and a Framework for Secure Mobile Ad-hoc Metworks", J. Security and Communication Networks, Special Issue: Security in Ad Hoc Networks and Pervasive Computing, Volume 3 Issue 5, September-October 2010, John Wiley and Sons, pp. 371–383 | Published. ©2010 John Wiley & Sons. |
| Chapters 4,5 | Shushan Zhao, Akshai Aggarwal, Shuping Liu and Huapeng Wu "A Secure Routing Protocol in Proactive Security Approach for Mobile Ad-hoc Networks", In Proc. Wireless Communications & Networking Conference (2008), IEEE, pp. 2627–2632 | Published. ©2008 IEEE. |
| Chapters 5 | Shushan Zhao and Akshai Aggarwal "Against mobile attacks in Mobile Ad-hoc Networks", In Proc. Information Theory and Information Security (2010), IEEE, pp. 499–502 | Published. ©2010 IEEE. |
| Chapters 5 | Shushan Zhao, Daniel Jaskiewicz and Jouni Karvo "A Deployment Tool for Public Safety Ad-hoc Networks", In Proc. 1st Communication System Software and Middleware (2006), IEEE, pp. 1–6 | Published. ©2006 IEEE. |
| Chapters 7 | Shushan Zhao and Akshai Aggarwal "General-purpose Identity Hiding Schemes for Ad-hoc Networks", In Proc. International Symposium on Intelligent Ubiquitous Computing (2009), IEEE, pp. 349–352 | Published. ©2009 IEEE. |

otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

# Abstract

Concept of Mobile Ad-hoc Networks (MANETs) was brought up a few decades ago with assumed prosperous future. Unfortunately, we do not see many practical applications of them in real life. Security of MANETs is a big concern considered by investors and industries, and hinders them from putting MANETs into application. Requirements of security, and difficulties to meet these requirements have been stated clearly already; yet solutions to these difficulties are not quite clear. Cryptographic technologies seem to be capable of satisfying most of the requirements, which has been proved in Internet or wired networks. However, most of the technologies, including symmetric and traditional asymmetric cryptography (such as Public Key Infrastructure (PKI)), are inapplicable or inconvenient to use in MANETs context. Identity-based Cryptography (IBC), as a special form of asymmetric cryptography, carries many features interesting for MANETs. IBC has been studied a lot recently by researchers of MANET security, and many applications have been proposed and claimed to address this difficult problem. However, it is still the case that most of the solutions are not sound enough to be used in a practical MANET.

This thesis starts with an intensive survey on the proposals of applications of IBC in MANETs, and points out the issues, limitations and weaknesses in these proposals and also in IBC itself. The thesis proposes a novel framework with key management and secure routing scheme integrated aiming to address these issues. This scheme brings these contributions: compared to symmetric key solutions, it has more functionality derived from asymmetric keys, and is more secure due to using 1-to-$m$ broadcasting key instead of only 1 group broadcasting key, and has less keys to store per node due to using asymmetric keys instead of pairwise symmetric keys; compared to traditional asymmetric cryptography solutions, the storage and communication requirements are lower due to IBC properties; compared to previous IBC solutions, it has no key management and secure routing interdependency cycle problem. Security of the proposed scheme is proved

and performance of the scheme is simulated and analyzed in the thesis. To the end of a complete solution for an arbitrary MANET running in an arbitrary environment, the thesis proposes enhancements to counter various attacks and options to abate or eliminate limitations and weaknesses of IBC. The proposed scheme has a wide range of applicability for various MANETs with little or no administrative overhead depending on situations where it is considered.

# Acknowledgements

I would like to express my deepest gratitude to my supervisors, Dr. R. D. Kent and Dr. A. K. Aggarwal, for providing some original ideas, inspiring discussions and valuable comments.

I would also thank Dr. Richard Frost for instructions on conducting the survey in the thesis work, and thank my thesis committee members, Dr. Jaekel, Dr. Kobti, Dr. Wu, and external reader, Dr. Stojmenovic, for providing me constructive suggestions and comments.

I was partly sponsored by GA/TA-ship of University of Windsor, and OGSST scholarship of the Ministry of Colleges and Universities, Ontario, during my studies and research. I appreciate this indispensable support.

I would also express my gratitude, from bottom of my heart, to my parents and family for their understanding, love and support during my PhD studies.

Finally, my sincere thanks go to all people who have supported, helped, and inspired me.

# Contents

# Abbreviations and Acronyms

| | |
|---|---|
| AODV | Ad hoc On-Demand Distance Vector |
| CBC | Certificate-based Cryptography |
| DSR | Dynamic Source Routing |
| ETSI | European Telecommunications Standards Institute |
| GloMo | Global Mobileinformation Systems |
| GPS | Global Positioning System |
| IBC | Identity-based Cryptography |
| IETF | Internet Engineering Task Force |
| JTRS | Joint Tactical Radio System |
| KM | Key Management |
| MANET | Mobile Ad-hoc Network |
| NTDR | Near-Term Digital Radio |
| OLSR | Optimized Link State Routing protocol |
| PBC | Pairing-based Cryptography |
| PKG | Private Key Generator |
| PKI | Public Key Infrastructure |
| PRNET | Packet Radio Network |
| RSRE | Royal Signal and Radar Establishment |
| SR | Secure Routing |
| SURAN | Survivable Radio Networks |
| TA | Trusted Authority |
| TI | Tactical Internet |
| TIA | Telecommunications Industry Association |
| TTP | Trusted Third Party |
| WIDENS | Wireless Deployable Network System |

# Table of Notations

| Symbols | Meanings |
|---|---|
| $\mathbb{Z}$ | set of integers |
| $\mathbb{Z}_n$ | set of integers mod $n$ |
| $\mathbb{F}_q$ | the finite field with $q$ elements |
| $\mathbb{Z}_q^*$ | the multiplicative group of integers modulo prime number $q$. $\mathbb{Z}_q^* = \{a \vert 1 \leq a \leq q-1\}$ |
| $E/\mathbb{F}_p$ | elliptic curve over $\mathbb{F}_p$ |
| $\mathbb{G}_1$ | subgroup of the additive group of points of $E/\mathbb{F}_p$ |
| $\mathbb{G}_2$ | subgroup of the multiplicative group of the finite field $\mathbb{F}_{p^2}^*$ |
| $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ | a bilinear map between two cyclic groups $\mathbb{G}_1, \mathbb{G}_2$ |
| $P$ | an arbitrary point in $E/\mathbb{F}_p$ |
| $d_{ID}$ | private key of $ID$ |
| $Q_{ID}$ | public key of $ID$ |
| $s$ | master secret key |
| $P_{pub}$ | system public key |
| $H_{(i)}$ | a hash function. When multiple hash functions are used in a system, an integer $i$ is used as subscript. |

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Motivation

Mobile Ad-hoc Networks (MANETs) (the term Wireless Ad-hoc Networks is used in the literature interchangeably) have been an active research topic for several decades. The first stage of research was concentrated on efficient formation of an ad-hoc network, i.e. routing setup. Then many researchers realized that without assurance of security, formation of a network is meaningless—the network can be easily broken or taken over by an adversary. During the last two decades, security of MANETs has gained more and more attention.

Cryptography is a solution that can meet most of the security requirements. More specifically, cryptographic solutions can be classified into two categories—Symmetric Key Cryptography and Asymmetric Key Cryptography. The former has limited functionality and cannot provide a complete solution by itself. Among the latter, Public Key Infrastructure (PKI) is a most poplar solution in wired networks. Unfortunately, in MANETs, there are many difficulties or barriers that impede application of PKI to MANETs. About 10 years ago, Identity-based Cryptography (IBC) emerged as a new cryptographic technology. As a special and simplified form of asymmetric cryptography, it has many advantages to MANETs, and has aroused much research interest. Most of recent development on MANET security is related to IBC. There have been a large number of proposals using IBC for MANET security.

However, after an intensive study and survey, we noticed and identified some issues on applying IBC to MANETs. The main issues pertain to Key Management (KM) and Secure Routing (SR). Both of these components are essential to a security scheme. Unfortunately,

previous studies in the literature seem to treat these two components separately.

Indeed, they are interdependent on each other. You cannot generate either of the components following these schemes if you do not have the other one already, and if you have two of them from different schemes, you cannot couple them together in a system. Additionally, these schemes are subject to many attacks due to loose coherency between these two components. The issues we found motivated us to launch this research to find a solution.

## 1.2 Research Objectives

The main objective of this research is to find a solution that addresses the identified issues and is better than previous ones. Specifically, we aim to propose a novel key management and secure routing framework that can be applied to the design of a practical MANET without the issues we have identified. On the one hand, this framework should have advantages of IBC which has already been accepted as a prospective solution for MANET security. On the other hand, this solution should address the issues we have identified above and known issues of IBC schemes published in the literature. The efficiency and performance of the framework in other aspects should not degrade compared to previous ones. The framework should be scalable to practical size of MANETs. The framework should be feasible and applicable to practical MANETs. The framework should be extensible to accommodate specific requirements of various customers and in various scenarios.

## 1.3 Contributions and Applicability

This thesis studies MANET security requirements and solutions. Concentrated on IBC solutions, the thesis points out issues of applying this latest and most promising technology to MANET security. In light of the discovered issues, a novel framework for MANET security is proposed in this thesis. The proposed scheme addresses key management and secure routing interdependency cycle problem of previous IBC schemes. This scheme brings these contributions: compared to symmetric key solutions, it has more functionality derived from asymmetric keys, and is more secure due to using 1-to-$m$ broadcasting key instead of only 1 group broadcasting key, and has less keys to store per node due to using asymmetric keys instead of pairwise symmetric keys; compared to traditional asymmetric cryptography (PKI) solutions, the storage and communication requirements

are lower due to IBC properties; compared to previous IBC solutions, it has no KM-SR interdependency cycle problem, and is immune to insider attacks and mobile attacks and many other routing attacks.

The result of this work presents a feasible security solution to a wide range of MANETs where there is an administrator that generates and distributes initial system parameters to all nodes, and the administrator can authenticate the identity of a node and assign the initial private key to it. Basically this includes all MANETs where IBC is applicable, with an extra requirement—a controlled deployment phase. Examples of this type of MANETs include, but are not limited to: sensor networks, wearable computer systems in military, public safety networks, and emergency and disaster rescue teams. This scheme seems to be the best security solution to these networks so far.

## 1.4 Thesis Outline

The rest of the thesis is organized as follows: Chapter 2 presents a background study related to the research work presented in this thesis. Chapter 3 presents issues we have identified when applying IBC to MANET security, and limitations and weaknesses of IBC itself. Chapter 4 proposes a novel key management and secure routing integrated framework to address issues of applying IBC to MANETs. Chapter 5 analyzes security features of the framework with mathematical proof, and presents enhancements to counter various attacks. Chapter 6 presents information about the simulation environment, simulation results, and related discussions. Chapter 7 presents solutions to limitations and weaknesses of IBC itself applicable to the framework and other IBC schemes. Conclusion and future work are given in Chapter 8.

# Chapter 2

# Background

This chapter presents a background study related to the research work presented in this thesis. Section 2.1 presents an overview of Mobile Ad-hoc Networks. Section 2.2 summarizes security challenges and requirements, and cryptographic solutions on high level. Section 2.3 introduces Identity-based cryptography.

## 2.1 An Overview of Mobile Ad-hoc Networks

There has been a surge of interest in ad hoc networks in recent decades. An ad-hoc (or "spontaneous") network is a local area network or other small network, especially one with wireless or temporary plug-in connections, in which some of the network devices are part of the network only for the duration of a communications session or, in the case of mobile or portable devices, while in some close proximity to the rest of the network. A Mobile Ad-hoc Network (MANET) is defined as an autonomous system of mobile routers (and associated hosts) connected by wireless links, the union of which form an arbitrary graph [25]. A MANET comprises a collection of two or more devices equipped with wireless communication and networking capability. Such devices can communicate with another node that is immediately within their radio range or one that is outside their radio range, intermediate nodes forwarding or relaying packets in the latter scenario [84, 37]. This kind of network is very similar to cellular networks, but support from base stations is not necessarily required. Actually, a MANET can be an extension or a redundant backup of cellular networks. The allure of providing anytime, anywhere services without infrastructure makes such networks very attractive.

The development and exploitation of ad hoc wireless communication has been started

since 1970's. Earlier projects on ad hoc wireless communication include Packet Radio Network (PRNET) program, Survivable Radio Networks (SURAN), and Royal Signal and Radar Establishment (RSRE)

The origin and early development of MANETs were attributed to the needs of battlefield communication. In 1994 Defense Advanced Research Projects Agency (DARPA) initiated Global Mobile (GloMo) [57] Information Program which recently concluded. The flat (peer-to-peer) and hierarchical network architectures were studied. The hierarchical architecture uses modular system of link and network layer algorithms to support distributed, real time multimedia applications in MANETs. It has three components: clustering techniques, location management and virtual circuit setup and repair.

The generation of adaptive, multi-band multi-mode radios—the Joint Tactical Radio System (JTRS) offers improved flexibility over half-duplex, single-channel radios at higher layers of the system because of the ability to transmit and receive on different bands and using different waveforms [42]. The Near-Term Digital Radio (NTDR) [77] program of DARPA benefited from the results of the GloMo program and implemented many of the technologies developed during the SURAN program.

At present, one particularly active application of MANETs is interconnection of sensors in industrial, commercial, or military settings. Sensors are typically small wireless devices measuring environmental inputs and transmitting them to control centres [3]. There are many live projects going on in this area. For example, the Berkeley Wireless AC Meter/Switch (ACme) Project. The goal of this project is to enable wireless energy/power measurement and control of AC devices. This device fills the gap between inexpensive LCD watt-meters (e.g. Kill-A-Watt) and expensive networked enterprise energy monitors. ACme uses the ADE7753 energy monitor chip for energy and power measurements, the SHARP solid-state relay for power switching, and the Berkeley EPIC wireless module for communication [50].

Another application is that of emergency response and rescue. MANETs are well suited for such applications because of their ability to create connectivity rapidly when the existing communication infrastructure has been destroyed. One example of this application is the European Project entitled WIreless DEployable Network System (WIDENS) launched in 2005 which aims to offer a common communication channel through a wireless ad hoc network to all actors in an emergency situation in the field of operation at the time of intervention, for each organization and across organizations [98].

Other prospective applications of MANETs include Vehicular Ad-Hoc Networks (VANETs) where vehicles can share up-to-date traffic information on the fly, and Mesh Net-

works where end users are connected to each other via broadband channels based on multiple connections.

In the academic community, the MANET chartered Working Group was established in 1997 within Internet Engineering Task Force (IETF). The MANET activities are focused on studying routing specification with the goal of supporting network scaling up to hundreds of routers [19].

## 2.2 Security of Mobile Ad-hoc Networks

Research on security of MANETs remains active, despite years of exploration, in both academia and industry. This is partially due to the fact that no mature solution is widely accepted and also to the growing availability of small, personalized mobile devices with peer to peer communication capability through wireless channels.

General security requirements for MANETs include [1]:

- *Data Confidentiality* that keeps data secret to outsiders,
- *Data Integrity* that prevents data from being altered,
- *Data Freshness* that keeps data in the correct order and up-to-date,
- *Data Availability* that ensures data to be available on request,
- *Data & Identity Authentication* that verifies that the data or request came from a specific, valid sender,
- *Non-repudiation* that ensures a node cannot deny sending a message.

Security mechanisms that are widely used and proven to be effective in wired networks are not always applicable to MANETs. Attacks that can be effectively detected and prevented in wired networks have been big security challenges in MANETs. Examples include, but are not limited to, identity/address spoofing, message tampering and forgery, message replay, etc. Compared to wired networks, the combination of the following characteristics of MANETs makes it especially difficult to achieve security requirements:

- Lack of a network infrastructure and online administration.
- Network topology and node membership dynamics.
- The potential for insider attacks.

- Computing and communication capacity constrained resources.
- Wireless link vulnerabilities.

Security proposals in early research are typically attack-oriented. They often first identify several security threats and then enhance the existing protocol or propose a new protocol to thwart them. Such solutions are designed explicitly against limited attack models. They work well in the presence of designated attacks but may collapse under combined or unanticipated attacks [89].

Cryptography is then used to support a general design framework. Cryptographic solutions can satisfy the above requirements except *Data Availability* which requires assistance of other technologies. Cryptography techniques used in MANETs can be classified into two categories, namely, *symmetric key based* and *asymmetric key based*. In symmetric key based schemes, if an attacker compromises the symmetric key of a group of users, then all encrypted messages for that group will be exposed. Asymmetric key based schemes can provide more functionalities than symmetric ones. For example, key distribution is much easier, authentication and non-repudiation are available, and compromise of a private key of a user does not reveal messages encrypted for other users in the group. However, asymmetric key based schemes are generally more expensive computationally.

Traditional asymmetric cryptography is used widely and effectively in the Internet; it relies on a Public Key Infrastructure (PKI), and can be called Certificate-based Cryptography (CBC), in contrast to identity-based cryptography. The success of PKI depends on the availability and security of a Certificate Authority (CA), a central control point that everyone trusts. With PKI, an entity has a pair of private key and public key. The private key is bound to its public key that is signed by the CA with CA's public key. The public key and corresponding signature of the CA are presented in a public key certificate (PKC). In communication, the recipient needs to know the PKC of the sender and the public key of the CA, in order to authenticate the sender and verify the message. PKCs can be stored in the recipient in advance, or retrieved on-the-fly from CA or centralized certificate repository. However, in general MANETs, applying PKIs by maintaining a central control point for CA or certificate repository is clearly not always feasible. Another obstacle that impedes PKI's employment in MANETs is the heavy overhead of transmission and storage of PKCs.

## 2.3 Identity-based Cryptography

Identity-based cryptography (IBC) is a special form of public key cryptography. It is an approach that seeks to eliminate the requirement of a CA and PKCs. Since 2001, IBC has attracted increasing attention from security researchers. Some properties of IBC make it especially suitable for MANETs. Fang *et al* [35, 95] summarize the advantages of IBC to MANETs:

- Easier to deploy without any infrastructure requirement. This saves certificate distribution, while bringing gratuitous pairwise keys without any interaction between nodes.
- Its resource requirements, regarding process power, storage space, and communication bandwidth, are much lower.
- The public key of IBC is self-proving and can carry much useful information.

We believe that IBC, with its rapid development in recent years, is a promising solution for MANET security problem.

### 2.3.1 A Brief History of Identity-based Cryptography

IBC is in the category of *asymmetric key based* cryptography. It specifies a cryptosystem in which both public and private keys are based on the identities of the users. The idea of IBC was first proposed by Shamir [81] in 1984. Such a scheme has the property that a user's public key is an easily calculated function of his identity, while a user's private key can be calculated for him by a trusted authority, called a Private Key Generator (PKG). The identity-based public key cryptosystem can be an alternative for certificate-based PKI, especially when efficient key management and moderate security are required. Compared to traditional PKI, it saves storage and transmission of public keys and certificates, which is especially attractive for devices forming MANETs. Thus, application of IBC to MANETs is an important research topic in areas of both cryptography and MANETs.

For a long time after Shamir published his idea, the development of IBC was very slow. Joux [51], in 2000, showed that Weil pairing can be used for "good" by using it in a protocol to construct three-party one-round Diffie-Hellman key agreement. This was one of the breakthroughs in key agreement protocols. After this, Boneh and Franklin [10] presented at Crypto 2001 an identity-based encryption scheme based on properties of bilinear pairings on elliptic curves, which is the first fully functional, efficient and provably secure

identity-based encryption scheme. In Asiacrypt 2001, Boneh, Lynn and Shacham proposed a basic signature scheme using pairing, the BLS scheme [13], that has the shortest length among signature schemes in classical cryptography.

Subsequently, a number of cryptographic schemes based on the work of Boneh *et al* [10] and [13] were proposed. This type of identity-based cryptography is also named Pairing-based Cryptography (PBC). There are also a few IBC schemes using other approaches, for instance, Cocks' scheme is based on the quadratic residuosity problem [24]. Most proposals for MANET security in the literature use PBC.

### 2.3.2 Preliminaries of Identity-based Cryptography

In [81], Shamir introduces a novel type of cryptographic scheme, the so-called identity-based cryptosystem, which enables any pair of users to communicate securely and to verify each other's signatures without exchanging private or public keys, without keeping key directories, and without using the services of a third party.

Shamir states that "The scheme is based on a public key cryptosystem with an extra twist: instead of generating a random pair of public/secret keys and publishing one of these keys, the user chooses his name and network address as his public key. Any combination of name, social security number, street address, office number or telephone number can be used provided that it uniquely identifies the user in a way he cannot later deny, and that it is readily available to the other party. The corresponding secret key is computed by a PKG and issued to the user when he first joins the network." Figure 2.1 illustrates his idea: In an identity-based cryptosystem, the recipient's identity $i$ is used to generate the encryption key, and the decryption key is derived from $i$ and a random seed $k$. In an identity-based signature scheme, the signature key is generated from sender identity $i$ and a random seed $k$, and the verification key is derived from sender's identity $i$.

In his paper, Shamir specifies the requirements of an implementation of such a scheme and lists the implementation principles:

- The choice of keys is based on a truly random seed $k$. When the seed $k$ is known, secret keys can be easily computed for a non-negligible fraction of the possible public keys.
- The problem of computing the seed $k$ from specific public/secret key pairs generated with this $k$ is intractable.

Based on these requirements, he states that the RSA scheme is not capable of supporting

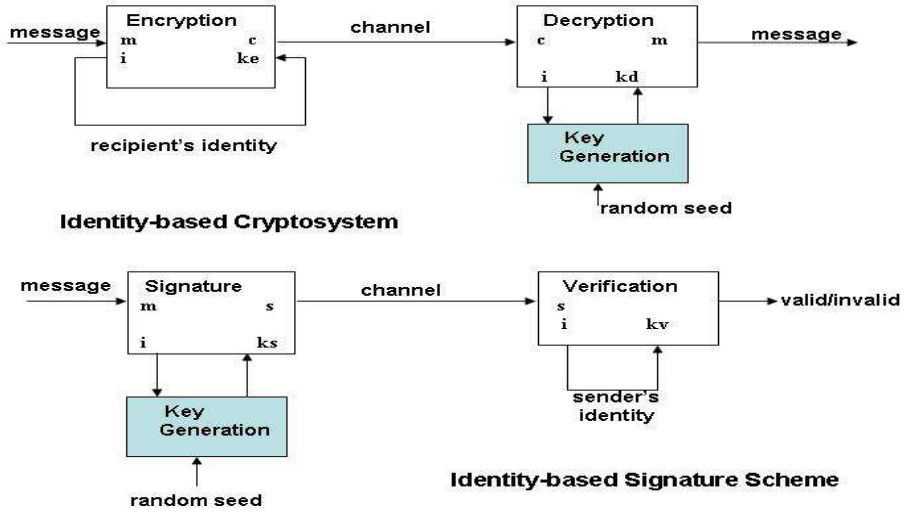



Figure 2.1: Shamir's Identity-based Cryptosystem and Signature Scheme ([81, p. 52])

his scheme.

He states that at that stage they have concrete implementation proposals only for identity-based signature schemes, but conjectures that such cryptosystems exist and encourage the readers to look for such systems.

Currently, most IBC schemes, and all PBC schemes, are based on assumptions of hard **Diffie-Hellman (DH)** problems [1] in elliptic curves. The most frequently used assumptions are summarized below: [34, p. 7] (Refer to Table of Notations for notations and explanations. Unless otherwise stated, we use the same notations throughout the thesis.)

- **Computational Diffie-Hellman (CDH) problem** in $\mathbb{G}_1$: there is no efficient algorithm to compute $\hat{e}(P,P)^{ab}$ from $P, aP, bP \in \mathbb{G}_1$ where $a, b \in \mathbb{Z}_q^*$. [2]

- **Weak Diffie-Hellman (WDH) problem** and **Static Diffie-Hellman (SDH) problem** in $\mathbb{G}_1$: there is no efficient algorithm to compute $sQ$ from $P, Q, sP$, where

[1] A general **Diffie-Hellman (DH) problem** is to calculate $g^{xy}$ from $g^x$ and $g^y$ in a group.

[2] The general form of a bilinear map is denoted $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$, where $\mathbb{G}_1$ and $\mathbb{G}_3$ are cyclic, and $\mathbb{G}_2$ is not necessarily cyclic.A *Symmetric Bilinear Map* is denoted $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ between two cyclic groups $\mathbb{G}_1, \mathbb{G}_2$ of order $q$ for some large prime $q$, where $\mathbb{G}_1$ is the group of points of an elliptic curve over $\mathbb{F}_p$ and $\mathbb{G}_2$ is a subgroup of $\mathbb{F}_{p^2}^*$

$P, Q \in \mathbb{G}_1$ and $s \in \mathbb{Z}_q^*$.

- **Bilinear Diffie-Hellman (BDH) problem** in $(\mathbb{G}_1, \mathbb{G}_2, \hat{e})$: there is no efficient algorithm to compute $\hat{e}(P, P)^{abc} \in \mathbb{G}_2$ from $P, aP, bP, cP \in \mathbb{G}_1$ where $a, b, c \in \mathbb{Z}_q^*$.

- **Decisional Bilinear Diffie-Hellman (DBDH) problem** in $(\mathbb{G}_1, \mathbb{G}_2, \hat{e})$: there is no efficient algorithm to decide if $r = \hat{e}(P, P)^{abc}$ given $r \in \mathbb{G}_2$ and $a, b, c \in \mathbb{Z}_q^*$

Boneh and Franklin's scheme, published in [10], is the first fully functional IBC scheme. The paper refers to Shamir's idea of the Identity-based Encryption (IBE) scheme [81], and several proposals for IBE schemes such as [30, 83, 85, 64]. They consider none of them to be fully satisfactory due to unrealistic requirements, such as users not colluding, the long time required for private key generation, and tamper-resistant hardware.

Security of their system is based on the BDH problem, an analogue of the computational Diffie-Hellman assumption on elliptic curves. They build the IBE system from a symmetric bilinear map and use the Weil pairing on elliptic curves as an example of such a map.

A cryptographic bilinear map satisfies the following properties [34, p. 6]:

1. **Bilinear**: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and all $a, b \in \mathbb{Z}_q^*$. This can be restated in the following way. For $P, Q, R \in \mathbb{G}_1, \hat{e}(P + Q, R) = \hat{e}(P, R)\hat{e}(Q, R)$ and $\hat{e}(P, Q + R) = \hat{e}(P, Q)\hat{e}(P, R)$.

2. **Non-degenerate**: $\hat{e}(P, P) \in \mathbb{F}_{p^2}^*$ is an element of order $q$, and in fact a generator of $\mathbb{G}_2$. In other words, $\hat{e}(P, P) \neq 1$

3. **Computable**: Given $P, Q \in \mathbb{G}_1$ there is an efficient algorithm to compute $\hat{e}(P, Q)$.

Their scheme is specified by four randomized algorithms [10, p. 215]:

- Setup: The algorithm maps arbitrary string identities to points on an elliptic curve. Set the system public key $P_{pub}$ as $sP$ where $s$ is a random number in $\mathbb{Z}_q^*$, and $P$ is an arbitrary point in $E/\mathbb{F}_p$ of order $q$. Choose a cryptographic hash function $H : \mathbb{F}_{p^2} \rightarrow \{0, 1\}^n$ for some $n$. Choose a cryptographic hash function $G : \{0, 1\}^* \rightarrow \mathbb{F}_p$. The system parameters are $params = \langle p, n, P, P_{pub}, G, H \rangle$. The master-key is $s \in \mathbb{Z}_q$.

- Extract: For a given string $ID \in \{0, 1\}^*$, the algorithm builds public key for $ID$: $Q_{ID} = G(ID)$, a point in $E/\mathbb{F}_p$ mapped from $ID$, and the private key $d_{ID}$ as $d_{ID} = sQ_{ID}$.

- Encrypt: Choose a random $r \in \mathbb{Z}_q$, and set the ciphertext to be $C = \langle rP, M \oplus H(g_{ID}^r) \rangle$ where $g_{ID} = \hat{e}(Q_{ID}, P_{pub}) \in \mathbb{F}_{p^2}$
- Decrypt: Let $C = \langle U, V \rangle$ be a ciphertext encrypted using the public key of $ID$, decrypt $C$ using the private key $d_{ID}$: $V \oplus H(\hat{e}(d_{ID}, U)) = M$

Further, they analyze the security of their scheme, and state that the scheme has chosen ciphertext security in the random oracle model assuming Weak Diffie-Hellman.

The scheme proposed in their paper is subsequently improved by many other researchers, and widely adopted in many identity-based security schemes.

Following Boneh and Franklin's scheme [10], many PBC schemes have been proposed. Modified Weil Pairing and Tate Pairing are examples of cryptographic bilinear maps. Currently, active research is being carried out to obtain efficient algorithms to compute pairings.

## 2.4 Summary of the Chapter

This chapter presented background of the research topic area of MANETs, research challenges in security of MANETs and some solutions, and finally an introduction to Identity-based Cryptography. The next chapter will present issues of applying IBC to MANET security we have identified during this research.

# Chapter 3

# Issues of Applying IBC to MANETs

In this chapter, we present and discuss issues we found when applying IBC to MANET security. Simply put, there are two issues: key management (KM) and secure routing (SR). Section 3.1 reviews key management schemes in the literature, mainly those for master key, private key and group key generation. Section 3.2 reviews secure routing schemes in the literature. Section 3.3 discusses the issues we found in key management and secure routing which motivated us for this research. Section 3.4 reviews limitations and weaknesses of IBC itself and existing solutions.

## 3.1 Key Management Using IBC

Cryptographic techniques are often at the center of solving security problems in MANETs and hence need key management. Key management in IBC requires key generation and distribution methods, and ideally key protection and revocation. This section reviews and discusses proposals for IBC key management in MANETs.

### 3.1.1 Master Key and Private Key Generation

Most of the master key and private key generation schemes are derived from and are variants of Boneh and Franklin's scheme [10]. The criteria to judge this type of scheme is use of their four primitive algorithms. In this section, we first review some examples based on traditional threshold cryptography of Zhou *et al* [99] and discuss the limitations of these schemes, and then discuss some proposals that attempt to improve traditional threshold cryptography. We also study some key generation schemes tweaked for specific purposes: e.g. high privacy, compromise-tolerance, or light-weight.

### Threshold Cryptography

Many IBC schemes use threshold cryptography which originated from Shamir [80], for their key management. Shamir gives a solution to the problem of sharing a secret among a number of users in [80]. In his paper, he identifies the problem of how to divide data $D$ into $n$ pieces in such a way that $D$ is easily reconstructed from any $t$ pieces, but even complete knowledge of $t-1$ pieces reveals absolutely no information about $D$.

Shamir proposes a $(t, n)$ threshold scheme to solve this problem based on polynomial interpolation: given $t$ points in the dimensional plane $(x_1, y_1) \ldots (x_t, y_t)$, with distinct $x_i$'s, there is one and only one polynomial $q(x)$ of degree $t-1$ such that $q(x_i) = y_i$ for all $i$. To divide the secret $D$ into $n$ pieces, he suggests picking a random $t-1$ degree polynomial $q(x) = a_0 + a_1 x + \cdots + a_t x^{t-1}$ in which $a_0 = D$, and each piece is the value of the polynomial at the $n$ points: $D_1 = q(1), \ldots, D_i = q(i), \ldots, D_n = q(n)$. Thus any subset of $t$ of the pieces can determine the coefficients of the polynomial (using e.g. Lagrange interpolation) and thus the secret data at a certain point. He suggests the use of modular arithmetic instead of real arithmetic. The set of integers modulo a prime number $p$ forms a field in which interpolation is possible.

This scheme was later employed by many researchers to construct a distributed PKG in IBC and to solve security problem in MANETs.

Zhou *et al* [99] suggest the use of Shamir's threshold scheme to secure ad hoc networks. The authors identify the problem to establish a key management service using a single CA in ad hoc networks. They suggest distributing this service to an aggregation of nodes.

Zhou *et al* refer to the work of Desmedt *et al* [32, 31] and indicate that they use the theory of threshold cryptography as a basis for their work. The authors propose a distributed CA architecture and PKI used in ad hoc networks. The CA service, as a whole, has a public/private key pair $K/k$. The public key $K$ is known to all nodes in the network, whereas the private key $k$ is divided into $n$ shares $s_1, s_2, ..., s_n$ with one share for each server. To provide the certificate signing service, threshold cryptography algorithm is used—for a message $m$, server $i$ can generate a partial signature $PS(m, s_i)$ using its share $s_i$ and forward the signature to a combiner. If $t$ out of $n$ partial signatures are collected by the combiner, they can jointly perform the operation correctly.

The idea of distributed CA has been subsequently adopted for distributed PKG in many IBC proposals in MANETs later.

### Key Generation Using Traditional Threshold Cryptography

PKG plays a fundamental role in an identity-based cryptosystem, but it is not trivial to have a robust PKG in a MANET environment. As Zhou *et al* have suggested [99], a CA service of PKI can be distributed to multiple nodes in a MANET environment. This idea is also applicable to IBC.

Khalili *et al* [52] propose to use IBC to secure ad hoc networks. The authors refer to the work of Zhou *et al* [99] and Bobba *et al* [5] and identify the problem that all proposed key management solutions assume either pre-existing shared secrets among nodes or the presence of a common PKI. They propose to combine efficient techniques from identity-based and threshold cryptography to provide a mechanism that enables flexible and efficient key distribution while respecting the constraints of ad-hoc networks. At the time of network formation, the participating nodes form a threshold PKG, and generate—in a distributed fashion—a master public key. The master secret key is shared in a $t$-out-of-$n$ threshold manner by this initial set of $n$ nodes. All nodes in the network can use their identities as their public keys. The secret key, corresponding to the public key, is computed by having the node obtain $t$ shares of their key from $t$-out-of-$n$ of the original nodes. All subsequent communications are encrypted and decrypted using the master public key and the ID of the recipient. The authors based their proposal on Boneh's identity-based cryptosystem algorithms [10].

As a detailed implementation of Khalili's idea, Deng *et al* [29, 28] propose an identity-based key management and authentication system for MANET, using identity-based and threshold cryptography. The proposed approach consists of two components: distributed key generation and identity-based authentication. This paper describes algorithms for master key generation, distributed private key generation, new master key share creation. The system was built on the assumption that each mobile node has a mechanism to discover its one-hop neighborhood and to get the identities of other nodes in the network. The key generation component provides the network master key pair and the public/private key pair to each node in a distributed way. The system public key/master key pair is computed collaboratively by the initial network nodes without constructing the master key at any single node, as Shamir and Zhou suggested [80, 99] [1]. The public key of node $ID$

[1]Each node $C_i$ randomly chooses a secret $x_i$ and a polynomial $f_i(z)$ over $\mathbb{Z}_q$ of degree $t-1$, such that $f_i(0) = x_i$. Node $C_i$ computes his sub-share for node $C_j$ as $ss_{ij} = f_i(j)$ for $j = 1, 2...n$ and sends $ss_{ij}$ securely to $C_j$. After receiving $n-1$ sub-shares, node $C_j$ can compute its share of master private key as $S_j = \sum_{i=1}^{n} ss_{ij} = \sum_{i=1}^{n} f_i(j)$. Any coalition of $t$ shareholders can jointly recover the secret as in basic secret sharing: $s = \sum_{i=1}^{t} S_i l_i(z) mod\ q$ , where $l_i(z)$ is the Lagrange coefficient. Due to the homomorphic property of share refreshing, the jointly generated master key is equal to $\sum_{i=1}^{n} f_i(0)$.

can be computed as $Q_{ID} = H(ID||ExpireTime)$.

Another implementation of Deng's scheme is described in Zhange *et al*'s work [93]. The authors implemented a scheme with distributed master key generation, private key generation, secret share update, and secret share generation for a new joining node. One thing they did not mention is how secret shares are distributed to other nodes from one node.

Xia's scheme [87] is also very similar to Deng's scheme: A set of Distributed PKG (DPKG) nodes collaboratively generate system public key and master key in a fully distributed manner; Shares can be updated among PKGs; New nodes can get their shares from PKGs and become new PKG nodes.

Differences from Deng's scheme are:

1. This scheme does not use temporary PKI for secret share distribution as in Deng's scheme. Instead, it employs a self-generated public/private key pair in the following way: each DPKG node computes a temporary public key and sends it to other DPKG nodes. Secret shares are encrypted and decrypted using this temporary public key.

2. The author applies IBC to OLSR routing protocol, particularly use HELLO messages and TC messages in OLSR to select and mark DPKG nodes, while Deng *et al* apply IBC to DSR routing protocol.

These differences lead to the following problems:

1. Each DPKG node has to store in memory the temporary public keys of other DPKG nodes.

2. System public key and master key collection process is not secure, because only public channels are available at this stage.

3. The keys generated are not guaranteed secure, because it does not provide any security protection for OLSR routing protocol it relies on.

All of these schemes use threshold cryptography to distribute the functionality of PKG to multiple nodes. Due to threshold cryptography, these schemes have the following issues:

1. Interdependency cycle between secure routing and security services: These schemes rely on some existing routing or online administration mechanisms (e.g. out-of-band communicant, side channel) to distribute secret shares among the distributed

PKG nodes. On the other hand, as we will show later, most secure routing protocols rely on secure keys. We identify this as interdependency cycle between key management (KM) and secure routing (SR) in IBC, and as a specialization of noted problem of interdependency cycle between security services and secure routing [65, 88] . These schemes cannot be used in secure routing protocols.

2. Proximity-caused insecurity: In some circumstances where a node can move in order to access to more nodes, one way to avoid the KM-SR interdependency cycle problem is to have a threshold number of authorized users that are physically close to each other (i.e., within one-hop communication distance so that routing is eased). This incurs another related problem—the proximity-caused insecurity: it is possible that an adversary compromises these nodes within a short period of time (e.g., by capturing the nodes and/or compromising them one by one ) [88]. Furthermore, the proximity-based solution is not applicable to fully distributed key generation schemes where all nodes participate in and contribute to the key generation and thus routing connecting all nodes (not only among a threshold number of nodes) is still required.

3. Mobile Attacks: Threshold cryptography is subject to mobile attacks, in which a mobile adversary could move to compromise multiple nodes and reveal the secret shares of them in order to recover the secret. To counter mobile attacks, the above proposals use secret refreshing mechanism in which secret shares are updated in intervals and new shares cannot be combined with old ones to recover the secret. They assume there is only one mobile adversary in the network and a mobile adversary cannot compromise enough authentic nodes within the share refreshing period. Many researchers, e.g Merwe *et al* in [65], do not think this assumption is practical.

We will recall and discuss this problem further in Section 3.2 shortly.

**Multicast Group for Threshold PKG**

Li *et al* [58] point out that share refreshing in [99] needs a secure channel for delivering subshares, of which Zhou *et al* did not provide the implementation. They propose a signcryption scheme that exactly provides a way for secure transmission, by using periodic private keys, multicast group of PKGs, and key proxy. Their work is based on work of Shamir [81], Zhou *et al* [99] and Boyen [16].

Li *et al* introduce a key proxy for key generation. A key proxy is selected from a group of server nodes: all server nodes form and maintain a few multicast groups according to location. A node floods its Routing REQuest (RREQ) to find a route to the server nodes group. When it receives Routing REPlies (RREPs) from server nodes, it selects a server node, say $u$, which has the shortest path to itself as its key proxy. The routing information to the node $u$ is stored. When it wants to update its private key later, it sends its Private key update REQuest (PREQ) to $u$ and $u$ multicasts the PREQ to all server nodes. The private key of a node is updated periodically. Server node computes a partial private key of the client ($d_{A,i}$) using its master key share, then signcrypts and sends it in a Private key update REPly (PREP) message to $A$.

In order to check malicious server nodes, at the initial time of the network, PKG publishes a piece of verification information consisting of $s_i \cdot P$ for each server node $i$. To check the validity of partial key it receives from $i$, node $A$ needs only to check whether the equation $\hat{e}(Q_A, s_i \cdot P) = \hat{e}(d_{A,i}, P)$ holds.

Li *et al* use "proactive threshold" similar to Zhou *et al*'s [99], with two modifications: replacing secure channel with multicast, and replacing a secret share with a vector. The share vector is encrypted and multicast to the server nodes group. Every server node can only decrypt its own share.

This scheme distributes partial private keys of PKG server nodes to the network before starting for future secure communication, in a way like certificates in PKI. This is against IBC advantages. The multicast group of PKGs is fundamental in the scheme, but a critical question remaining open in this work is how the multicast group is formed. Secure multicast routing cannot be established without secure keys. Thus the KM-SR interdependency cycle problem is not addressed.

**Offline Threshold PKG**

Zhang *et al* [96] propose a distributed PKG (D-PKG) scheme to distribute PKG of IBC to multiple nodes, based on work of Shamir [81], Zhou *et al* [99] and Boneh *et al* [10]. The master key of the IBC system is distributed to D-PKGs in an offline manner, and then a threshold number of D-PKG's can function as PKG. In each D-PKG, the Trusted Authority (TA) supplements the network bootstrapping process with the following operations [96, p. 3517]:

1. Determine a $(t-1)$-degree ($1 \leq t \leq N$) polynomial, $h(x) = s + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1} (mod\ q)$.

2. Select $n$ $(t \leq n \leq N)$ nodes as D-PKGs (denoted by $SH$). Each node in $SH$ gets a share of $s$ as $s_k = h(k)$.

3. Calculate a set of share commitments as $SC = \{P_k = s_k \cdot P \in \mathbb{G}_1 | 1 \leq k \leq n\}$.

$SH$ and $SC$ are appended to the public system parameters and sent to all nodes. Similar to schemes using traditional threshold cryptography presented above, any combination of $t$ D-PKGs can collectively reconstruct the system master-key $s$.

These D-PKG's collaboratively provide the PKG service: Node $B$ sends them a private-key sub-request containing its public key $ID_B$. Upon receiving the request, each chosen D-PKG sends back a sub-reply containing a partial private key: $d_{B,i} = s_i H_1(ID_B || other\_Info)$. $other\_Info$ may contain version number or expire time etc. $B$ can verify its authenticity using $P_i$: $\hat{e}(d_{B,i}, P) = \hat{e}(H_1(ID_B || otherInfo), P_i)$ [2]. After obtaining $t$ authentic private-key pieces, $B$ can calculate the complete private key in the same way computing the master-key.

This scheme is similar to schemes using traditional threshold cryptography, but differs in the following ways: this scheme distributes secret shares offline, and thus does not require on-line secure channels for secret share distribution; the secret shares of this scheme are not refreshed or updated, thus it is more subject to mobile attacks. Although the master key generation does not require secure channels, the private key generation still needs them; thus, KM-SR interdependency cycle is not addressed. Also, the share commitments of each D-PKG are used like certificates which are distributed to the network nodes before network starts. This is against IBC advantages.

**Public Channels for Threshold PKG**

Ren *et al* [75] propose another D-PKG scheme. The scheme eliminates the secure channel requirement by using mutual authentication in public channels.

The key generation and issuing works as follows: A user $U_{ID}$ chooses a password $pwd$ and computes $H_1(ID), H_1(pwd), H_2(pwd)$. Then it publishes the tuple $\langle ID, H_1(ID), H_1(pwd), H_2(pwd) \rangle$. The D-PKGs store them in their database. User selects a random number $r$ and computes a request and sends the request to D-PKGs. D-PKGs checks the validity of the request and computes blinded partial private key and sends it to the user. The user upon receiving blinded partial private keys verifies them and unblinds the private key using the proprietary knowledge of $r$.

[2] The verification process is same as Li's scheme *et al* [58] in subsection 3.1.1

The authors claim that the protocol does not require any secure channel to issue the private key and is secure. However, D-PKGs have to store a password for each user, in the same way as the distributed CA works in PKC mechanisms. This violates the advantages of identity-based cryptosystems, and requires online service from D-PKGs. Also, the paper did not mention how requests and secret shares are transmitted in public channel. We assume they use broadcast in the discussion below. In that case, the KM-SR interdependency cycle problem is addressed, but two problems remain: first, security of the private keys is not guaranteed because they only use hashing to protect the private keys; second, it is not efficient because of communication and computational overhead of broadcast.

**A Threshold Key Generation Scheme with Compromise-tolerant Key-update Parameters**

Fang *et al* [95] propose a key generation scheme that provides compromise-tolerant feature for private keys. This is achieved by dividing public/private keys into node-specific and phase-specific components, and predistributed key-update parameters.

The cryptographic materials distributed to each node before network deployment include: pairing parameters: $\langle p, q, \hat{e}, H_1, P, s_1P, s_2P\rangle$, public and private keys: $\langle Q_{ID,0}, d_{ID,0}\rangle$, phase salt: $salt_1$, key-update parameters: $\langle\{v_i(x), l_i(ID)\}_{i=1,\ldots,m}\rangle$, where $m$ is the maximum possible phase index, $H_1$ is a hash function that maps a string to a non-zero element in $\mathbb{G}_1$, $s_1$ and $s_2$ are two distinct master keys. PKG distributes $s_2$ to D-PKGs using threshold secret sharing, each D-PKG $V \in \Omega$ holds a secret share $s_{2V}$ and a set of values $\{P_{2V} = s_{2V} \cdot P | V \in \Omega\}$ where $\Omega$ is the D-PKG set, and $|\Omega| = n$.

Each public/private key pair is both node-specific and phase-specific. At phase-$i$, node $A$'s public key is $Q_{A,i} = \langle H_1(ID_A), H_1(salt_i)\rangle$, private key is $d_{A,i} = \langle s_1 \cdot H_1(ID_A), s_2 \cdot H_1(salt_i)\rangle$. The first element of each key is node-specific, and the second element is phase-specific. Initially, the PKG issues $Q_{A,1}$ and $d_{A,1}$ to node A. $A$ can acquire phase-specific element $Q_{i+1} = H_1(salt_{i+1})$ and $d_{i+1} = (s_2 \cdot H_1(salt_i))$, where $salt_{i+1} = salt_i + 1$, from the D-PKG set through key update. In the key update, a D-PKG node $Z$ contacts $t-1$ D-PKG, and collects $t$ shares of $d_{i+1}$ and generates $d_{i+1}$ using a $t$-out-of-$n$ threshold cryptography. $Z$ then broadcasts $d_{i+1}$ to unrevoked nodes securely using a variant of the self-healing group key distribution scheme by Liu *et al* [61].

The key update parameters also facilitate key revocation feature, which we will discuss in a later section.

This scheme employs threshold cryptography for generation of phase-specific componenets of private keys online. It is not clearly stated how D-PKGs communicate with each other to exchange secret shares. This process either relies on secure routing which leads to KM-SR interdependency cycle problem, or relies on broadcasting which incurs insecurity and extra traffic overhead. In addition, the scheme does not have good scalability because the size of key-update parameters to be distributed to nodes before network deployment is proportional to number of phases and number of D-PKGs, both of which can become very large.

**A Non-threshold Key Issuing Scheme for High Key Privacy**

Threshold PKG key generation allows redundant PKGs for high availability of master key. The opposite way is to use a chain of key privacy authorities (KPAs) to protect master key for high privacy. Lee *et al* [54] propose a secure key issuing protocol in which a private key is issued by a key generation center (KGC) and then its privacy is protected by multiple key privacy authorities (KPAs). For all $i = 1, \cdots, n$, $KPA_i$ chooses his master key $s_i$ and computes his public key $P_i = s_i P$. Then KPAs cooperate sequentially to compute the system public key $Y = s_0 s_1 ... s_n P$.

A user $ID$ gets its private key in three stages [54, p. 73]:

1. In key issuing stage, a node sends its identity $ID$ and blinding factor $X = xP$ to the KGC and requests him to issue a partial private key. The KGC issues a partial private key to the user in a blinded manner: $Q'_0 = H_3(\hat{e}(s_0 X, P_0)) s_0 Q_{ID}$, together with a signature: $Sig_0(Q'_0) = s_0 Q'_0$. Here $H_3(\hat{e}(s_0 X, P_0))$ is a blinding factor. User can unblind it using his knowledge of $x$ [3].

2. In key securing stage, the user requests multiple KPAs in a sequential manner to provide key privacy service by sending $ID$, $X$, $Q'_{i-1}$ and $Sig_{i-1}(Q'_{i-1})$. Then KPAs return the private key shares: $Q'_i = H_3(\hat{e}(s_i X, Pi)) s_i Q'_{i-1}$ and signature $Sig_i(Q'_i) = s_i Q'_i$ in a blinded manner.

3. Finally, in key retrieving stage, the user unblinds it to retrieve the real private key: $d_{ID} = \frac{Q'_n}{H_3(\hat{e}(P_0,P_0)^x) \cdots H_3(\hat{e}(P_n,P_n)^x)} = s_0 s_1 \cdots s_n Q_{ID}$. The user can verify the correctness of his private key by $\hat{e}(d_{ID}, P) = \hat{e}(Q_{ID}, Y)$.

The authors have analyzed the security of this scheme and state that since the private key of a user is computed cooperatively by the KGC and n KPAs, the privacy of user's

[3] $H_3(\hat{e}(s_0 X, P_0)) = H_3(\hat{e}(s_0 xP, P_0)) = H_3(\hat{e}(P_0, P_0)^x)$

private key is kept if at least one authority remains honest. Only the legitimate user who knows the blinding parameter can unblind the message to retrieve the private key.

This scheme was not originally designed for MANETs. In a MANET environment, it has the following weaknesses: first, all KPAs are required to be online and available, which is not feasible in MANETs; second, secure routing is required to get partial key and signature, which is in KM-SR interdependency cycle.

**A Non-PBC Lightweight IBC Key Generation Scheme**

Saxena [79] proposes a scheme of public key cryptography for MANET analogous to identity-based cryptography with some claimed advantages. This scheme can be considered as a lightweight IBC. This work is based on work of Zhao *et al* [99], Shamir [80] and Feldman [36] on threshold cryptography, and on the work of Boneh *et al* [10] on IBC.

The author suggests the use of Feldman's *Verifiable Secret Sharing (VSS)* [36] to generate private keys and public keys. In order to setup the system, a dealer (or a set of co-founding members) first chooses appropriate parameters $(p, q, g)$ for the group, and selects a polynomial $f(z) = a_0 + a_1 z + \cdots + a_t z^t$ in $Z_q$, where $a_0$ is the group secret. The dealer keeps the polynomial secret and publishes commitments to the coefficients of the polynomial, as $w_i = g^{a_i}(mod\ p)$, for $i = 0, \cdots, t$. To join the group, a user $M_i$ sends its unique identifier $id_i$ to the dealer who issues it its secret share $x_i = f(id_i)(mod\ q)$ as the private key for $M_i$. The public key $y_i = g^{x_i}(mod\ p)$ of $M_i$ can be computed by $M_j$ as $y_i = \prod_{j=0}^{t}(w_j)^{id_i^j}(mod\ p)$. Also $M_i$ can compute $M_j$'s public key as: $y_j = \prod_{i=0}^{t}(w_i)^{id_j^i}(mod\ p)$, and pairwise shared key as: $k_{ij} = y_j^{x_i} = g^{x_j x_i} = k_{ji}(mod\ p)$. With these keys, they define the sign/verify and encrypt/decrypt methods as counterparts to Boneh's (see [79, p. 382] for detail).

The author points out that the proposed scheme can be viewed as an IBC based on threshold assumption. Knowing the identifier of a particular user and also the public key of the trusted center, one can send encrypted messages and verify signatures. This is equivalent to identity-based encryption and signature. The author further states that unlike other IBC schemes, the proposal is based on standard (discrete logarithm) assumptions, and thus is much more efficient than these prior IDC schemes.

According to Xu *et al* [88], Saxena's scheme is arguably subject to Sybil attacks. Besides, the scheme publishes per-node parameter $w_i$ to all nodes to compute public key of user $i$, which is similar to certificate-based schemes and against advantages of IBC.

**A PKI-IBC Hybrid Key Management Scheme**

Traditional PKI is based on PKC. In MANETs, because the computational and communication resources required by PKC operations are very limited, and also a centralized CA is not reliable, traditional PKI is considered unsuitable. By applying IBC, new hybrid PKIs can be setup and adapted to MANETs.

In [60, 49], Lin *et al* identify the difficulty of applying traditional PKI security architecture to MANET. They suggest the use of a hybrid architecture that combines the good sides of both traditional PKI and IBC, and propose a cluster-organized key management scheme.

Based on former work of Boneh *et al* [10], Huang *et al* [48], Zhou *et al* [99] and Shamir [81], they propose a key management scheme and integrate it into secure routing protocols. The proposed network framework is a two-layer hierarchical structure performing key generation, key distribution, and storage. The bottom layer is responsible for internal cluster domain authentication using IBC, and the upper layer, root CA, is responsible for external cluster domain authentication.

In every cluster domain, cluster heads only maintain identities of members, without needs to store and distribute public keys. The cluster head serves as the PKG for cluster members. When a node joins the network, it is given a master public-key belonging to a cluster domain. Furthermore, each node also applies for a personal private-key from its cluster domain head, and uses it to achieve routing packets and messages encryption/decryption capability. The identity-based key generation and distribution use Boneh's algorithms.

The authors state that the simulation results demonstrate that the scheme can reduce computing loads of central CA and key repositories. However, at the same time, the scheme adds much additional overhead to inter-cluster communication.

### 3.1.2 Group Key Generation and Agreement

In cases when a message is intended for every node in a group, using public/private keys and pairwise communication generates tremendous traffic overhead. A symmetric group key minimizes the traffic bandwidth, and is more efficient. The advantage of the group broadcast key is that it needs only at most $n$ private keys to be generated and distributed to $n$ nodes, whereas pairwise communication schemes need $n(n-1)/2$ and $n(n-1)$ keys generated and distributed respectively.

A group key can be generated by one member of the group and distributed to other

members. A group key can also be contributed and agreed by multiple members. A group key can be either dynamic, which means in each broadcast message the group key is different; or static, which means the group key does not change in each broadcast message once it is determined. In this subsection, we classify group key generation and agreement schemes based on these criteria.

**Dynamic Group Key Generation Based on Node-specific Broadcast Secret**

If the members of a group of nodes share a secret that is unknown to non-members, it is intuitive that they can generate a share group key based on this secret. Many group key generation schemes are based on this idea. The differences only lie in how the shared secret is generated and how it is distributed to members.

Bohio *et al* [7] propose a non-probabilistic method for computing unique broadcast keys for different groups. Based on the work of Cha *et al* [18], they use identity-based pairwise symmetric keys as the building block for their broadcast scheme. They state such keys are computed non-interactively by the nodes, which reduces communication overhead and simplifies key management in pairwise communication.

The group key is generated in this way: Let $K_{1N}$ be the broadcast secret of node 1 for any group of $N$ nodes. Node 1 computes its broadcast parameter $P_{1-brdcst}$ as: $P_{1-brdcst} = K_{1N} \cdot Q_{id_1}$, and distributes it to all candidate nodes using respective pairwise encryption. To sign and encrypt a message $M$, node 1 computes:
$h = H_3(M)$, where $H_3 : \{0,1\}^* \rightarrow \{0,1\}^*$;
$K_{1-brdcst} = H_2(\hat{e}(Q_{id_1}, P)^{(r+h)})$, where $r \in Z_q^*, H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^m$; $C = M \oplus K_{1-brdcst}$, $U = rP, V = K_{1N}^{-1}(r+h)P$.
The broadcast message is: $\langle C, U, V \rangle$. Every node in the group can compute the same broadcast key $K_{1-brdcst}$ as node 1 from $H_2(\hat{e}(P_{1-brdcst}, V))$ and decrypt the message from the cipher text $C$ as: $M = C \oplus K_{1-brdcst}$; After decrypting message , its hash can be computed as: $h = H_3(M)$, and authentication is verified by checking if $\hat{e}(K_{1N}Q_{id_1}, V) = \hat{e}(Q_{id_1}, U + hP)$ holds.

In [6], Bohio *et al* continue their work and indicate that the use of pairwise communication creates additional bandwidth overhead in case of broadcast messages. They propose an authenticated broadcast scheme based on symmetric keys and a corresponding signature scheme. Based on work of Boneh *et al* [10] and Bohio *et al* [7], the authors extend pairwise shared key generation method proposed in [78] — $K_{AB} = K_{BA} = \hat{e}(Q_{id_A}, sQ_{id_B})$, and propose a method for computing collision-free broadcast keys that

can be used for different groups in the network and changed as the group membership varies. Such keys can be useful in the context when it is important to have all the broadcast keys unique without causing additional handshake between the nodes.

Compared to Bohio *et al* [7], the authors simplify the scheme: Node $1$ computes its broadcast parameter $P_{1-brdcst}$ as: $P_{1-brdcst} = K_{1N} \cdot P$, and distribute it to all candidate nodes using respective pairwise encryption. Every node will then compute the broadcast key of node $1$ as $K_{1-brdcst}$ using the hash function $H_3 : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow (0,1)^m$. The key $K_{1-brdcst} = H_3(P_{1-brdcst})$. To generate unique broadcast secret $K_{1N}$ for node $1$, let $D_{1N} = \hat{e}(sQ_{id_1}, Q_{id_2} + Q_{id_3} + \cdots + Q_{id_n}) = \hat{e}((sQ_{id_1}, Q_{id_2}) \cdot \hat{e}((sQ_{id_1}, Q_{id_3}) \cdots \hat{e}((sQ_{id_1}, Q_{id_n})$ and $K_{1N} = H_2(D_{1N})$. Further, the authors use this group key to sign group messages $M$: $\langle U, V \rangle = \langle rQ_{id_1}, K_{1N}^{-1}(r+h)Q_{id_1} \rangle$ where $r \in Z_q^*$, $h = H_4(M)$. And the receiver can verify if $\hat{e}(P_{1-brdcst}, V) = \hat{e}(P, U + hQ_{id_1})$ holds.

The authors point out one potential problem of this scheme is that it might be possible for malicious nodes to generate computational overhead for other nodes by sending unnecessary broadcast messages. The countermeasure is the non-repudiation and authentication provided by the signature in the scheme.

In [8]—the extended version of [7] and [6]—the authors reiterate their scheme to generate collision-free broadcast keys for different groups and an authenticated broadcast scheme based on symmetric keys and a corresponding signature scheme. On the basis of the former two papers, the authors present two variants of their former scheme to generate group keys hidden to the TA:

The first scheme is based on group identity. A group public key $Q_{GRP-ID}$ is to be generated by the TA based on any group identity or arbitrary string. The TA, using its master key, then computes the initial group key $D = s \cdot Q_{GRP-ID}$. Every node $i$ will then receive the point $D$ from the TA and will generate its private key $k_i$, a random secret, and compute the corresponding public key as $D_{i-pub} = k_i \cdot D$. All such individual public keys should be available from the TA. The participating nodes then get the public key of every node from the TA.

For the broadcast key, parameter $P_{1-brdcst} = K_{1N} \cdot P$ is computed as in the basic scheme with $K_{1N}$ being any random secret. The signature scheme would be used as in the basic model.

The second scheme is based on individual identity. The TA will compute the partial private key of any node $i$ as $D_i = s \cdot Q_{id-i}$. Node $i$ computes its private key as $k_i = H_3(x_i \cdot D_i)$, where $x_i$ is a random secret chosen by node $i$. It computes public key as $D_{i-pub} = k_i \cdot P$, and submits it to the TA. The pairwise and broadcast keys will be

computed similarly as the first scheme does.

It has been pointed out by Chien *et al* in [22] that the above signature scheme is vulnerable to the universal forgery attack that an adversary can forge signatures on any message.

For group key generation schemes based on broadcast secret, one issue is how the broadcast secret is distributed to other nodes in the group. If it is distributed by broadcasting, the issue turns to be scalability problem. Each node generates a group key secret and broadcasts it to other nodes. The number of messages and storage space are both $O(n)$, the broadcast traffic is $O(n^2)$ (each of $n$ nodes relays $n$ messages). If the group broadcast secret is distributed using respective pairwise communication, it requires an existing secure routing mechanism. The issue turns to be KM-SR interdependency cycle problem. Another issue is that each node generates a broadcast secret and distributes it to other nodes in the group. This is against the advantages of IBC schemes.

**Static Group Key Agreement Based on Diffie-Hellman Key Exchange**

An approach for group key generation is "recursive subgrouping"—dividing a large group to subgroups again and again, each subgroup contains a small number of sub-subgroups until a small number of members reached. For these small number of members, there are already key exchange protocols ready to use, e.g., 2-party or 3-party Diffie-Hellman key exchange protocol.

Chien *et al* in [21] and [22], propose a group key agreement protocol in this approach, based on work of Rhee *et al* [76], Kong *et al* [53] and Bohio *et al* [8], and apply IBC to these schemes. In their scheme, they divide the whole group into several cell groups and a control group, and each cell group is managed by its cell group controller independently of the other cell groups. Nodes within the same cell group share a cell group key, which can be generated by a distributive or contributory way.

They provide two versions of pair-wise key agreement: one is static and the other is dynamic. The static one uses the same static pair-wise key as Bohio-Miri's scheme [8]. The dynamic one, contrary to Bohio-Miri's scheme, is certificate-less. The protocol works as follows: $A \rightarrow B : P_A = aP$, $B \rightarrow A : P_B = bP$, where $a, b$ are random numbers. Then $A$ and $B$ independently compute a common session key based on $P_A$ and $P_B$.

On the basis of the pair-wise communication, they propose a *Tripartite key agreement protocol* which allows three parties establish their session keys. The scheme is modified from Hess' signature [44] for traditional public key setting. The protocol has two rounds.

In the first round, the entities broadcast their ephemeral public keys, e.g. $A \rightarrow B, C$: $\langle sid, ID_A, ID_B, ID_C, P_A, P'_A \rangle$, Node $A$ computes $P_A = aP, P'_A = a'P$, where $a$ and $a'$ are random numbers chosen by node $A$, $sid$ is session id. In the second round, the entities broadcast their confirmation (signatures) on the session and ephemeral public keys, e.g. $A \rightarrow B, C$: $\langle sid, v_A, u_A \rangle$, Node $A$ computes $m_A = H_3(sid, ID_A, ID_B, ID_C, P_A, P'_A, P_B, P'_B, P_C, P'_C), r_A = \hat{e}(P, P)^{K_A}, v_A = H_4(m_A, r_A)$ and $u_A = v_A S_A + k_A P$, where $K_A$ is a random number chose by node $A$. $B$ and $C$ broadcast similar messages. Then $A$ checks whether the following two equations hold: $v_B = H_4(m_B, \hat{e}(u_B, P) \cdot \hat{e}(Q_B, P_{pub})^{-v_B})$ and $v_C = H_4(m_C, \hat{e}(u_C, P) \cdot \hat{e}(Q_C, P_{pub})^{-v_C})$ [4]. After authenticating the message from the other two nodes, $A, B$, and $C$ share these session keys: $K^1_{A,B,C} = \hat{e}(P_B, P_C)^a, K^2_{A,B,C} = \hat{e}(P_B, P'_C)^a, K^3_{A,B,C} = \hat{e}(P'_B, P_C)^a, K^4_{A,B,C} = \hat{e}(P'_B, P'_C)^a, K^5_{A,B,C} = \hat{e}(P_B, P_C)^{a'}, K^6_{A,B,C} = \hat{e}(P_B, P'_C)^{a'}, K^7_{A,B,C} = \hat{e}(P'_B, P_C)^{a'}, K^8_{A,B,C} = \hat{e}(P'_B, P'_C)^{a'}$.

The tripartite key agreement scheme can be easily extended to share $n^3$ keys by sending $n$ ephemeral public values per node. The scheme then uses the ternary tree and bilinear map to establish the cell group key. Hierarchical ternary tree is a hierarchical tree, where the degree of a node is at most three. The keys corresponding to the key nodes are generated iteratively from bottom up to the root node, and the key corresponding to the root node is taken as the group key. If a node has three child nodes, then the tripartite key agreement scheme is adopted; otherwise, the two-party key agreement scheme is adopted.

This scheme addresses the scalability issue by subgrouping, but is subject to these problems: first, each node generates an ephemeral key and distributes it to group members, which is against advantages of IBC; second, key exchange messages use respective pairwise communication, which requires an existing secure routing mechanism.

**Static Group Key Agreement Based on Broadcast Ephemeral Keys**

Characteristics of MANETs make it difficult to generate a group key. Zhang's constant-round contributory key agreement scheme [92] avoids the two obstacles for contributory key agreement in MANETs: authenticating the exchanged information without an online Trusted Third Party (TTP), and resistance to unstable links.

Using the IBC scheme of Boneh *et al* [10], the authors revised the constant-round key agreement scheme proposed by Lee *et al* [55] that was on password-based. In round 1 of the new scheme, each node generates an ephemeral key $N_i \in \mathbb{Z}^*_q$, computes $z_i = N_i P$, and signs it using the signature scheme of Du *et al* [33]: $T_i = H(z_i) sQ_i + N_i P_{pub}$. The

[4] $\hat{e}(u_B, P) \cdot \hat{e}(Q_B, P_{pub})^{-v_B} = \hat{e}(v_B sQ_B + k_B P, P) \cdot \hat{e}(Q_B, sP)^{-v_B} = \hat{e}(sQ_B, P)^{v_B} \cdot \hat{e}(k_B P, P) \cdot \hat{e}(sQ_B, P)^{-v_B} = \hat{e}(P, P)^{k_B}$

node then broadcasts them with its ID:$\langle z_i, T_i, ID_i \rangle$.

In round 2, each of the group member firstly verifies $\hat{e}(\sum_{j\in\{1,\cdots,n\}\setminus\{i\}} T_j, P) = \hat{e}(\sum_{j\in\{1,\cdots,n\}\setminus\{i\}}(H(z_j)Q_j + z_j, P_{pub})$. Then group members are divided into two subgroups. Only one subgroup broadcasts messages, and two subgroup keys are generated once a time. Each node computes a group key based on two sub-group keys. In short, for every group key's information exchange at round 2, it only needs about half of group members to take part in, while all members can compute out the same session keys according to the broadcasted messages. This group is divided into two subgroups, and as long as one of these two subgroups does not meet with the link failures, this scheme will succeed.

This scheme requires an ephemeral key for each node which is stored on all other nodes. This is a drawback inherited from certificate-based cryptography, and is against the advantages of IBC.

**Static Group Key Generation Based on Identity-based Broadcast Encryption**

Zhang *et al* [91] propose another group key generation protocol that is quite different from the above schemes. The scheme is based on Identity-based broadcast encryption (IBBE) scheme [27]. In IBBE, one public key can be used to encrypt a message to any possible group of identities.

The proposed scheme only requires each group member to broadcast one message to set up the group key. Compared to Bohio's scheme, this scheme does not require a node to store any temporary or pseudo public key of other nodes. Compared to above schemes, the scheme does not require secure routing for key exchange message, because all messages are broadcasted. However, the group key generation is static and not suitable for dynamic networks, such as MANETs, because it requires all members be determined before protocol starts. In case of membership changes, for example, one member leaves or one new member joins, all members must start the process again. Besides, like other group key schemes discussed above, IBBE group keys are symmetric keys; but unlike them, IBBE is not integrated with any asymmetric private/public key scheme. A different set of parameters and algorithms is needed for asymmetric private/public keys generation which is indispensable for authentication and non-repudiation.

## 3.2 Secure Routing Protocols Using IBC

Routing in MANETs enables packet delivery from one node to another by way of intermediate nodes. It is the fundamental issue considered in MANETs, thus secure routing is a fundamental issue in MANET security. Secure routing ensures successful routing among authentic nodes with adversary nodes existing around or inside the network, and forms the bedrock of a secure MANET system. An important application of IBC in MANETs is to design secure routing protocols. Depending on what encryption/decryption and signature/verification schemes are used, and what routing protocols are used, there are various secure routing proposals using IBC.

### 3.2.1 Securing On-demand Routing Protocols

Lee, Kim, Chung and Yoon [56] apply previous IBC schemes [10, 72] to a DSR routing protocol.

In their routing protocol, the format of a route request packet is $\langle RReq, SourceID, DestinationID, seq, Sign_S(M), (IntermediateIDList), W, U, V\rangle$, where $M = \langle RReq ||SourceID||\ DestinationID||\ seq||\ W\rangle$, and $Sign_S(M)$ is a signature algorithm from [72]. Assume $Q_i = n_i \cdot P$ is the public key of a node ($n_S$ for the source node, $n_D$ for the destination node.[5]), and $d_i = s \cdot Q_i$ is its private key, the source node computes $W, U, V$ as follows: It generates a random string $\sigma_S \in \{0,1\}^n$, and computes $r = H_3(ID_{Source}, \sigma_S)$; Using $r$ and its private key $d_S = s \cdot n_S \cdot P$, it computes: $g = \hat{e}(P,P), \hat{e}(rP, d_S) = g^{r \cdot s \cdot n_S}$. Then $W = rP$, $U = g^{r \cdot s \cdot n_S} \times \sigma_S$, $V = (\hat{e}(sP, Q_{Dest}))^r \oplus r = g^{r \cdot s \cdot n_D} \oplus r$.

An intermediate node $i$ that receives route request packet verifies the signature value. If it is correct, node $i$ adds $ID_i$ to the $intermediateIDList$, computes the new value of $U$ by: $U = U \times \hat{e}(rP, d_i) = U \times g^{r \cdot s \cdot (n_S + \ldots + n_i)}$, and then rebroadcasts the packets generated.

A destination node $D$ that receives routing request packet and whose ID is matched to value of $DestinationID$ field in the packet performs the following procedure [6]: computes $r'$ using private key of $D$ and the values of packet received: $r' = V \oplus (\hat{e}(sP, Q_{Dest}))^r = V \oplus \hat{e}(W, s \cdot n_D \cdot P)$, gets the public key $Q_i = H_2(ID_i)$ of $ID_i$ that are described in $intermediateIDList$ and computes $A = \hat{e}(sP, \sum_{i=1}^{k} Q_i)^{r'} = \hat{e}(sP, \sum_{i=1}^{k} (n_i \cdot P))^{r'} = g^{r' \cdot s \cdot \sum_{i=1}^{k} n_i}$. Using $A$ value, $D$ computes $\sigma' = U \times A^{-1}$, and compares $r'$ and $H_3(ID_S, \sigma')$. If the two values are equal, $D$ makes route reply packet as $\langle RRep,\ seq,\ (ID_S, ID_1,\ ...,\ ID_k, ID_D), W, V \oplus \sigma', Sign_D(M)\rangle$, where $M = \langle RRep||seq||ID_S||ID_1||\ ...\ ||ID_k||ID_D$

[5] $n_i$ is only a helper for explanation purpose here, and is unknown to any node.

[6] with correction to the original paper

$||W||V \oplus \sigma'\rangle$.

After receiving the route reply packet, the intermediate nodes in routing path and source node $S$ verify the signature of $D$. And if it is correct, they add the path in the packet to their route cache.

This scheme is subject to wormhole attacks [68]—an adversary can tunnel a valid $RReq$ packet from an intermediate node to the destination to pretend that they are connected. It also misses a key management scheme.

### 3.2.2 Concatenated Signature for Intermediate Node List in On-demand Routing Protocols

Park, Myung and Lee [71] base their work on that of Boneh *et al* [10] and Quisquater [59], and apply IBC to on-demand routing protocols.

Their protocol is similar to Lee *et al*'s [56], but the signature and verification procedures are different:

When the source node sends $RReq$ to intermediate nodes, the packet format is: $\langle RReq|| ID_S||(r_S, Z_S)||Sign_S(H(M))\rangle$, where $M = \langle RReq||ID_S||(r_S, Z_S)\rangle$, $r_S = H(\hat{e}(P, sP)^x ||Q_S||RReq)$, $Z_S = xP_{pub} - r_S d_S = xsP - r_S sQ_S$, $x$ is a random number.

An intermediate node $X_i$ computes $k' = \hat{e}(P, Z_S) \cdot \hat{e}(sP, Q_S)^{r_S} = \hat{e}(P, P)^{xs}$ for the authentication of the node that sends the message, and it checks $r_S = H(k'||Q_S||RReq)$. If the verification is successful, the intermediate node can trust the received message and then it computes $r_X$ and $Z_S$ similarly, and broadcasts the message to the next node as: $\langle RReq||ID_S||ID_X||(r_S, Z_S)||(r_X, Z_X)||Sign_S(H(M))\rangle$.

When the destination node receives this message, it checks the destination address. If the destination address is the same as its address, it verifies the signature, $(r_S, Z_S)$ and $(r_X, Z_X)$. If the verification process is successful, it is ready to reply a message. The destination node sends a $RREP$ message to the source node. After passing intermediate nodes the reply message is like:
$\langle RRep||ID_D||ID_X||(r_D, Z_D)||(r_X, Z_X)||Sign_S(H(M'))\rangle$.

Park and Lee [69], Park, Myung and Lee in [70], Lee and Sriborrirux [55] present similar results separately.

These schemes have these common problems: A key management scheme is missing. Scalability is poor, since message signature is concatenated and can be quite large.

### 3.2.3 Aggregated Signature for Intermediate Node List in On-demand Routing Protocols

The concatenated signature of an intermediate node list can be very large, Song *et al* [82] apply identity-based multi-signature to routing protocols and propose an authentication mechanism with aggregation signature, based on the work of Bonel *et al* [12] and Cha *et al* [18].

In their scheme, an aggregate signature can be generated on distinct messages: assume $\sigma = (U, V)$ is the signature on messages $M_1, \cdots, M_{i-1}$, and $\sigma = (U', V')$ is the signature on message $M_i$, $U = rQ_{ID_i}, h = H_1(M_i), V = (r + h)d_{ID_i}$. The aggregator verifies that $M_i$ is different from any other messages. If it is true, it computes: $U = U + U' \in \mathbb{G}_1$, $V = V + V' \in \mathbb{G}_1$. Then $\sigma = (U, V)$ becomes the aggregate signature on $M_1, \cdots, M_i$. The destination can verify the validity of the aggregation signature: Given identities $ID_1, ..., ID_n$, distinct messages $M_1, ..., M_n$, and an aggregate signature $\sigma = (U, V)$, the verifier computes $h_i = H_1(M_i)$ for all $1 \leq i \leq n$. Then it checks whether $\hat{e}(\sum_{i=1}^{n} h_i Q_{ID_i} + U, P_{pub}) = \hat{e}(\sum_{i=1}^{n} [(h_i + r_i)Q_{ID_i}], P_{pub}) = \hat{e}(\sum_{i=1}^{n} [(h_i + r_i)d_{ID_i}], P) = \hat{e}(V, P)$ holds. If it is true, all the signatures are valid.

They then demonstrate in the paper the use of this scheme in on-demand routing protocols such as DSR and AODV, which is similar to [56].

This scheme is subject to wormhole attacks [68], and misses a key management scheme.

### 3.2.4 A Security Architecture to Secure OLSR

Adjih *et al* [2] propose a security architecture to secure OLSR using IBC.

Their proposal is based on work of Cha *et al* [18] and Boneh *et al* [13]. In their scheme, an (offline) TA is in charge of certifying or assigning keys of each node participating in the trusted network. Each node joining the network will have the public key of the TA. This key is denoted the global key. Later, any node entering the ad-hoc network could diffuse its public keys, with a specific key exchange protocol, with proper parameters and signatures. The key which is used later to sign message is called the local key, and can be either its global key, or newly generated private/public keys. A node would start originating OLSR control messages, signing them using the local key with a specific extension which prepends a special signature message.

Technical details of the scheme are not given in the paper, e.g. how keys are generated

and distributed, how packets are signed and encrypted.

## 3.3 Issues of Key Management and Secure Routing

Table 3.1 summarizes the main characteristics and problems of the master key and private key generation schemes.

Table 3.2 summarizes the main characteristics and problems of group key generation and agreement schemes.

Key management is an essential and fundamental service for ad hoc networks. Secure keys should be set up before other services can start. This can be achieved by predistribution of keys in network initialization phase. One advantage of IBC key management is that it saves storage and transmission of public keys and certificates. Many IBC key management proposals suggest generating master key and private keys online. There is a problem in this case. Consider the following scenario: we need to find a key management scheme to design a secure routing protocol. Since there is no routing for unicasting, the only way to distribute keys or key shares is broadcasting that is not secure. It turns out to be a group key agreement problem, and the group key agreement protocol cannot use unicast routing at that time. Thus key management should not rely on any other online service if keys are generated online. Unfortunately, many IBC key management schemes in the literature do not comply with this rule—they rely on secure routing or online administration mechanisms (e.g. out-of-band communicant, side channel) to generate or distribute keys.

Another issue that needs to be noted for schemes in which a master key is generated in a distributed manner (e.g. [52]) is Byzantine attacks. These schemes need an initial policy negotiation process that is a potential target for Byzantine or active adversaries. The system may be totally taken over by adversaries. For other schemes in which a TA is responsible for the master key generation, this issue does not exist.

For group keys, static group keys are less secure than dynamic group keys, while the latter takes more communication bandwidth in each message. In group key generation/agreement proposals, some use pairwise communication and unicast routing. Key generation/agreement messages are distributed via pairwise communication which relies on unicast routing. This leads to KM-SR interdependency cycle problem, e.g., in [8], the group broadcast key is distributed to all candidates using respective pairwise encryption. This process requires an existing secure routing mechanism.

In both master key and group key generation proposals, one problem is the use of

| **Year** | **Publication(s)** | **Main Idea & Contribution(s)** | **Online /Off-line TA** | **PKG** | **Key Share Distribution** | **problems** |
|---|---|---|---|---|---|---|
| 2003 | [52] | Idea of applying IBC and threshold cryptography to secure ad hoc networks | No | Fully distributed | Secure channel | 1. Technical details of key generation are not given. 2. KM-SR interdependency cycle. 3. Threshold cryptography weaknesses. 4. The network initialization stage is vulnerable to Byzantine failures. |
| 2004 | [29, 28] | A complete implementation of Khalili's Scheme | No | Fully distributed | Temporary PKI | 1. KM-SR interdependency cycle. 2. Threshold cryptography weaknesses. 3. The network initialization stage is vulnerable to Byzantine failures. |
| 2004 | [54] | Secure Key Issuing Protocol Using Key Privacy Authorities | Offline | Partially distributed | Not mentioned | 1. All KPAs are required to be online and available, which is not feasible in MANETs. 2. Secure routing is required to get partial key and signature, which is in KM-SR interdependency cycle. |
| 2005 | [58] | Multicast group of PKGs; Key proxy. | Offline | Partially distributed | Encrypted Multicast | 1. KM-SR interdependency cycle. 2. Distributes partial private keys of PKG server nodes to the network. |
| 2005 | [96] | Offline threshold D-PKG | Offline | Partially distributed | Pre-distribution | 1. KM-SR interdependency cycle. 2. More subject to mobile attacks. 3. Distributes share commitments of D-PKGs |
| 2005 | [79] | Lightweight IBC | Yes | Partially distributed | Not mentioned | 1. Subject to Sybil attacks [88]. 2. KM-SR interdependency cycle |
| 2006 | [95] | Compromise-tolerant Key Generation | Yes | Partially distributed | Not mentioned | 1. KM-SR interdependency cycle problem, or insecurity and broadcasting traffic overhead. 2. Poor scalability. |
| 2007 | [75] | Use of the blind signature to ensure the secure issuing of the private key shares in public channel | Yes | Partially distributed | Public channel | 1. Distribution and storage of password for each node. 2. Security of private keys is not protected. 3. Traffic overhead of broadcasting. |
| 2008 | [93] | Another IBC and threshold cryptography implementation. | No | Fully distributed | Not mentioned | KM-SR interdependency cycle |
| 2008 | [87] | Implementation of Deng's scheme in OLSR routing protocol | No | Fully distributed | Self-generated public/private key pair | 1. Each DPKG node has to store in memory the temporary public keys of other DPKG nodes 2. Master public key and master private key collection process is not secure, because only public channels are available at this stage. 3. Does not provide any security protection for OLSR routing protocol it relies on. 4. KM-SR interdependency cycle |
| 2006 | [60, 49] | A PKI-IBC hybrid key management scheme | Yes | Fixed on cluster head | PKI | Additional overhead for inter-cluster communication. |

Table 3.1: Summary of Master Key and Private Key Generation and Distribution Schemes

| Year | Publication(s) | Main Idea & Contribution(s) | Using unicast routing | Static/ Dynamic | Rounds | problems |
|---|---|---|---|---|---|---|
| 2004 | [7, 8, 6] | A method for computing collisionfree broadcast keys; Use of signatures in broadcast messages. | Yes | Dynamic | 2 | 1. Distribution of the broadcast secret leads to either KM-SR interdependency cycle or scalability problem. 2. Node specific secret is against IBC advantages. |
| 2005 | [92] | Authenticating the exchanged information without online TTP; Resistance to unstable links | No | Static | 2 | In round 1, each node generates an ephemeral key and broadcast it. |
| 2008 | [21, 22] | Subgrouping a 2-party/3-party Key Agreement | Yes | Static | 2 | 1. Each node generates an ephemeral key; 2. Key exchange messages use respective pairwise communication, which requires an existing secure routing mechanism. |
| 2008 | [91] | Set up the group key in one round based on IBBE | No | Static | 1 | 1. Not suitable for dynamic membership. 2. Not integrated with any asymmetric private/public key scheme. |

Table 3.2: Summary of Group Key Agreement Schemes

temporary or ephemeral public keys: One node generates a temporary or ephemeral public key and distributes it to other nodes. Other nodes then need to store it for later use. This process is more similar to the way a certificate-based cryptosystem works. It is inconsistent with the essence of IBC, and offsets the advantages of IBC.

Table 3.3 summarizes the main characteristics and problems of IBC routing protocols in MANETs. As an aside, in the network layer, no cryptography-based routing protocol is immune to denial-of-service (DoS) attacks. The adversary can bring the system down by hijacking packets and garbling messages which leads to receivers consuming limited resources on wastes. We do not consider this as a problem in routing protocols. A routing protocol must satisfy basic security requirements mentioned in Section 2.2. There have been some routing protocols for MANETs in environment with adversary nodes, which do not rely on secure keys, e.g.: Marti *et al* [63] uses a watchdog to monitor behavior of nodes and a pathrater to find routes among nodes trustworthy; Buchegger *et al* proposes CONFIDANT protocol [17] that rewards nodes forwarding packets and punishes nodes not forwarding packets; Michiardi *et al* proposes a reputation mechanism that extends pathrater [66] to more protocols and improves security by disallowing negative rating.

| **Year** | **Protocol(s)** | **Main Contribution(s)** | **Routing protocol based on** | **Require-ments not satisfied** | **problems** |
|---|---|---|---|---|---|
| 2003 | ODSRP [56] | A secure DSR routing protocol using IBC | DSR | Confidentiality, authenticity | 1. Missing a key management scheme. 2. Subject to wormhole attacks. |
| 2005 | LSRP [71] | Concatenated Signature and verification of routing messages in on-demand routing protocols | On-demand routing protocols | Confidentiality | 1. Missing a key management scheme. 2. Message signature is concate-nated and can be large. |
| 2005 | Multi-signature Routing Protocol [82] | A authentication mechanism with aggregation signature | On-demand routing protocols | Confidentiality | 1. Missing a key management scheme. 2. Subject to wormhole attacks. |
| 2005 | A Security Architec-ture to Secure OLSR[2] | The security issues of OLSR, and an architecture including multiple securing mechanisms. | OLSR | Not clear | Details not given. |

Table 3.3: Summary of Secure Routing Schemes

These routing protocols mainly aim at improving routing availability, and do not provide authentication of node's identity, confidentiality, integrity, freshness, and non-repudiation of routing messages, which rely on use of secure keys. To meet all of these requirements, a cryptosystem with a unique private key for each entity is required. However, from this and the previous sections, we can see that many key management schemes assume a secure routing is available; at the same time, many secure routing schemes assume secure keys are already available. This chicken-and-egg-like paradox is noted as SR-KM interdependency cycle problem.

IBC provides many advantages in terms of secure routing. Many simulation works from above publications show that IBC secure routing schemes improve efficiency over counterparts using traditional cryptosystems. However, as many of the above schemes fail to note, we summarize main issues of above proposals:

- Secure routing relies on secure keys that are not available before secure routing is set up.
- Many of the routing protocols are subject to various routing attacks, due to incomplete or flawed encryption/signature schemes.

- Each routing protocol has its own weakness. For example, the above routing protocols based on AODV are all subject to wormhole attacks.

## 3.4 Limitations and Weaknesses from IBC

We have mentioned many properties of IBC which make it especially attractive for MANETs. However, there are still some problems not completely addressed which impedes application of IBC in MANETs. In this section, we will study "key escrow", "identity disclosure", and "identity revocation" problems, and proposals to address them. We deliberately omit those explained in Section 3.1 or Section 3.2.

### 3.4.1 Identity Disclosure

The main advantage of IBC is that the public key of an entity is its identity that is piggybacked and explicit in the message. This leads to the problem of identity exposure — the identity of any node is exposed to all others. In some MANET systems, this is not desirable, e.g. for those used in battlefield, this may expose the identity of a commander to the enemy, which then enables traffic analysis and incurs great danger.

Special characteristics of MANETs lend them many security and privacy concerns. One concern is traffic analysis. By definition, it is a passive attack such that an adversary observes network traffic and infers sensitive information of the applications and/or the underlying system, for example, sensitive information about the communicating entities [41]. The information could be related to the identities of the communicating parties, or to the network traffic patterns or even to the changes in the traffic pattern. Both packet contents and header fields can reveal the information of packet sources and destinations. In wireless environments, the adversaries can easily capture transmitted packets and conduct traffic analysis. The shared wireless medium introduces opportunities for passive eavesdropping on data communications. Thus traffic analysis is one of the most subtle and unsolved security attacks against MANETs.

To prevent traffic analysis, anonymity is required in the communication. Pfitzmann and Hansen [73] defined the anonymity as the state of being not identifiable within a set of subjects, that is, the anonymity set. The anonymity set is the set of all possible acting subjects such as human beings, legal persons or computers.

**MASK for Anonymous Communications**

Zhang *et al* propose an IBC anonymous communication scheme in MANETs [94]. The authors identify the problem of malicious traffic analysis in MANETs due to the broadcast nature of radio transmission, and propose an anonymous on-demand routing protocol termed MASK. Derived from work of [10, 4], the protocol enables anonymous communications by allowing neighboring nodes to authenticate each other without revealing their identities.

The PKG pre-calculates a large set of collision-resistant pseudonyms and a corresponding secret point set. During the bootstrapping phase, a TA distributes system public parameters. Moreover, the TA furnishes each node $ID_i$ with a sufficiently large set $PS_i$ of collision-resistant pseudonyms and a corresponding secret point set. No one but the PKG can link a given pseudonyms to a particular node or identity, or deduce the corresponding secret point with non-negligible probability. Using $PS_i$ and nonces $n_1, n_2$, $A$ and $B$ can calculate $\gamma$ pairs of shared session key ($SKey$) and link identifier ($LinkID$) as: $K^{\gamma}_{AB} = H_2(K_{AB}||n1||n2||2\cdot\gamma)$, $L^{\gamma}_{AB} = H_2(K_{AB}||n1||n2||2\cdot\gamma+1)$ (see [94, p. 1943] for details). Such $\langle SKey, LinkID\rangle$ pairs are unique due to collision-resistant hash functions $H_1$ and $H_2$. The $LinkID$s will be used to identify the packets transmitted between $A$ and $B$ and the $SKey$ can be used to encrypt, integrity-protect, or authenticate the content of the packets if needed.

Based on this anonymous neighborhood authentication scheme, the authors propose an improved AODV routing protocol which enables communication between nodes without disclosing the real identity of the node.

The authors evaluate the computation costs of the critical cryptographic operations in their scheme. In this implementation, the routing information is not authenticated, they plan to combine MASK with other secure routing schemes to provide an anonymous yet secure routing protocol.

Problems of this scheme are: First, each node maintains a large set of pseudonyms and the corresponding private keys for each pseudonym. This is resource consuming, and against advantages of IBC. Second, it can only be used in their own routing protocol and not in any other protocol or any higher layer application, because it uses link identifier to transport packets among nodes without using real identities, but there seems no way to convert link identifier's back to identities.

### 3.4.2 Key Revocation Difficulty

Due to the weak physical protection of nodes, node compromises including key disclosures are very likely in MANETs. Meanwhile, the infrastructure for certificate or public key revocation does not exist in MANETs. Frequent key renewals to prevent such compromises are either computationally challenging in solution with distributed on-line key generation or infeasible in solutions with off-line key generation.

#### Appending "Expire Time" to the Identity

In the very beginning of IBC, the public key of node ID was computed as $D_{ID} = H(ID||ExpireTime)$ to allow identity revocation [10, 28, 29, 52].

Hoeper *et al* [47] propose a scheme for key revocation and key renewal using an IBC scheme in MANET. This work is based on their former work in [46], and the work of [26, 62]. To enable key renewal in IBC schemes, they introduce a new format for ID-based public keys: $D_{ID} = H(ID||t_i||v_i)$, where $t_i$ denotes the expiration date, and $v_i$ is the version number. The version number always starts with $1$ for every new expiry date and is incremented with each key renewal for the same date.

New keys can be issued for the same identity after the previous key has been revoked. And new nodes that join the network can learn about past accusations and revocations. Upon receiving a new key pair and re-joining the network, a node only needs to broadcast its new public key to $m$-hop neighborhood. The receivers update the version number in their revocation lists accordingly and set all accusation values for this node to zero. The level of security can be chosen as performance trade-off.

These proposals append extra information to an identity to generate a public key. This seemingly tiny change in public keys leads to some complications in MANETs: It was first intended for Internet applications where arbitrary identities are accepted, e.g. email services, and works well there. As in the network layer of MANETs where identities are usually fixed, such as MAC addresses or IP addresses, the identity in a packet can no longer be an arbitrary string, and there is no separate field for an extra identity. Furthermore, this scheme requires precise synchronization among all network nodes, which is difficult to achieve in a MANET environment.

#### Key-update Parameters

Zhang *et al* [95] propose to use key-update parameters to revoke voided public and private keys, using a variant of the self-healing group key distribution scheme by Liu *et al* [61].

The key generation was explained in Section 3.1.1.

Before network deployment, key-update parameters: $\langle\{v_i(x), l_i(ID)\}_{i=1,...,m}\rangle$, where $m$ is the maximum possible phase index, are distributed to all nodes. The PKG generates private keys $d_i$ for a node for all phases $i = 1, ..., m$ (strictly speaking, the phase-specific components of private keys which are then combined with node-specific components to generate a node's private key). The PKG calculates the differences $v_i$ between $d_i$ series and a polynomial series $u_i$, and distribute the difference series $v_i$ to all nodes. At a later phase, online D-PKGs only provide the $u_i$ series to unrevoked nodes. In this way, revoked nodes cannot update their private keys.

Key-update parameters are generated in this way: the PKG picks $m$ distinct $2t^c$-degree polynomials, denoted by $\{l_i(x) = \sum_{j=0}^{2t^c} l_{i,j}x^j(mod\ q)\}_{i=1,...,m}$ with $l_{i,j} \in \mathbb{Z}_q^*$, and $m$ distinct $t^c$-degree polynomials, denoted by $\{u_i(x) = \sum_{j=0}^{t^c} u_{i,j}x^j(mod\ q)\}_{i=1,...,m}$ with $u_{i,j} \in \mathbb{Z}_q^*$. The PKG then constructs $\{v_i(x) = d_{iy} - u_i(x)\}_{i=1,...,m}$, where $d_{iy}$ denotes $y$-coordinate of the elliptical curve point $d_i$ represents.

At phase $i$, a D-PKG node, say $Z$, collects secret shares and generates private key $d_i$. $Z$ broadcasts the following message: $B_i := \{ID_X\}_{X\in\Lambda} \bigcup \{U_j(x) = \xi_j(x)u_j(x) + l_j(x)\}_{j=1,...i}$, where $\Lambda$ denotes the set of nodes revoked until phase $i$, $\xi_j(x) = \prod_{x\in\Lambda}(x - ID_X)$. An unrevoked node $B$ can derive $U_i(ID) = \xi_i(ID_B)u_i(ID_B)+l_i(ID_B)$, and then get $u_i(ID_B) = \frac{U_i(ID_B)-l_i(ID_B)}{\xi_i(ID_B)}$ and then $d_{iy} = v_i(ID_B) + u_i(ID_B)$, while a revoked one $X$ cannot get $u_i(ID_X)$ because $\xi_i(ID_X) = 0$.

Though this scheme is novel and sound, there exists a possible drawback: The scheme does not have good scalability, since the phase-specific components of all phases need to be calculated before network deployment to get key-update parameters and furnish all nodes with them, and size of parameters to be distributed to D-PKGs is also proportional to number of D-PKGs.

### 3.4.3 Key Escrow

Key escrow is inherent in IBC. The PKG or the TA that generates private keys for nodes knows the private key of each node and can eavesdrop the traffic or impersonate it. Although it may be a desirable feature in some cases (e.g. in military hierarchy), it is a problem with some MANETs.

Solutions for key escrow problem in general IBC include:

- Using additional private/public key pairs [38]. This solution is not pure IBC scheme, and is against advantages of IBC.

- Using threshold cryptography to distribute the secret key to multiple nodes [10, 54, 15, 20, 67, 72]. We have mentioned problems incurred by threshold cryptography in Section 3.1.

- Key Exchange Protocols without Key-escrow: Hoeper and Gong [45] propose a set of key exchange protocols without key-escrow, based on the work of [10, 44].

  In these protocols, a TTP computes the private key for each node using a master key and node's public key $Q_{ID}$, and distributes the key over a secure channel during network initialization. After initialization, the TTP is not needed, and any two nodes share a pairwise secret key. To provide forward security and prevent the TTP from being a key escrow, the authors propose some protocols. A basic form of these protocols is: First, $K_{AB}$ is divided into two parts $K_e$ and $K_a$. Encryption under $K_e$ prevents all other nodes from reading the messages, whereas $K_a$ is used in a message authentication code (MAC) to enable mutual authentication. Then, Each of $A$ and $B$ chooses an arbitrary key $K_1$ and $K_2$ separately and exchanges them using $K_a$ and $K_e$. A shared session key can be set up as $K_{ses} = f(K_1, K_2)$. By replacing $K_1$ and $K_2$ with different forms, different properties can be obtained.

  The scheme is not applicable to routing protocols, because it assumes secure routing is ready.

## 3.5 Summary of the Chapter

This chapter reviewed previous schemes applying IBC to MANETs, and identified main issues of them: key management and secure routing interdependency cycle; use of temporary or ephemeral keys which degrades the scheme to a CBC-like system; vulnerabilities to certain attacks; extra overhead and insecurity of broadcasting. This chapter also listed some features of IBC itself that are considered as limitations and weaknesses in MANET context. We next will analyze these problems and propose a security framework that addresses these problems.

# Chapter 4

# A Novel Key Management and Secure Routing Integrated Framework

In the previous chapter, we identified the main issue on applying IBC to MANETs—key management and secure routing interdependency cycle. In this chapter, we analyze this issue and propose our solution to it—a key management and secure routing integrated framework. Other issues will be discussed in later chapters.

## 4.1 Basic Idea and Overview of the Framework

Secure routing is the bedrock of a secure MANET. The requirements mentioned in Section 2.2 are the basic requirements for a secure routing. To meet these requirements, a key management scheme is needed (confidentiality is not very important for routing messages, but is important for key management messages). On highest level, key management schemes can be classified into two categories. The first category makes use of prior security context distributed before network starts. The second category does not depend on any prior shared context, and is self-organized. We do not consider the second category of key management schemes capable for secure routing, because there is no way to meet the requirements for secure routing mentioned above unless it is guaranteed that no adversary node would participate in routing setup. For example, we cannot verify if the identity presented by a node really belongs to itself, or if a single node uses multiple identities. Thus we only consider the first category in our scheme. In this category, we can distribute security context in the form of symmetric keys or asymmetric keys. The latter includes Certificate-based Cryptography (CBC), and Identity-based Cryptography (IBC).

Symmetric key solutions are widely used in sensor networks. One reason for this fact is that in most sensor networks there are base stations available which simplifies symmetric key management, but base stations are not always available for general MANETs. Secure pairwise communication with symmetric keys requires too many keys to be distributed and stored on nodes. In a traditional symmetric key management scheme, if there are $n$ nodes in a network, each node needs to store $n-1$ keys and a total of $n(n-1)/2$ keys need to be generated and distributed in the network. Symetric key scheme can only provide pairwise authentication and non-repudiation, and does not support network wide authentication and non-repudiation, because each key is shared by a pair of nodes at least.

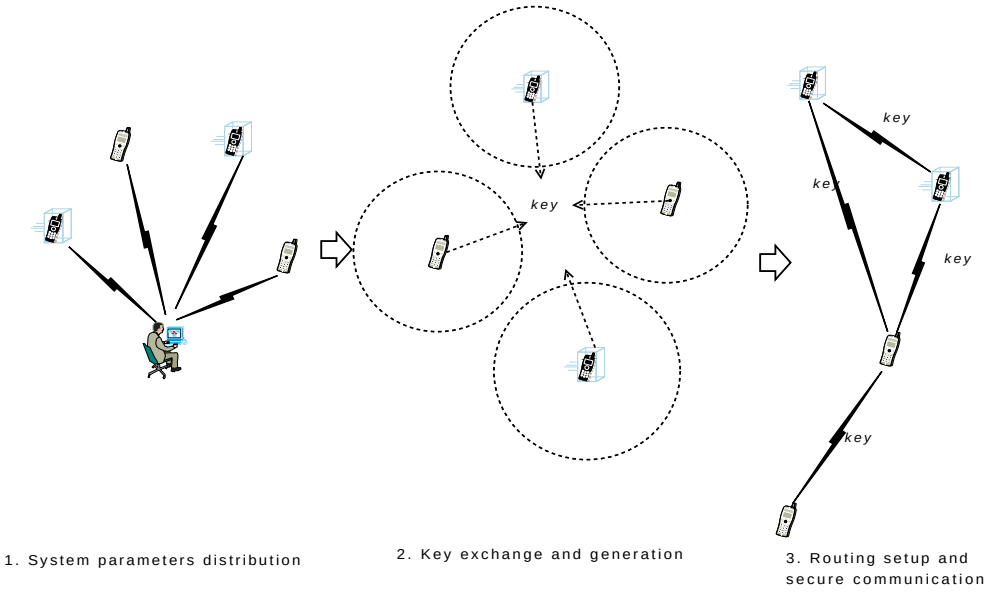
To meet the requirements of secure routing, a cryptosystem with a unique private key for each entity is required. However, many key management schemes assume a secure routing is available; at the same time, many secure routing schemes assume secure keys are already available. For example, Zhou *et al* [99] propose a distributed CA architecture that can be used in a CBC. The distributed CA can sign private keys of nodes in a distributed fashion. Many IBC schemes generate private keys in the same approach using distributed Private Key Generator (PKG) nodes. These schemes rely on some existing routing or online administration mechanisms (e.g. out-of-band communicant, side channel) to distribute secret shares among the distributed PKG nodes. Thus, they cannot be used to set up secure routing that would require secure keys. This is noted as *KM-SR interdependency cycle problem* (Chapter 3).

We summarize main features and drawbacks of symmetric cryptography, CBC and traditional IBC in Table 4.1

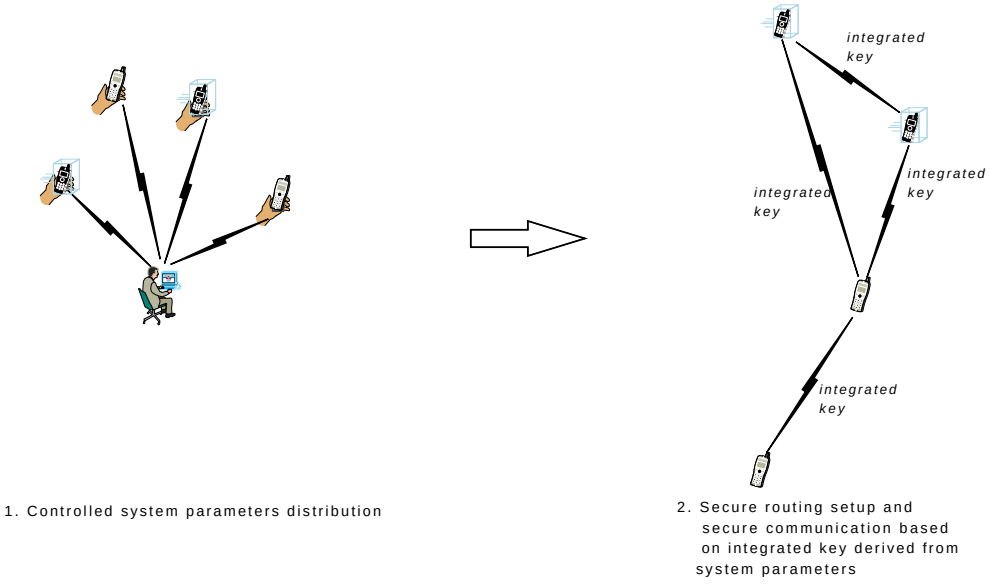
From the above table, we can see that if we can remove KM-SR interdependency cycle from IBC, it would be the best solution for key management and secure routing.

We consider using IBC in a secure routing scheme because IBC has the following advantages in secure routing which has already been noted in the literature:

- IBC eases the process of key distribution. Key exchange messages can be spared. Pairwise keys are available with only a few security parameters distributed at the network deployment phase, which is not possible with traditional symmetric key or CBC cryptosystems. The sender and receiver share a default pairwise key $K_{AB} = \hat{e}(d_A, Q_B) = \hat{e}(d_B, Q_A) = K_{BA}$ without any extra distribution and storage of keys. This is critical to routing protocols because until routing is set up there seems no way to distribute or negotiate secret keys among nodes. Traditional symmetric or asymmetric cryptography requires a large amount of keys to distribute and store.




(a) Previous KM-SR scheme




(b) Proposed KM-SR Integrated Framework

Figure 4.1: Comparison between Previous and Proposed KM-SR Integrated Framework

| **KM scheme** | **Number of keys to store per node** | **Security Features** | **KM-SR Interdependency** |
|---|---|---|---|
| Symmetric Key Cryptography | $O(n)$ ($n$: the number of nodes in the network) | Confidentiality, integrity | No |
| CBC | $O(n)$ ($n$: the number of nodes in the network) | Confidentiality, integrity, authentication, non-repudiation | Yes |
| Traditional IBC | Constant | Confidentiality, integrity, authentication, non-repudiation | Yes |
| Our goal | Constant | Confidentiality, integrity, authentication, non-repudiation | No |

Table 4.1: Features and Drawbacks of Symmetric Cryptography, CBC and Traditional IBC

- IBC improves efficiency of secure routing. Once secure keys are avaiable, IBC can be applied to either on-demand routing protocols like Dynamic Source Routing (DSR) and Ad-hoc On-Demand Distance Vector (AODV) Routing, or link state routing protocols like Optimized Link State Routing (OLSR). The routing messages are encrypted and signed by the sender and decrypted and verified by the receiver using IBC. To protect routing messages, on same security level, IBC encryption/decryption is faster, and IBC signature/verification is shorter.

Nevertheless, we note that the first advantage does not help in initial routing setup. First, in many existing proposals, pairwise keys are available only after initial routing is set up. Second, even if pairwise keys are available at initial routing setup phase, they still have no use then because pairwise routing is not available. It is only useful after routing is set up, e.g. for routing update.

Here we see a gap between key management schemes and secure routing schemes, the reason is that the first phase of routing setup must definitely be broadcasting, as is done in every routing protocol. A secure broadcasting scheme is needed for initial routing setup. This is an insight into the KM-SR interdependency cycle problem.

The solution to break the KM-SR interdependency cycle problem needs to have the following properties as prerequisites:

- A key management scheme not relying on secure routing, as suggested by Hegland *et al* in [43]. This is required because secure routing should not be working without secure keys.
- A secure broadcasting scheme that meets all the security requirements listed in Section 2.2. This is required because the first stage of a secure routing is secure broadcasting.

We here propose a KM-SR integrated framework. The kernel of this framework is a key management integrated routing protocol which breaks KM-SR interdependency cycle. The design of this protocol is based on these notions:

- Key management should not rely on secure routing.
- Secure keys should be available before a routing protocol starts working.
- Secure routing starts from secure broadcasting.
- To prevent routing attacks, a routing protocol must encrypt and authenticate every message and packet, not only end-to-end, but also hop-by-hop.
- Some routing protocols have security or efficiency weaknesses.

The Key Management Integrated Routing Protocol starts with a trusted and protected network. With the secret system parameters, the nodes communicate with each other securely and set up routing table. The only way of communication before routing setup is broadcasting. The scheme utilizes system parameters of IBC to derive node-specific 1-to-$m$ broadcast keys. These node-specific 1-to-$m$ broadcast keys are used to broadcast routing messages to all neighbors of a node or all other nodes in the network. The routing protocol decides the destinations of the routing messages. The node-specific 1-to-$m$ broadcast keys are essential for secure routing. 1-to-1 keys cannot be used in routing protocols, because there is no routing between any two nodes. Group-shared $m$-to-$m$ keys are not secure enough, because there is no authentication and non-repudiation, and is especially vulnerable to compromise because one compromised key reveals all encrypted messages for that group.

Based on limited distribution of system parameters and the integrated 1-to-$m$ broadcast keys, a secure routing can be set up. In this scheme, a 1-to-$m$ broadcast key is available for each node at the routing setup phase thanks to good features of IBC. Combining the use of 1-to-$m$ broadcast keys and private keys provides confidentiality, integrity, authentication and non-repudiation at routing setup phase. Compared to previous IBC solutions, this proposal can setup secure routing without the KM-SR interdependency cycle, while the amount of data distributed remains almost the same. In this scheme, secure keys are available before a routing protocol starts working, so that a secure routing that meets the requirements mentioned in Section 2.2 can be set up. To prevent routing attacks, such as wormhole and blackhole, the routing protocol encrypts and authenticates every message and packet, not only end-to-end, but also hop-by-hop.

Figure 4.1 illustrates the basic idea of the proposed scheme and the difference between the previous schemes and the proposed scheme. In previous schemes, there are three steps to set up routing and we see four problems in these schemes:

- Interdependency cycle between step 2 and 3.
- There is no protection in secure key setup messages, thus generated keys are not guaranteed secure.
- The system is subject to potential insider attacks, because initial nodes are not authenticated.
- The system is subject to mobile attacks, and can be taken over by the adversary [97].

The key point here is integrated key generation. There is no explicit key exchange messages or key generation phase. Step 2 of previous solutions is total unnecessary. And secure routing is ready to start immediately after system parameter distribution, and all these 4 problems are addressed in the scheme. The only new requirement is more control in secret distribution phase to authenticate participating nodes and secure the distribution. We think it is worthwhile to do a little work at beginning instead of doing a lot later, as an old saying goes— “Well begun is half done”, let alone for many practical MANETs, this is not an extra work.

## 4.2 Secure Key Generation and Secure Routing Setup

The KM-SR integrated scheme comprises a secure routing protocol and integrated secure keys. Secure routing protocol can be based on any standard routing protocol. In our work,

we choose to use OLSR protocol. The reasons for choosing OLSR are:

- First, it has high routing efficiency and low traffic overhead by utilizing multipoint relay (MPR). The core optimization of OLSR is the flooding mechanism for distributing link state information, which is broadcast in the network by selected MPR nodes. As a further optimization, only partial link state is diffused in the network. The OLSR backbone for message flooding is composed of MPRs. Each node selects its MPRs from its symmetric 1-hop neighbor nodes such that a message emitted by a node and repeated by the MPR nodes will be received by all 2-hop neighbors. As a result, in order to achieve a network-wide broadcast, a broadcast transmission needs only to be repeated by just a subset of the neighbors of a node—this subset constitutes the MPR set of the node [74].

- Second, it has good extensibility in packet and message format. OLSR allows for the encapsulation of numerous independent messages as extensions within a single OLSR packet. Each message contains a common header which allows for neighbouring nodes to correctly accept and retransmit the message back into the network. Messages are sent into the entire network by using a MPR scheme, and local nodes prevent duplicate retransmissions of previously processed messages through the use of duplicate tables. Control traffic in OLSR is exchanged through two different types of messages: *HELLO* and *TC* (Topology Control) messages. *HELLO* messages are exchanged periodically among neighbor nodes, in order to detect links to neighbors and to signal MPR selection. *TC* messages are periodically flooded to the entire network, in order to diffuse link state information to all nodes. For both *HELLO* and *TC* messages, it is easy to add new fields (such as signature) in them, and it is also easy to add new fields (such as signature) or even new messages (we do not use this feature in this scheme) to an OLSR packet [23].

Secure keys comprise one system public key, one private key per node, one 1-to-$m$ broadcast key per node, and one pairwise key per pair—$n(n-1)/2$ keys in total. All these keys are derived from IBC system parameters without any other overhead or any interaction between nodes. And each node only needs to store a limited number of secure parameters.

An off-line system administrator is required for the system setup. The system parameters are the same as in Boneh-Fraklin's IBC scheme [10]: Let $\mathbb{G}_1, \mathbb{G}_2$ be two cyclic groups of order $q$ for some large prime $q$, where $\mathbb{G}_1$ is the group of points of an elliptic curve over $\mathbb{F}_p$ and $\mathbb{G}_2$ is a subgroup of $\mathbb{F}^*_{p^2}$, and $\mathbb{F}_p$ is the finite field with prime $p$

elements. $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a symmetric bilinear map between $\mathbb{G}_1$ and $\mathbb{G}_2$. The off-line administrator sets the system public key $P_{pub}$ as $sP$ where $s$ is a random number in $\mathbb{Z}_q^*$, and $P$ is an arbitrary point in $E/\mathbb{F}_p$ of order $q$. The system parameters are $params = \langle p, q, n, P, P_{pub}, H, H_0, H_1 \rangle$, where $H : \{0,1\}^* \rightarrow Z_q^*$ is a random oracle hash function for message signature, $H_0 : \mathbb{F}_{p^2} \rightarrow \{0,1\}^m$ is a random oracle hash function for symmetric key generation, and $H_1 : \{0,1\}^* \rightarrow \mathbb{F}_p$ is a random oracle hash function for mapping an identity string to a point on $E/\mathbb{F}_p$. For a given string $ID \in \{0,1\}^*$, the off-line administrator generates the private key $d_{ID}$ as $d_{ID} = sQ_{ID}$, where $Q_{ID} = H_1(ID)$ is a point in $E/\mathbb{F}_p$ mapped from $ID$, and functions as the public key of $ID$. Every node gets the system parameters and its private key from the administrator before the network starts up. This can be achieved by gathering authentic nodes and distributing secret securely right before deploying, for example, by face-to-face communication, infra-red, or Radio Frequency (RF) communication in a small and protected area.

After deployment, nodes start to communicate with each other securely based on furnished secrets. The only means of communication now is broadcasting. A node broadcasts routing messages protected by a node-specific 1-to-$m$ broadcast key and its private key. For this purpose, most existing cryptosystems cannot be used, because they are for 1-to-1 communication, i.e. sending a message to a specific recipient. In our case, we need to send secure routing messages to multiple recipients, in a way that meets all the security requirements listed in Section 2.2. We propose a encryption/decryption and signature/verification cryptosystem for this 1-to-$m$ broadcasting scheme. The cryptosystem is based on work of Boneh *et al* [13] and Cha *et al* [18].

A node $A$ generates a 1-to-$m$ broadcast key in this way:

$A$ computes

$$g = \hat{e}(d_A, P)$$
$$k = H_0(g^r),$$

where $r \in_R \mathbb{Z}_q^*$ and $rQ_A \neq \infty$ (point at infinity).

Other nodes can compute the key as follows:

$$\begin{aligned} g^r &= \hat{e}(Q_A, P_{pub})^r \\ &= \hat{e}(rQ_A, P_{pub}) \text{ and} \\ k &= H_0(g^r), \end{aligned}$$

where $rQ_A$ is attached in the message.

When node $A$ sends a routing message, $A$ encrypts and signs the message as follows:

- Uses $k$ as encryption key in a symmetric encryption/decryption method such as *AES* to encrypt a message $M$, and gets $M'$.
- Node $A$ signs an encrypted message $M'$:
  - Calculates $h = H(M')$.
  - Calculates signature $\sigma = (h + r)d_A$, where $r$ is the same as what was used in encryption.
  - Sets $\langle M', \sigma, rQ_A \rangle$ in the message field of routing message

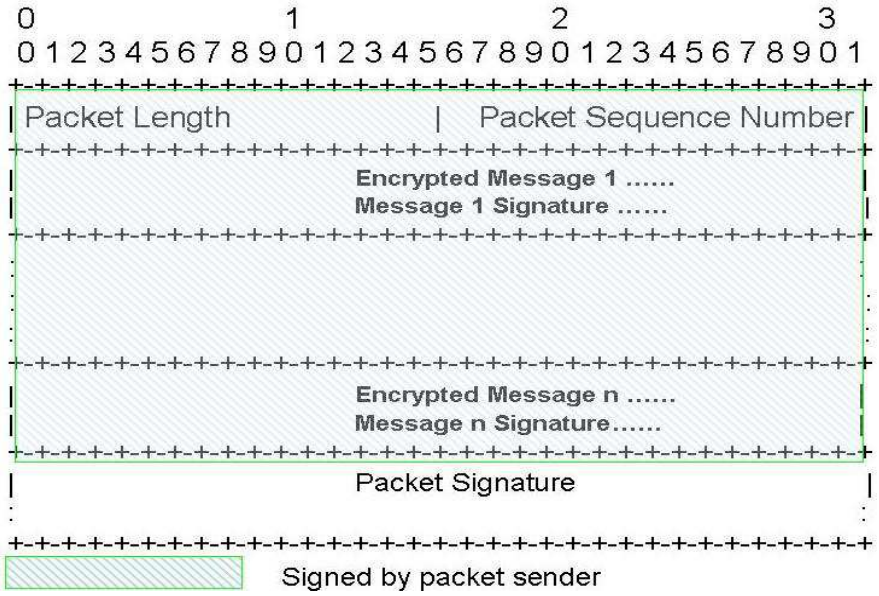
Node $A$ then encapsulates messages in a packet and signs the packet as what is to be described shortly and broadcasts it to other nodes. The signature of a packet is attached in the packet. Figure 4.2(a) illustrates an OLSR packet with packet header, signature, and encrypted and signed messages. Figure 4.2(b) illustrates an OLSR message with message header, signature, and encrypted and signed message content.

In OLSR, after a routing message packet is broadcasted, each node that has received it disassembles it and reassembles a new packet if there exists any message to be forwarded, for example the TC message, and may need to modify some mutable fields, such as *Time To Live* (TTL) and *Hop Count* (HC) fields. Therefore, we need a packet signed by its sender, either the originator or the forwarder of messages. The signature of a packet is generated using the private key of the packet sender, for example $B$ signs a packet $\mathcal{P}$ as follows:

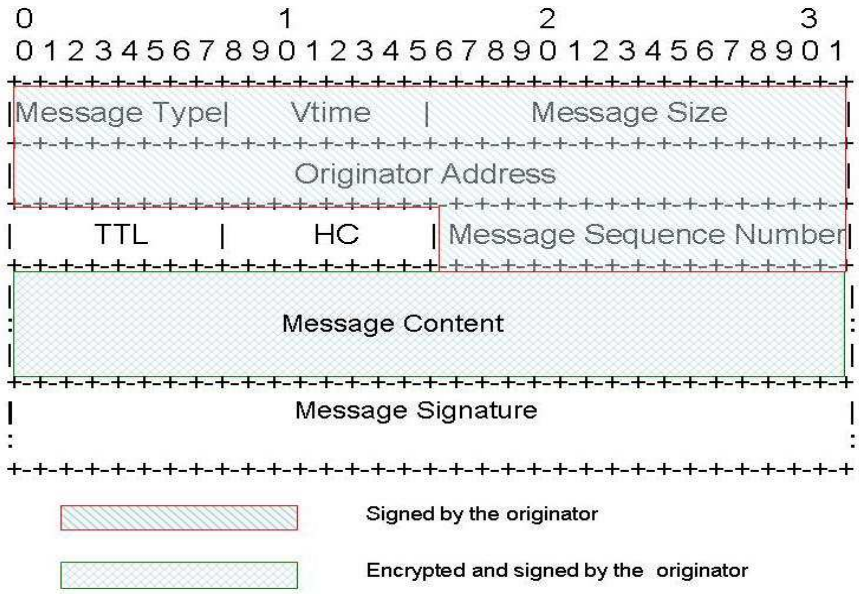
- Calculates $h = H(\mathcal{P})$.
- Calculates signature $\sigma = (h + r)d_B$, where $r \in_R \mathbb{Z}_q^*$ and $rQ_B \neq \infty$.
- Puts $\langle \sigma, rQ_B \rangle$ in the packet signature field of the packet.

In our scheme, when a node receives a packet $\mathcal{P}'$ from node $B$, it verifies the signature in the way described below:

- Calculate $h = H(\mathcal{P}')$, and $hQ_B$.
- Calculate if $\hat{e}(P, \sigma) = \hat{e}(P, (h + r)d_B) = \hat{e}(P_{pub}, (hQ_B + rQ_B))$ holds.




(a) A protected OLSR packet




(b) A protected OLSR message

Figure 4.2: Protected OLSR Packet and Message

If the signature is valid, it further processes messages in the packet. It checks message header, and accepts messages destined to it, and re-broadcasts messages to others if any.

For an accepted message $\langle M', \sigma, rQ_A \rangle$ from node $A$, a node verifies and decrypts the message as follows:

- Verifies the signature:
  - Calculate $h = H(M')$, and $hQ_A$.
  - Calculate if $\hat{e}(P, \sigma) = \hat{e}(P, (h + r)d_A) = \hat{e}(P_{pub}, (hQ_A + rQ_A))$ holds.
- If the signature of a message is valid, it calculates the 1-to-$m$ broadcast key $k$ of the message originator as explained earlier, and processes the message.

With the authenticated routing messages, the nodes in the network can set up a routing table among them using the shortest path algorithm as specified in standard OLSR protocol.

## 4.3 Summary of the Chapter

In this chapter, we proposed a novel IBC framework with key management and secure routing integrated scheme that addresses interdependency between these two aspects. The scheme starts with a constant number of initial secrets to be furnished on participating nodes. From the initial secrets, each node derives a 1-to-$m$ broadcast key. Nodes can then broadcast routing messages that are protected by the 1-to-$m$ broadcast key and its specific private key. This is a pure IBC scheme with no key management and secure routing interdependency cycle. In the next few chapters, we will analyze security features and performance of the framework, and enhancements against various attacks, limitations and weaknesses of IBC itself.

# Chapter 5

# Security Analysis and Enhancements of the Framework

In this chapter, we prove the security of the scheme theoretically, and analyze security features of this framework. Enhancements against various attacks are also discussed in this chapter.

## 5.1 Security features and proof

Security of the proposed scheme is based on assumptions well-established and theorems proved in this paper.

**Assumption 1** *System parameters are distributed only to authentic nodes of the network and kept secret to adversaries at least until routing is set up.*

**Assumption 2** *AES cryptosystem used in this scheme is hard enough, so that an adversary cannot break the system and learn the plaintext if s/he does not know the key.*

**Assumption 3** *Static Diffie-Hellman (SDH) problem is hard in group $\mathbb{G}_1$, i.e.: Given $(P, Q, aP)$ for $P, Q \in \mathbb{G}_1$ and for some $a \in Z_q^*$, there is no efficient algorithm to compute $aQ$.*

**Theorem 1** *Suppose Assumption 3 is true and the hash functions are random oracles. The signature scheme in Section 4.2 is secure against existential forgery under an adaptive chosen-message attack.*

***Proof***. We prove the security of the signature scheme against existential forgery under adaptive chosen-message attacks in the random oracle model. Existential unforgeability under a chosen-message attack [40] for a signature scheme (KeyGen, Sign, and Verify) is

defined using the following game between a challenger and an adversary $\mathcal{A}$:

- **Setup.** The challenger runs algorithm KeyGen to obtain a public key PK and private key SK. The adversary $\mathcal{A}$ is given PK.
- **Queries.** Proceeding adaptively, $\mathcal{A}$ requests signatures with PK on at most $q_S$ messages of his/her choice $M_1$, ..., $M_{q_s} \in \{0,1\}^*$. The challenger responds to each query with a signature $\sigma_i = Sign(SK, M_i)$.
- **Output.** Eventually, $\mathcal{A}$ outputs a tuple $\langle M, \sigma, r \cdot PK \rangle$ where $r$ is a random number, and wins the game if
  1. $M$ is not any of $M_1$, ..., $M_{q_s}$.
  2. Verify($\langle M, \sigma, r \cdot PK \rangle$) = valid.

We define $AdvSig_{\mathcal{A}}$ to be the probability that $\mathcal{A}$ wins in the above game, taken over the coin tosses of KeyGen and of $\mathcal{A}$.

We prove this using *proof by contradiction* method. Suppose $\mathcal{A}$ is a forger algorithm that breaks the signature scheme with running time $\tau$ and $AdvSig_{\mathcal{A}} = \varepsilon$. We show how to construct an algorithm $\mathcal{B}$ that solves SDH in $\mathbb{G}_1$ with $AdvSig_{\mathcal{B}}$ at least $\varepsilon'$ and running time at most $\tau'$, for all $\varepsilon'$ and $\tau'$ satisfying: $\varepsilon' \geq \frac{\varepsilon}{e^2(q_s+1)^2}$ and $\tau' \leq \tau + c_1(q_H + 3q_S)$, where $c_1$ is a constant, $e$ is the base of the natural logarithm, $q_H$ is the number of queries $\mathcal{A}$ made to the hash function $H$, and $q_S$ is the number of queries $\mathcal{A}$ made for signature. This will contradict the fact that $\mathbb{G}_1$ is a SDH group which is given by Assumption 3.

Let $P$ be a generator of $\mathbb{G}_1$, $s \in_R Z_q^*$ is the master key kept secret to $\mathcal{A}$ and $\mathcal{B}$. Algorithm $\mathcal{B}$ is given $P, sP \in \mathbb{G}_1$. $\mathcal{B}$'s goal is to output $sQ'_A \in \mathbb{G}_1$ given a random point $Q'_A = r_A Q_A$ in $\mathbb{G}_1$ (without knowing $r_A$).

**Setup.** Algorithm $\mathcal{B}$ starts by giving $\mathcal{A}$ $P$ and system public key $(s+r')P = sP + r'P$ where $r' \in_R Z_q^*$.

$H$**-queries.** When $\mathcal{A}$ queries the oracle $H$ at a point $M_i \in \{0,1\}^*$, algorithm $\mathcal{B}$ responds as follows:

1. If the query already appears on the $H$-list in a tuple $\langle M_i, w_i, b_i, c_i \rangle$ then algorithm $\mathcal{B}$ responds with $H(M_i) = w_i \in Z_q^*$
2. Otherwise, $\mathcal{B}$ generates a random coin $c_i \in \{0,1\}$ so that $Pr[c_i = 0] = 1/(q_S + 1)$.
3. Algorithm $\mathcal{B}$ picks a random $b_i \in Z_q^*$ and computes $w_i \leftarrow (1 - c_i) + b_i \in Z_q^*$

4. Algorithm $\mathcal{B}$ adds the tuple $\langle M_i, w_i, b_i, c_i \rangle$ to the $H$-list and responds to $\mathcal{A}$ by setting $H(M_i) = w_i$.

**Identity-mapping-queries.** For any given identity $ID_i \in \{0,1\}^*$, Algorithm $\mathcal{B}$ uses the random oracle $H$ to produce the corresponding point $Q_i$ on $\mathbb{G}_1$:

1. Algorithm $\mathcal{B}$ runs $H$-queries to obtain a $v_i \in Z_q^*$ such that $H(ID_i) = v_i$. Let $\langle ID_i, v_i, j_i, k_i \rangle$ be the corresponding tuple on the $H$-list. If $k_i = 0$ then $\mathcal{B}$ reports failure and terminates.

2. Otherwise, we know $k_i = 1$ and hence $v_i = (1 - k_i) + j_i = j_i \in Z_q^*$. Define $Q_i = v_i P$.

3. Output $Q_i$.

**Signature-queries.** Let $M_i$ be a signature query issued by $\mathcal{A}$ with identity $ID_i$. Algorithm $\mathcal{B}$ responds to this query as follows:

1. Algorithm $\mathcal{B}$ runs Identity-mapping-queries to obtain a $Q_i \in \mathbb{G}_1$. Let $\langle ID_i, v_i, j_i, k_i \rangle$ be the corresponding tuple on the $H$-list. We have $Q_i = j_i P$.

2. Algorithm $\mathcal{B}$ runs $H$-queries to obtain a $w_i \in \mathbb{G}_1$ such that $H(M_i) = w_i$. Let $\langle M_i, w_i, b_i, c_i \rangle$ be the corresponding tuple on the $H$-list. If $c_i = 0$ then $\mathcal{B}$ reports failure and terminates.

3. Otherwise, we know $k_i = 1$ and hence $v_i = j_i \in Z_q^*$, and $c_i = 1$ and hence $w_i = b_i \in Z_q^*$.

4. Define $\sigma_i = (s + r')(w_i Q_i + r_i Q_i)$, where $r_i \in_R Z_q^*$. Algorithm $\mathcal{B}$ generates $r_i$ and calculates $\sigma_i$:
   $\because Q_i = v_i P, \therefore \sigma_i = (s + r')(w_i Q_i + r_i Q_i) = (s + r')(w_i + r_i)(v_i P) = (w_i + r_i)(v_i s P) + (w_i + r_i)(v_i r' P)$.

   Observe that $\hat{e}(P, (s+r')(w_i Q_i + r_i Q_i)) = \hat{e}((s+r')P, (w_i Q_i + r_i Q_i))$ and therefore $\sigma_i$ is a valid signature on $M_i$ under the public key $(s + r')P$ for $Q_i$.

5. Algorithm $\mathcal{B}$ gives message signature tuple $\langle M_i, \sigma_i, r_i Q_i \rangle$ to algorithm $\mathcal{A}$.

**Output.** Eventually algorithm $\mathcal{A}$ produces a message signature tuple $\langle M_A, \sigma_A, r_A Q_A \rangle$ for $ID_A$ such that no signature query and identity-mapping-query were issued for $M_A$ and $ID_A$ (as we supposed existential forgery is possible at the beginning of the proof).

If there is no tuple on the $H$-list containing $M_A$, then $\mathcal{B}$ issues a signature query itself for $H(M_A)$ to ensure that such a tuple exists. If there is no tuple on the $H$-list containing $ID_A$, then $\mathcal{B}$ issues a signature query itself for $H(ID_A)$ to ensure that such a tuple exists.

We assume $\sigma_A$ is a valid signature on $M_A$ for $ID_A$ under the given public key. If it is not, $\mathcal{B}$ reports failure and terminates.

Next, algorithm $\mathcal{B}$ finds the tuple $\langle M_A, w, b, c\rangle$ and $\langle ID_A, v, j, k\rangle$ on the $H$-list.

If $c = 1$ or $k = 1$, $\mathcal{B}$ reports failure and terminates.

Otherwise $c = 0$ and $k = 0$, and therefore $w = H(M_A) = 1 + b$ and $v = H(ID_A) = 1 + j$.

Hence, according to definition of $\sigma$, after step 3 of the signature query, $\mathcal{B}$ knows $\sigma_A = (s + r')(wQ_A + r_AQ_A) = (s + r')[(1 + b)Q_A + r_AQ_A]$, and according to definition of $Q_A$, $sQ_A = s(vP) = s(1+j)P = (1+j)(sP)$, and then outputs $s(r_AQ_A)$ as $s(r_AQ_A) = \sigma_A - r'(r_AQ_A) - (1 + b)r'Q_A - (1 + b)(1 + j)(sP)$.

This completes the description of algorithm $\mathcal{B}$.

To determine the probability $\varepsilon'$ that algorithm $\mathcal{B}$ solves the SDH problem, we analyze the three events for algorithm $\mathcal{B}$ to succeed:

1. $\mathcal{E}_1$: $\mathcal{B}$ does not abort as a result of any of $\mathcal{A}$'s signature queries.
2. $\mathcal{E}_2$: $\mathcal{A}$ generates a valid message-signature forgery $\langle M_A, \sigma_A, r_AQ_A\rangle$.
3. $\mathcal{E}_3$: Event $\mathcal{E}_2$ occurs and $c = 0$ for the tuple containing $M_A$ on the $H$-list.

$\mathcal{B}$ succeeds if all of these events happen. The probability $Pr[\mathcal{E}_1 \wedge \mathcal{E}_3]$ is

$$Pr[\mathcal{E}_1 \wedge \mathcal{E}_3] = Pr[\mathcal{E}_1] \cdot Pr[\mathcal{E}_2|\mathcal{E}_1] \cdot Pr[\mathcal{E}_3|\mathcal{E}_2 \wedge \mathcal{E}_1] \tag{5.1}$$

The following claims give a lower bound for each of these terms.

**Claim 1.** *The probability that algorithm $\mathcal{B}$ does not abort as a result of algorithm $\mathcal{A}$'s signature queries is at least $1/e^2$. Hence $Pr[\mathcal{E}_1] \geq 1/e^2$*

***Proof***. According to the definition of signature-query above, for a signature query to fail, the possibility is that it fails in a $H$-query to obtain a $w_i \in \mathbb{G}_1$ such that $H(M_i) = w_i$, or it fails in a $H$-query to obtain a $v_i \in \mathbb{G}_1$ such that $H(ID_i) = v_i$. In either failure, the probability is $Pr[c_i = 0] = 1/(q_S + 1)$ or $Pr[k_i = 0] = 1/(q_S + 1)$. The probability of no failure is thus $Pr[c_i = 1] = 1 - 1/(q_S + 1)$ or $Pr[k_i = 1] = 1 - 1/(q_S + 1)$.

Since $\mathcal{A}$ makes at most $q_S$ signature queries, the probability that $\mathcal{B}$ does not abort as a result of $H(M_i) = w_i$ is at least $Pr[H(M_i) = w_i] = Pr[c_i = 1] = (1 - 1/(q_S + 1))^{q_S} \geq$

$1/e$, and the probability that $\mathcal{B}$ does not abort as a result of $H(ID_i) = v_i$ is at least $Pr[H(ID_i) = v_i] = Pr[k_i = 1] = (1 - 1/(q_S + 1))^{q_S} \geq 1/e$.

Hence, the probability that $\mathcal{B}$ does not abort in a $H$-query is at least $Pr[\mathcal{E}_1] = Pr[H(M_i) = w_i] \cdot Pr[H(ID_i) = v_i] \geq 1/e \cdot 1/e = 1/e^2$. □

**Claim 2.** *If algorithm $\mathcal{B}$ does not abort as a result of algorithm $\mathcal{A}$'s signature queries then algorithm $\mathcal{A}$'s view is identical to its view in the real attack. Hence, $Pr[\mathcal{E}_2|\mathcal{E}_1] \geq \varepsilon$.*

***Proof***. The public key given to $\mathcal{A}$ is from the same distribution as a public key produced in Setup step. Responses to $H$-queries are as in the real attack since each response is uniformly and independently distributed in $\mathbb{G}_1$. All responses to signature queries are valid. Therefore, $\mathcal{A}$ will produce a valid message-signature tuple with probability at least $\varepsilon$. Hence, $Pr[\mathcal{E}_2|\mathcal{E}_1] \geq \varepsilon$. □

**Claim 3.** *The probability that algorithm $\mathcal{B}$ does not abort after algorithm $\mathcal{B}$ outputs a valid forgery is at least $1/(q_S + 1)^2$. Hence, $Pr[\mathcal{E}_3|\mathcal{E}_2 \wedge \mathcal{E}_1] = 1/(q_S + 1)^2$.*

***Proof***. Given that events $\mathcal{E}_1$ and $\mathcal{E}_2$ happened, algorithm $\mathcal{B}$ will abort only if $\mathcal{A}$ generates a forgery signature tuple $\langle M_A, \sigma_A, r_A Q_A \rangle$ for which the tuple $\langle M_A, w, b, c \rangle$ on the $H$-list has $c = 1$ or the tuple $\langle ID_A, v, j, k \rangle$ on the $H$-list has $k = 1$. $c$ and $k$ are independent of $\mathcal{A}$'s current view. Therefore, according to the generation function of $c_i$ and $k_i$, $Pr\,[c = 0|\mathcal{E}_1 \wedge \mathcal{E}_2] = 1/(q_S + 1)$, and $Pr\,[k = 0|\mathcal{E}_1 \wedge \mathcal{E}_2] = 1/(q_S + 1)$. Hence, $Pr\,[\mathcal{E}_3|\mathcal{E}_1 \wedge \mathcal{E}_2] = 1/(q_S + 1)^2$ as required. □

Combining the bounds from the claims above in Equation 5.1 shows that algorithm $\mathcal{B}$ produces the correct answer with probability $AdvSig_{\mathcal{B}} = \varepsilon' \geq \frac{\varepsilon}{e^2(q_s+1)^2}$ as required. $\mathcal{B}$'s running time is the same as $\mathcal{A}$'s running time plus the time it takes to respond to $q_S$ signature queries, and $2q_S + q_H$ $H$-queries (each signature query calls $H$-query twice). Each query requires a constant time $c_1$. Hence, the total running time is at most $\tau' \leq \tau + c_1(q_H + 3q_S)$ as required. This completes the proof of Theorem 1. □

The proof of Theorem 1 resembles the method and procedure used in [13].

**Theorem 2** *Suppose Assumption 1 is true. The 1-to-$m$ broadcast key $k$ in Section 4.2 is only known to authentic nodes.*

***Proof***. The 1-to-$m$ broadcast key is $k = H_0(g^r)$. To get $k$, one needs to know $g^r$. Since

$$\begin{aligned} g^r &= \hat{e}(d_A, P) \\ &= \hat{e}(Q_A, P_{pub})^r \\ &= \hat{e}(rQ_A, P_{pub}) \end{aligned}$$

There are three ways to calculate $g^r$:

- Know $d_A$ and $P$.
- Know $Q_A$, $P_{pub}$ and $r$.
- Know $rQ_A$ and $P_{pub}$.

For an adversary, $Q_A$ and $rQ_A$ are publicly known from a message or packet; $d_A$, $P$ and $P_{pub}$ are unknown according to Assumption 1; $r$ cannot be calculated from $rQ_A$ according to discrete logarithm problem on elliptic curves. Only from $Q_A$ or $rQ_A$, an adversary cannot deduce $k$. □

Based on the above assumptions and theorem, it is clear that the proposed scheme provides confidentiality, integrity, authentication, freshness and non-repudiation.

- *Confidentiality:* All routing messages are encrypted using the 1-to-$m$ broadcast key. To decrypt the message, an entity needs to know decryption key $k$. According to Theorem 2 and Assumption 3.2, an adversary cannot calculate the key and cannot decrypt the message without a correct key.
- *Integrity:* All routing messages and packets are signed by message originators and packet senders. The recipients verify the signature of routing packets and messages before accepting them. Since the hash function is collision resistant, an altered image cannot go through with an unchanged signature. Thus the integrity is verified.
- *Authentication:* All routing messages are signed by the originators. According to Theorem 1, an adversary not knowing the private key of a node cannot generate a valid signature; thus a message with a valid signature can authenticate the identity of the originator of the message. For the same reason, an adversary cannot forge a message with a valid signature, and authenticity of messages is ensured.
- *Freshness:* Since all routing messages are signed by the originators, according to Theorem 1, an adversary cannot forge a message or modify any part of a message.

With a timestamp inside a message and system-wide synchronization, a replayed message or an out-of-order message can be easily differentiated from original one. The freshness of routing messages can be ensured.

- *Non-repudiation:* A routing message is signed by the originator using its private key, and a routing packet is signed by a packet sender using its private key. The private key is only known by exactly one node. According to Theorem 1, any other node or any adversary cannot forge a signature on a message and identity; thus the node that has generated or forwarded and signed the message cannot deny signing the message.

## 5.2 Supporting Threshold Cryptography without Mobile Attacks

Threshold cryptography is of great importance to IBC: Many IBC schemes use it for key generation and update/refresh, for addressing identity disclosure and key escrow problems, or for dynamic membership management. We need to support threshold cryptography because we might need some of these features in some circumstances. However, as we descripted in Chapter 3, threshold cryptography has three issues when used in IBC schemes.

- Interdependency cycle between secure routing and key management: The KM-SR integrated framework has addressed this issue.
- Proximity-caused insecurity: Since there is no KM-SR interdependency cycle, there is no need to bring a threshold number of PKG nodes in a proximity, and there is no proximity-caused insecurity.
- Mobile attacks: This is the only remaining issue we need to tackle in this section.

In mobile attacks, a mobile adversary could move to compromise multiple nodes and reveal the secret shares of them in order to recover the secret. To counter mobile attacks, many proposals use secret refreshing mechanism in which secret shares are updated in intervals and new shares cannot be combined with old ones to recover the secret, e.g. [99]. They assume a mobile adversary cannot compromise enough authentic nodes within the share refreshing period. We do not think this assumption is practical, as suggested by Merwe et al in [65]. We also see some loopholes in this solution against mobile attacks:

1. If there are more than one mobile adversaries in a $(n, t+1)$ threshold cryptosystem, they can compromise $t+1$ nodes and reveal $t+1$ shares within one refreshing interval, and collude to recover the secret.

2. A single adversary can compromise a node at one time, leave a backdoor, and read new shares later, if the compromised node has not been detected and revoked in time. Then in a refreshing interval, the adversary does not need to compromise $t+1$ nodes, but only need to collect $t+1$ shares. This saves a great magnitude of time and makes it possible for the adversary to win the game.

Once the system master key is recovered, the adversary can induct new adversary nodes by generating legal private keys for them, masquerade authentic nodes, and launch Sybil attacks. They can even take over the entire network.

### 5.2.1 Enhancement against Mobile Attack

Based on the KM-SR integrated framework, we propose an enhancement that refreshes or updates master key and private keys so that the mobile attackers cannot compromise these keys. The assumption of the scheme is a cooperative ad-hoc network in which identities of nodes are authenticated by the TA and system parameters are distributed by the PKG before dispatching, so that the initial status of the network is secure.

Previous solutions suggest dividing system master key among all online nodes or a small group of them. The mobile attackers can recover the master key by compromising a number of nodes holding the secret shares above the threshold. To improve security, the basic idea of our scheme is that the secret is divided into two parts: static part and dynamic part. At any time after the dynamic part has been generated, the working system master key is the combination of static part and dynamic part. The static part is kept offline, so that adversaries cannot locate and compromise that part. The dynamic part is generated and refreshed online as previous solutions do.The online nodes do not know the static part, so the mobile adversaries cannot get that part even if they compromise enough number of the online nodes.

With the KM-SR integrated framework proposed in previous chapter, when the network starts up, nodes start exchanging routing messages. Eventually, a secure routing is set up using routing messages protected by the initial secret. To avoid key compromise by mobile attackers, we update the master key $s$, system public key $P_{pub}$, and private key of each node corresponding to $s$. The new master key is updated with two parts: static

part—$s_{sta}$ that is always equal to initial master key $s$ generated by the offline administrator, and dynamic part—$s_{dyn}$ generated by all nodes contributively. New system public key and private keys are determined by the new master key. We use $(n, t+1)$ threshold cryptography to generate $s_{dyn}$ so that $s_{dyn}$ can then be recovered by $t+1$ nodes out of $n$ nodes who participated in the generation. This increases availability of key generation or update in a dynamically changing network environment.

After the initial secure routing is set up, nodes can communicate with each other using pairwise communication. Since there is no more KM-SM interdependency cycle, the previous key management schemes (such as those in [99] and [28]) are applicable. Nodes can now update their private keys on-line, to fight against crypto-analysis against initial secrets, or to fight against mobile attacks. To this end, we suggest dividing the master key, system public key, or a private key into a static part and a dynamic part, and updating the dynamic part only. The static part is the initial master key, system public key, or the initial private key.

Assume there are $n$ nodes in the initial network. Each node $C_i$ randomly chooses a secret $k_i$ and a polynomial $f_i(z)$ over $\mathbb{Z}_q$ of degree $t$, such that $f_i(0) = k_i$. Node $C_i$ computes his sub-share for node $C_j$ as $ss_{ij} = f_i(j)$ for $j = 1, 2...n$ and sends $ss_{ij}$ securely to $C_j$ using pairwise secret key ($K_{AB} = \hat{e}(d_A, Q_B) = \hat{e}(d_B, Q_A) = K_{BA}$) or secret session key (for example, the session key generation in [45]). After receiving $n-1$ sub-shares, node $C_j$ can compute its share of dynamic part of master key $s_{dyn}$ as $S_j = \sum_{i=1}^{n} ss_{ij} = \sum_{i=1}^{n} f_i(j)$. Any coalition of $t+1$ shareholders (assume the $t+1$ nodes form a set $\mathbb{T}$) can jointly generate a new secret key $s_{dyn}$ as in basic secret sharing [99]:

$$s_{dyn} = \sum_{i=1}^{t+1} S_i l_i(x)|_{x=0} \pmod{q}, \tag{5.2}$$

where $l_i(x) = \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)}, (i, j \in \mathbb{T})$ is the Lagrange coefficient, $x_i$ and $x_j$ are node indexes derived from node identities. Due to the homomorphic property of share refreshing, the jointly generated dynamic part of master key is $s_{dyn} = \sum_{i=1}^{n} k_i = \sum_{i=1}^{n} f_i(0)$.

For each initial node with identity $ID_i$ in the network to get the new system public key $P'_{pub}$ and refresh its private key $d'_i$, it contacts $t$ nodes using pairwise secret keys. Each of the $t$ nodes generates for the requesting node a secret share of the dynamic part of system public key $S_i P$, and new dynamic part of private key $S_i Q_i$, using its own share of dynamic part of master key—$S_i$, and sends them to the requesting node. After collecting $t$ shared secrets from other nodes separately and generating its own share, the $(t+1)$-th

share, the requesting node can calculate the new dynamic part of system public key

$$\begin{aligned} P_{pub_{dyn}} &= \sum_{i=1}^{t+1} [(S_i \cdot P) \cdot l_i(x)|_{x=0}] \\ &= \sum_{i=1}^{t+1} [(S_i \cdot P) \frac{\prod_{j \neq i}(-x_j)}{\prod_{j \neq i}(x_i - x_j)}] \end{aligned} \tag{5.3}$$

and its new dynamic part of private key

$$\begin{aligned} d_{i_{dyn}} &= \sum_{i=1}^{t+1} [(S_i \cdot Q_i) \cdot l_i(x)|_{x=0}] \\ &= \sum_{i=1}^{t+1} [(S_i \cdot Q_i) \frac{\prod_{j \neq i}(-x_j)}{\prod_{j \neq i}(x_i - x_j)}]. \end{aligned} \tag{5.4}$$

Note that $P_{pub_{dyn}} = \sum_{i=1}^{n} f_i(0) \cdot P = s_{dyn} \cdot P$ and $d_{i_{dyn}} = \sum_{i=1}^{n} f_i(0) \cdot Q_i = s_{dyn} \cdot Q_i$. It then combines the parts from $t$ nodes and its initial system parameters to get the system public key and its private key. As a result, the new master key is

$$s' = s + s_{dyn}; \tag{5.5}$$

the new system public key is

$$P'_{pub} = P_{pub} + P_{pub_{dyn}}; \tag{5.6}$$

and the new private key of node $i$ is

$$d'_i = d_i + d_{i_{dyn}}. \tag{5.7}$$

Note that the new master key $s'$ actually does not exist on any node. We show it just for explaining the new public key and new private key. The initial public key $P_{pub}$ (now the static part of new public key) is kept by all nodes so that the offline administrator can sign a message using its initial private key and other nodes can verify it using the initial public key. This is useful when the off-line administrator admits a new node to join the network.

### 5.2.2 Support for Dynamic Membership

When a new node with ID $ID_p$ needs to join, its public key is explicit as $Q_p = H_1(ID_p)$. What it needs to get is system public key and its private key. The question here is that the new node knows nothing about the latest secret among authentic nodes then and cannot join their communication, and the authentic nodes cannot judge if the new node is an adversary or not. To tackle this question, the offline administrator is essential. The new node first contacts the offline administrator and gets the initial system public key $P_{pub}$ (now the static part of new public key), its initial private key $d_p$ (now the static part of its private key), and an "entrance ticket" signed by the administrator using its private key. The new node then contacts $t+1$ nodes, presents the "entrance ticket" that can be verified using the initial system public key (now the static part of new system public key), and gets $t+1$ shares of the dynamic part of system public key $P_{pub_{dyn}}$, and dynamic part of its private key $d_{p_{dyn}}$. Since the secure routing is already established, the new node can communicate with remote nodes through its neighbors. It then combines the dynamic part of system public key and dynamic part of private key with initial ones (static parts), as explained above: $P'_{pub} = P_{pub} + P_{pub_{dyn}}$ and $d'_p = d_p + d_{p_{dyn}}$.

One thing needing to note is that the new node does not have a share of dynamic part of master key $s_{dyn}$, $S_p$, as original nodes. This may deteriorate the usability or performance of threshold cryptography in the long run, since if $n - t$ original nodes have left the network or died at some time, there are not enough nodes to accept new nodes, and the network can no longer be autonomous or self-organizing. To generate new shares of dynamic part of master key $s_{dyn}$, each of the $t+1$ nodes, holding secret share $S_i$, generates for new node $p$ a new share from its own share using Lagrange polynomial: $S_i \cdot l_i(p) = S_i \cdot \frac{\prod_{j \neq i}(x_p - x_j)}{\prod_{j \neq i}(x_i - x_j)}$. After receiving $t+1$ shares from these nodes, the new node can calculate its own share simply by summing up these shares: $S_p = \sum_{i=1}^{t+1} S_i l_i(p)$. At this point, the new node has no difference than the initial nodes, and can join the network communication and secret updates.

When a node leaves the network, nothing needs to be done unless the number of remaining nodes approaches $t+1$ in which situation a new lower $t+1$ is determined at each node by calculating on the number of nodes in the routing table. For a returning node that leaves the network temporarily, it needs to check the version of keys when returning. If its version is not the latest, it needs to go through the procedure for a new node; otherwise it can return and join directly.

To ensure the consistency of the system public key and private keys, we suggest each

secret share have an attached sequential number count for each secret update. Only shares with the same sequential numbers can generate new keys collaboratively. In this scheme, each node can get the system public key and its new private key, without master key being revealed to any node. The adversaries cannot get the master key unless they compromise both the offline administrator and at least $t + 1$ online nodes.

### 5.2.3 A Working Example of the Scheme

We design an example to illustrate the scheme. Assume there are 5 initial nodes, and the initial master key generated by offline administrator (i.e. the static part of master key) is 6. The contributions of a new dynamic part of master key by initial nodes are shown in Table 5.1. The procedure of generating shares of dynamic part of master private key is shown in Table 5.2.

| **Node** | **Secret and Polynomial** |
|---|---|
| 1 | $5 + 10x + x^2$ |
| 2 | $12 + 20x + 2x^2$ |
| 3 | $15 + 30x + 3x^2$ |
| 4 | $18 + 40x + 4x^2$ |
| 5 | $22 + 50x + 5x^2$ |
| **Dynamic Part of Master Key** | $5 + 12 + 15 + 18 + 22 = 72$ |

Table 5.1: Contributions of Dynamic Part of a New Master Key

| **Node** | **Sub-shares from Node ($ss_{ij}$)** | | | | | **Share of Master Key** |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | $(\sum_{i=1}^{5} ss_{ij})$ |
| 1 | 16 | 34 | 48 | 62 | 77 | 237 |
| 2 | 29 | 60 | 87 | 114 | 142 | 432 |
| 3 | 44 | 90 | 132 | 174 | 217 | 657 |
| 4 | 61 | 124 | 183 | 242 | 302 | 912 |
| 5 | 80 | 162 | 240 | 318 | 397 | 1197 |

Table 5.2: Sub-shares and Shares of Dynamic Part of a New Master Key

| New Node | Share of Dynamic Part of Master Key | | | | | New Share |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | |
| 1-5 | 237 | 432 | 657 | 912 | 1197 | - |
| 6 | | 144 | | -1824 | 3192 | 1512 |
| ... | ... | ... | ... | ... | ... | ... |

Table 5.3: New Node (6) Gets Its Share of Dynamic Part of Master Key

To recover the dynamic part of the master key from 3 nodes, say 2,4,5, using $(3, 5)$ threshold cryptosystem, these nodes first calculate Lagrange polynomials then the dynamic part of new master key $s_{dyn} = \sum_{i=1}^{3} S_i l_i(0) = 432 \cdot l_1(0) + 912 \cdot l_2(0) + 1197 \cdot l_3(0) = 72$. And the new master key is $s' = 72 + 6 = 78$.

When a new node Node-$6$ joins the network, it gets its static part from offline administrator, and share of dynamic part of new master key calculated from shares of node $2$, $4$, $5$, as is shown in Table 5.3.

## 5.3 Enhancement against Blackhole Attacks

In blackhole attacks, an adversary node advertises itself as having the shortest path to some other nodes, and then receives the traffic to these nodes. The adversary then can choose to drop the traffic or redirect it to nodes pretending to be the destination. There are two cases in this type of attacks:

### 5.3.1 Without Compromised Nodes

This type of attacks can be prevented by the KM-SR integrated framework already: an adversary node cannot advertise forged routing messages to other nodes, because it cannot forge the required signature.

### 5.3.2 With Compromised Nodes

If there are some nodes compromised, the adversary can advertise forged routing messages using the authentic identity and signature of a compromised node. To fight against this attack, we can verify a routing information with its neighbor's attestation or certificate assuming its neighbor is not compromised. We need some extension in routing messages:

At beginning of routing setup, each empty *HELLO* message (the initial *HELLO* message before neighbor recognized) has its key information (such as timestamp) and originator address signed by the originator, and attach the key information and signature in the message (originator address is implicit in the message). Later, each *HELLO* message carries updated key information and signature, and each neighbor advertised in a *HELLO* or *TC* message is accompanied with the key information and signature from the original *HELLO* message from that neighbor (the advertised neighbor is the originator when verifying the signature).

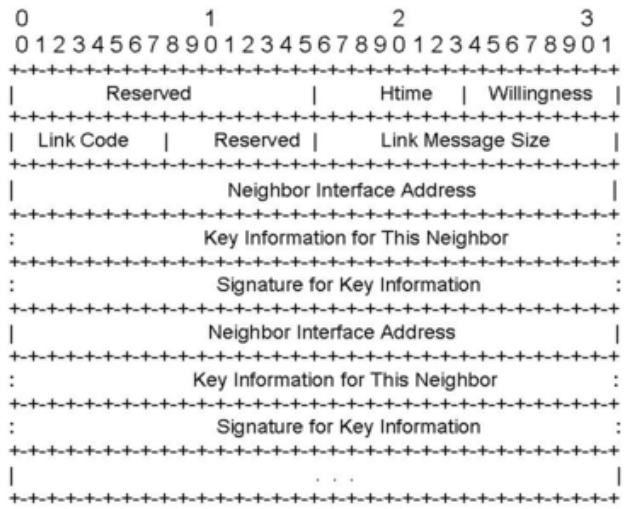
This scheme is a simplified version of Raffo's [74]. In his proposal, neighbor's attestation is updated for each update of link status, so that attestation is nested and can be very large. For example, a neighbor information in a *HELLO* message with *SYM_NEIGHBOR* link status needs to include attestation for a *HELLO* message with *SYM_LINK* link status which needs again to include attestation for a *HELLO* message with *ASYM_LINK* link status. Finally, a neighbor information in a *HELLO* message with *SYM_NEIGHBOR* link status needs 3 key information fields and 3 attestations; otherwise, blackhole attacks could be successfully launched. In our scheme, we only include one attestation—the original one from that neighbor, and require that a symmetric link can be set up only when both ends of the link provide valid attestations from each other in their *TC* messages; in this way, a false *HELLO* message does not lead to a false route.

Figure 5.1(a) and Figure 5.1(b) show the structure of a *HELLO* message and a *TC* message with neighbor's attestations.

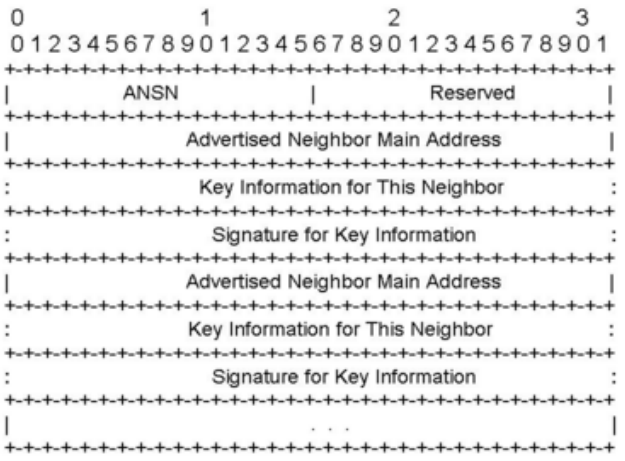
If there are enough nodes compromised in the network so that they can collude to forge false routing advertisement, per-advertisement signature is insufficient to detect the attack. We will need measures against denial-of-service attacks to be discussed shortly.

## 5.4 Against Wormhole Attacks

In wormhole attacks, adversaries can collude to transport routing packets out of band. In a routing packet, there are two types of routing messages: the local messages and the global messages. The global messages are meant to be propagandized to all nodes; so wormhole attacks to these messages are not harmful, but actually favorable. Only messages that are meant to be exchanged locally, for example, neighbor advertisement, should not be distributed out of neighborhood. To detect the local routing messages distributed out-of-neighborhood, we can use time-based method and location-based method.




(a) OLSR *HELLO* message with neighbor's attestations




(b) OLSR *TC* message with neighbor's attestations

Figure 5.1: Structure of OLSR *HELLO* and *TC* Message with Neigbor's Attestations

- **Time-based method** If precise timestamps is available in routing messages, our KM-SR framework has them signed in the message by the message originators. A packet from a previous time or a different place which travels through more hops bears a previous timestamp, and can be easily distinguished from correct ones.

- **Location-based method** In MANETs where GPS or other locating devices are equipped on nodes, location-based method can be used to fight against wormhole attacks. In one of our previous work, we designed an OLSR plug-in Deployment Information (DI) message to exchange deployment information. The specification of the DI message can be found in our previous paper [98], but we present a short summary here. Figure 5.2(a) shows the structure of the DI message which includes:

  – Node Location & Velocity

  – Node Deployment Profile: The deployment profile serial number is used to indicate the current deployment profile that is configured in the node.

  – Node Status Information: Node Status Information field is used to carry alarms to the Deployment Tool. Such alarms include equipment malfunctions and configurable alarms specified in the node deployment profile.

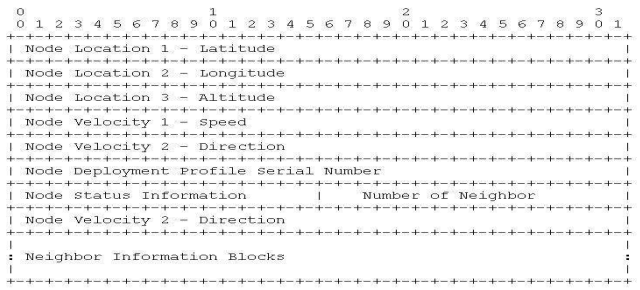
  – Number of Neighbours: The Number Of Neighbors field is the number of one-hop neighbours for which there is a link quality measurement result available.

  – Neighbour Information Block: A Neighbour Information Block (see Figure 5.2(b)) contains a one-hop neighbour description. There is a neighbourhood information block for each one hop neighbour, and the neighbourhood information block has two fields:

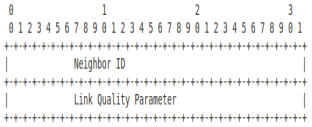
    * The *Neighbour ID* identifies the one-hop neighbour. In our implementation, we used IPv4 address as an identifier. We assume that the addresses are allocated and configured in advance.

    * The *Link Quality Parameter* describes the level of connection to the neighbour based on link measurement.

Although this DI message plug-in was not originally designed for security purpose, it provides enough information to fight against wormhole attacks if put in use in our proposed KM-SR integrated framework.




(a) The Structure of the DI Message




(b) Neighbour Information Block

Figure 5.2: OLSR Deployment Information Message

## 5.5 Against Other Routing Attacks

With or without modification or addition, the scheme is also immune to the following attacks:

- **Spoofing and Sybil Attacks:** With this type of attacks, an adversary node attempts to take over the identity of another node. These attacks can be prevented by authentication feature of the cryptographic scheme.

- **Eavesdropping and Traffic Analysis:** Due to wireless communication features of MANETs, an adversary may eavesdrop and analyze traffic in the air. This type of attacks are prevented by confidentiality feature of the scheme.

- **Modifying a routing message or packet:** An adversary may try to modify a routing message or packet content to propagandize false routing information. This is prevented by integrity feature of the scheme.

- **Denial-of-service Attacks:** Denial-of-service (DoS) attacks can be applied by adversaries on various layers of an MANET system to block proper transmission of legitimate traffic. In general MANETs, one type of DoS attacks is adversary nodes advertising false or true routing messages and participating in routing setup but actually not in traffic forwarding, thus leads to network malfunction. This type of attacks are not possible in our KM-SR integrated framework, because authentication of nodes and messages prevents this behavior by adversaries. Nevertheless, some behaviors of authentic nodes may have also this effect (e.g. **Selfishness** described below). Another type of DoS attacks is to propagandize false routing messages that are meant to consume limited communication and computational resources of authentic nodes.

On network and application layers, monitoring and accusation systems are effective countermeasures against DoS attacks. The proposed framework does not provide a monitoring and accusation system by itself, but provides a solid bedrock for such a system.

- **Selfishness:** This is actually not an attack, but rather a malfunction of some nodes of the system. An authentic node may have some reasons not to forward traffic that it should forward, for example, to save its energy, or it is just blocked in some specific terrain. The effect of selfishness is similar to denial-of-service attacks. This type of attacks can be detected and prevented with the measures used against denial-of-service attacks in which selfish nodes are treated equally as adversary nodes.

- **Rushing Attacks**: In reactive routing protocols, to limit the overhead of flooding, each node typically forwards only one ROUTE REQUEST originating from any Route Discovery. If the ROUTE REQUESTs for this Discovery forwarded by the attacker are the first to reach each neighbor of the target, then any route discovered by this Route Discovery will include a hop through the attacker ( this is because when a neighbor of the target receives the rushed REQUEST from the attacker, it forwards that REQUEST, and will not forward any further REQUESTs from this Route Discovery.) Thus if the adversary forwards route request faster than valid ones, then the discovered route would include the adversary. This only applies to reactive routing protocols. In our scheme, we choose a proactive routing protocl and do not have to worry about this type of attacks.

- **Record-and-replay Attacks**: A node can record a message or a packet from some place and replay it somewhere else, or record a packet at some time and replay it some time later.

  Record-and-replay attacks on message level are not available in the proposed KM-SR integrated framework, since every packet is signed and verified and there is no possibility to replace a message in a packet. If we only signed the messages but not the packets, the adversary would be able to apply record-and-replay attacks by replacing a message with a recorded valid message.

  On packet level, the adversary may record and replay an entire packet which contains inappropriate routing messages. To detect an out-of-order message, the message sequence number in a message is meant for this. To detect an out-of-date

routing message, we can include in the signed routing messages time-based information. To detect a local routing message distributed out-of-neighborhood is similar to detecting wormhole attacks, we can use location-based information as was described in Section 5.4.

## 5.6 Summary of the Chapter

This chapter presented proof of security of the cryptographic scheme of the framework, analysis of the security features of the framework, and immunity and enhancements against various attacks. Performance-related issues will be discussed in the next chapter.

# Chapter 6

# Simulation Results and Performance Analysis of the Framework

In this chapter, we analyze computational complexity of the cryptographic scheme and efficiency of the KM-SR framework from perspectives of transmission overhead and end-to-end delay. We also demonstrate performance of the framework with practical simulation. For scalability, we give an evaluation model according to communication and computational overhead per node.

## 6.1 Computational Complexity and Efficiency Analysis

The encryption/decryption and signature/verification schemes used in our KM-SR framework are of high efficiency and low complexity, compared to other existing cryptographic schemes.

- Encryption: The 1-to-$m$ broadcast encryption key generation takes 1 pairing computation, 1 exponentiation computation, and 1 hash computation. Note that the most time-consuming pairing computation $\hat{e}(d_A, P)$ can be precomputed once and for all so that encryption requires no pairing computation. Encryption is fast due to use of symmetric key cryptography.

- Decryption: The 1-to-$m$ broadcast decryption key generation takes 1 pairing computation and 1 hash computation. Decryption is fast due to use of symmetric key cryptography.

- Signature: The signature operation takes 1 hash computation, 2 scalar multiplica-

tion computation. One optimization of the scheme is to use a fixed $r$ for each $Q_A$ and save $rQ_A$ for $Q_A$. This saves both communication and computational resource, with sacrifice of storage space.

- Verification: The verification operation takes 1 hash computation, 1 scalar multiplication computation, and 2 pairing computation.

Comparison to existing cryptographic schemes is summarized in Table 6.1 and Table 6.2

| | **Our Scheme** | **BF [10]** | **HIDE [39]** | **DHIDE [39]** | **IBEWOR [11]** |
|---|---|---|---|---|---|
| **Encryption** | 1E+1H+1X | 1E+2H+1P +1M+1X | 1E+2H+1M +1P+1E | 1E+2H+1M +1P+1X | 1E+5M |
| **Decryption** | 1H+1P+1X | 1H+1P+1X | 1H+1P+1X | 1H+1P+1X | 1A+1I+1M |
| Computations: A—addintion, H—hashing, I—inversion, E—exponentiation, M—scalar multiplication, P—pairing, X—XOR (To compare fairly on the same security level, we evaluate with XOR instead of *AES* in our scheme here.) | | | | | |

Table 6.1: Comparison of Our Encryption/Decryption Scheme with Others

| | **Our Scheme** | **BLS [14]** | **BSS [9]** | **MSS [9]** | **AGG [12]** | **ZSS [90]** |
|---|---|---|---|---|---|---|
| **Signature** | 1H+2M | 1H+1M | 1H+3M | 1H+1M | 1H+1M | 1H+1I +1M |
| **Verification** | 1A+1H+ 1M+2P | 1H+2P | 1H+2P | 2P | 1H+2P | 1A+1H +1M+2P |
| Computations: A—addintion, H—hashing, I—inversion, M—scalar multiplication, P—pairing | | | | | | |

Table 6.2: Comparison of Our Singature/Verification Scheme with Others

From these tables we can see that our encryption/decryption scheme is among the highest efficient ones, and our signature/verification scheme is just slightly worse than the highest efficient ones. As a side note, we have considered the highest efficient schemes and realized that they do not exactly match our key management scheme in MANET context.

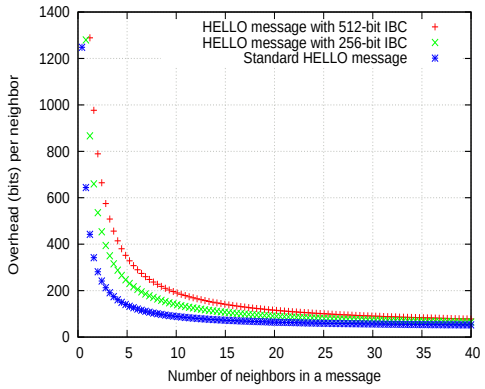
## 6.2 Transmission Overhead Analysis

In this section, we evaluate the transmission overhead of the secure routing protocol, compared to standard OLSR.

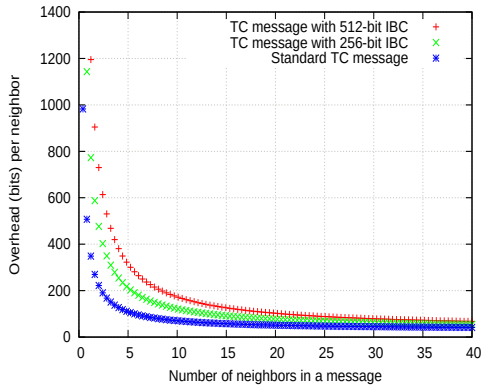
### 6.2.1 IBC Encryption/Signature Overhead in OLSR Packets

The average size of a standard OLSR *HELLO* packet is $488 + 40n$ bits, and of a standard OLSR *TC* packet is $384 + 32n$ bits, where $n$ is the number of advertised neighbors of this node, considering the IPv4 header (160 bits), the UDP header (64 bits), and the OLSR packet header (32 bits + 96 bits per message) [23, 74]. We assume each OLSR packet contains only *HELLO* or *TC* messages. This is the worst case scenario, as including more control messages in a packet would reduce the overhead.

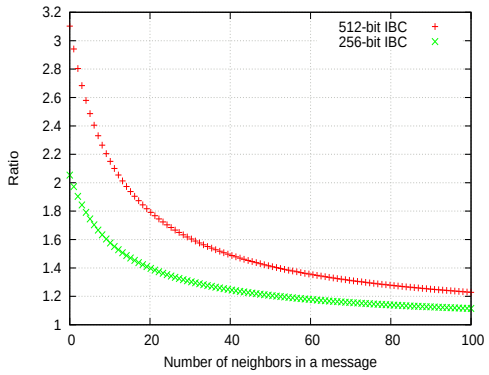
In the proposed scheme, a signature is a point on an elliptic curve. To save space, we can transmit only the x-coordinate and a sign bit. The overhead added by a packet signature is 512+1=513 bits with 512-bit IBC, and 256+1=257 bits with 256-bit IBC. The overhead added by a message signature and encryption is 512+1=513 bits with 512-bit IBC, and 256+1=257 bits with 256-bit IBC. Thus, the size of a packet with a *HELLO* message advertising $n$ neighbor nodes is: $513 + 513 + 488 + 40n = 1514 + 40n$ bits when using 512-bit IBC, and $257 + 257 + 488 + 40n = 1002 + 40n$ bits when using 256-bit IBC. The size of a packet with a *TC* message advertising $n$ neighbor nodes is: $513+513+384+32n = 1410+32n$ bits when using 512-bit IBC, and $257+257+384+32n = 898 + 32n$ bits when using 256-bit IBC. Figure 6.1(a) and Figure 6.1(b) show the overhead per neighbor in a standard *HELLO*/*TC* message and IBC encrypted and signed *HELLO*/*TC* message, in a network with 1 to 40 potential neighbors per node. The figures demonstrate that overhead for per neighbor advertised decreases dramatically in all scenarios and the difference between encrypted messages and non-encrypted messages quickly becomes ignorable when the network size and the number of potential neighbors per node increase. More intuitively, Figure 6.1(c) and Figure 6.1(d) show the ratio of IBC scheme overhead per neighbor to standard overhead which is approaching 1 when $n$ is getting larger. This is because when more neighbors are included in a message, the overhead of signature of packet and message is shared by more nodes, and per node overhead is decreased and approaching to standard level. This means that the cost per node is lower in a dense network (in which each node has more than 20 neighbors) than in a sparse network.



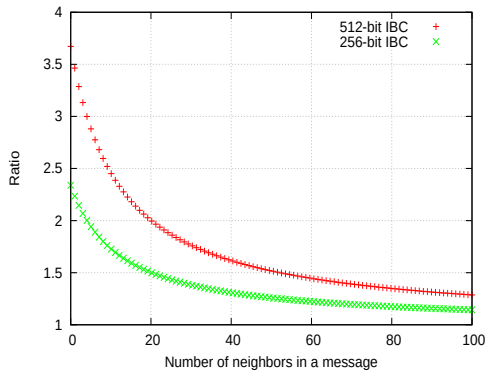

(a) OLSR *HELLO* message overhead per neighbor




(b) OLSR *TC* message overhead per neighbor




(c) Ratio of IBC *HELLO* message overhead to standard overhead



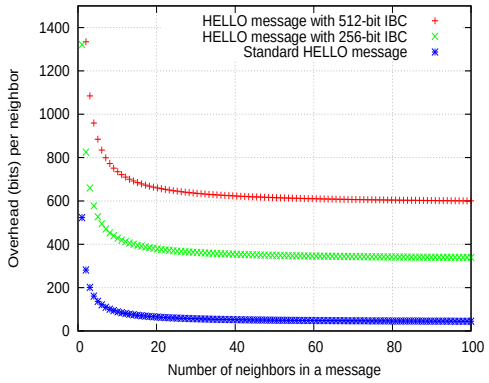

(d) Ratio of IBC *TC* message overhead to standard overhead

Figure 6.1: Transmission Overhead of a OLSR Packet with a *HELLO* or *TC* Message

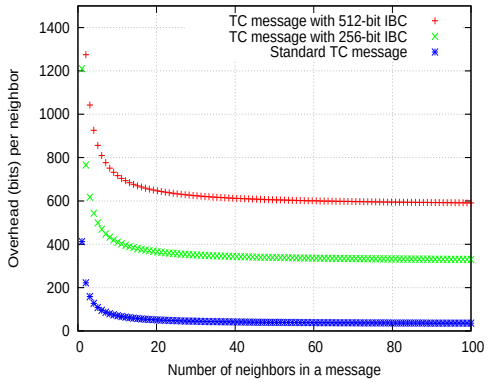
### 6.2.2 IBC Encryption/Signature and Neighbor's Attestation Overhead in OLSR Packets

The transmission overhead added by neighbor's attestation is large. For each neighbor advertised, a key information and signature are attached (originator address is implicit in the message). Assume we use a 32-bit timestamp as key information, and 513 or 257 bits for signature. The size of a packet with a *HELLO* message advertising $n$ neighbor nodes is: $1514 + (40 + 32 + 513)n = 1514 + 585n$ bits when using 512-bit IBC, and $1002 + (40 + 32 + 257)n = 1002 + 329n$ bits when using 256-bit IBC. The size of a packet with a *TC* message advertising $n$ neighbor nodes is: $1410 + (32 + 32 + 513)n = 1410 + 577n$ bits when using 512-bit IBC, and $898 + (32 + 32 + 257)n = 898 + 321n$ bits

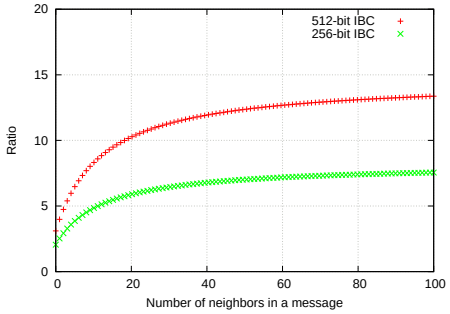
when using 256-bit IBC. Figure 6.2(a) and Figure 6.2(b) show the overhead per neighbor in a standard *HELLO*/*TC* message and IBC encrypted and signed *HELLO*/*TC* message with neighbor's attestation, in a network with 1 to 100 potential neighbors per node. Figure 6.2(c) and Figure 6.2(d) show the ratio of overhead of IBC encrypted and signed *HELLO*/*TC* messages with neighbor's attestation to standard overhead, in a network with 1 to 100 potential neighbors per node.



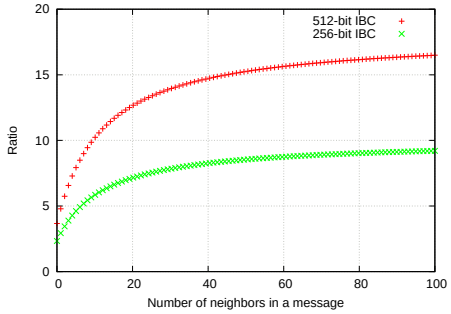

(a) OLSR *HELLO* message with neighbor's attestation overhead per neighbor




(b) OLSR *TC* message with neighbor's attestation overhead per neighbor




(c) Ratio of IBC *HELLO* message with neighbor's attestation overhead to standard overhead




(d) Ratio of IBC *TC* message with neighbor's attestation overhead to standard overhead

Figure 6.2: Transmission Overhead of a OLSR Packet with a *HELLO* or *TC* Message with Neigbor's Attestation

From these figures, we can see that per neighbor overhead remains almost constant when number of neighbors increases. This is because neighbor's attestation is per-neighbor information, when number of neighbors increases, attestation data increases proportionally.

### 6.2.3 An Optimization to Transmission Overhead

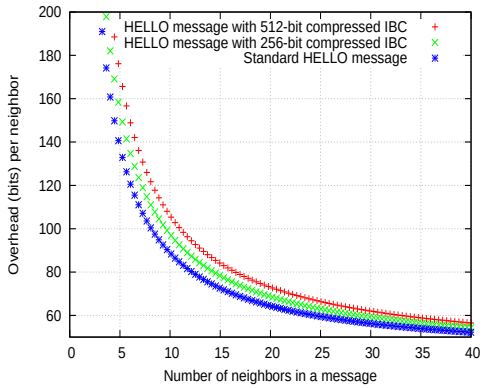
In situations where communication capacity is limited, one can choose to trade off between transmission overhead and computational overhead. As was noted in [13], some elliptic curves produce $n$-bit signatures and the discrete log problem on these curves is reducible to a discrete log problem in a finite field of size approximately $2^{6n}$. Using this type of curves, for $n$-bit security we get signatures of size $n/6$ bits. This can dramatically reduce size of signatures.

The size of an OLSR packet with a *HELLO* message advertising $n$ neighbor nodes is: $\lceil 513/6 \rceil \times 2 + 488 + 40n = 660 + 40n$ bits when using 512-bit IBC, and $\lceil 257/6 \rceil \times 2 + 488 + 40n = 574 + 40n$ bits when using 256-bit IBC. The size of a packet with a *TC* message advertising $n$ neighbor nodes is: $\lceil 513/6 \rceil \times 2 + 384 + 32n = 556 + 32n$ bits when using 512-bit IBC, and $\lceil 257/6 \rceil \times 2 + 384 + 32n = 470 + 32n$ bits when using 256-bit IBC. Figure 6.3(a) and Figure 6.3(b) show the overhead per neighbor in a standard *HELLO*/*TC* message and a *HELLO*/*TC* message with compressed IBC signatures, in a network with 1 to 40 potential neighbors per node. Figure 6.3(c) and Figure 6.3(d) show the ratio of overhead of a compressed IBC encrypted and signed *HELLO*/*TC* message to standard overhead, in a network with 1 to 100 potential neighbors per node.

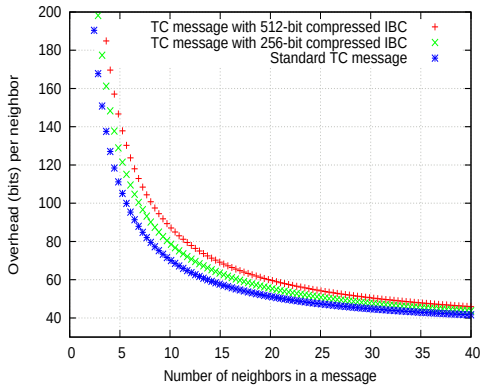
The size of a packet with a *HELLO* message advertising $n$ neighbor nodes with neighbor's attestation is: $\lceil 513/6 \rceil \times 2 + 488 + (40 + 32 + \lceil 513/6 \rceil)n = 660 + 158n$ bits when using 512-bit IBC, and $\lceil 257/6 \rceil \times 2 + 488 + (40 + 32 + \lceil 257/6 \rceil)n = 574 + 115n$ bits when using 256-bit IBC. The size of a packet with a *TC* message advertising $n$ neighbor nodes with neighbor's attestation is: $\lceil 513/6 \rceil \times 2 + 384 + (32 + 32 + \lceil 513/6 \rceil \times 2)n = 556 + 150n$ bits when using 512-bit IBC, and $\lceil 257/6 \rceil \times 2 + 384 + (32 + 32 + \lceil 257/6 \rceil)n = 427 + 107n$ bits when using 256-bit IBC.

Figure 6.4(a) and Figure 6.4(b) show the overhead per neighbor in a standard *HELLO*/*TC* message and *HELLO*/*TC* message with neighbor's attestation and compressed IBC signatures, in a network with 1 to 100 potential neighbors per node. Figure 6.4(c) and Figure 6.4(d) show the ratio of overhead of a compressed IBC encrypted and signed *HELLO*/*TC* message with neighbor's attestation to standard overhead, in a network with 1 to 100 potential neighbors per node.

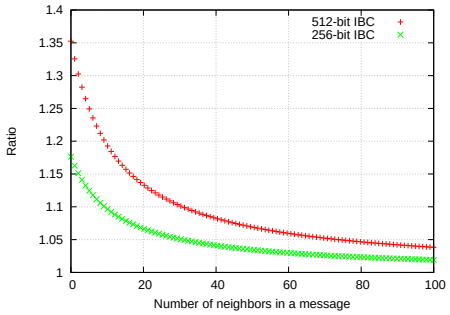
From these figures, we can see that the overhead is significantly reduced by compressed IBC signature compared to original one. We also see that neighbor's attestation brings too much overhead. If this feature is needed, we strongly suggest using compressed IBC signature.



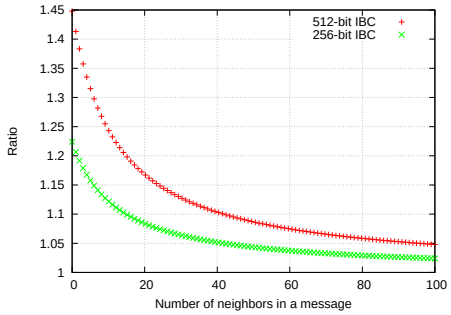

(a) OLSR *HELLO* message overhead per neighbor with compressed IBC




(b) OLSR *TC* message overhead per neighbor with compressed IBC overhead




(c) Ratio of *HELLO* message with compressed IBC overhead to standard overhead

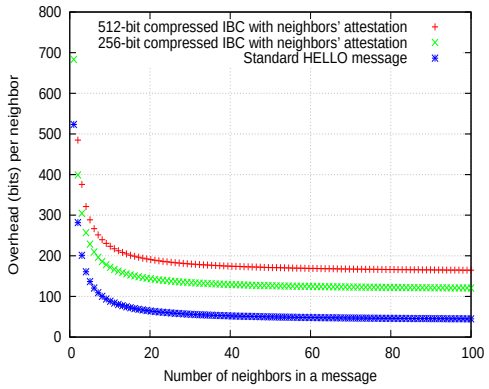



(d) Ratio of *TC* message with compressed IBC overhead to standard overhead

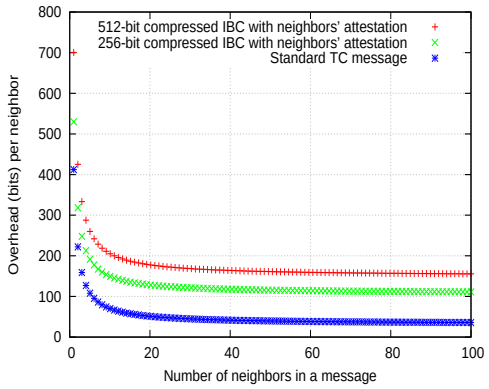
Figure 6.3: Transmission Overhead of a OLSR Packet with a *HELLO* or *TC* Message with Neigbor's Attestation and Compressed IBC Signatures

## 6.3 Simulation Setup

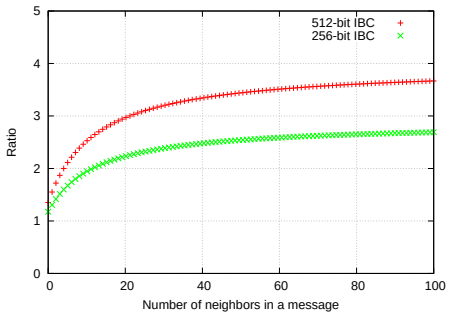
We have implemented the KM-SR framework in NS-2, a popular network simulator for MANETs, using cryptographic primitives from MIRACL library version 5.5. The bilinear map $\hat{e}$ we use is the Tate pairing. The elliptic curve $\mathbb{E}$ we use is Type A supersingular curve $y^2 = x^3 + x$ defined over the finite field $\mathbb{F}_p$ ($p$ is a prime and $p \equiv 3 \bmod 4$). We use a 160-bit Solinas prime $2^{159} + 2^{17} + 1$ as $q$, and use a 256-bit $p$ and a 512-bit $p$ to compare performance. We simulate an ad hoc network with 10 to 40 nodes uniformly deployed in a $700 \times 500\ m^2$ square field. The physical-layer path loss model is the two-ray model. The radio propagation range for each node is 250 meters and the channel capacity is 2 Mb/s. The base MAC protocol used is the DCF of IEEE 802.11. Node mobility uses the random



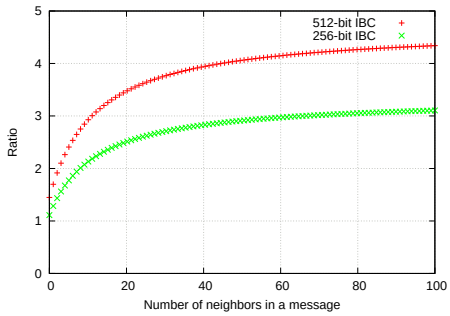

(a) OLSR *HELLO* message with neighbor's attestation overhead and compressed IBC per neighbor




(b) OLSR *TC* message with neighbor's attestation and compressed IBC overhead per neighbor




(c) Ratio of *HELLO* message with neighbor's attestation and compressed IBC overhead to standard overhead




(d) Ratio of *TC* message with neighbor's attestation and compressed IBC overhead to standard overhead

Figure 6.4: Transmission Overhead of a OLSR Packet with a *HELLO* or *TC* Message with Neigbor's Attestation and Compressed IBC Signatures

waypoint model with maximum pause time 10 seconds and maximum speed 1 m/s. CBR sessions are used to generate network data traffic at rate of 20 kb/s and packet size of 512 bytes. We execute the simulation on a computer with Intel Core-2 Duo 2.8GHz CPU and running RedHat Linux AS4.

## 6.4 Simulation Results and Analysis

We start the simulation with the environment and parameters mentioned above. We run simulations in 9 rounds which are the permutation of network size 10/20/40 and cryptographic settings no-cryptographic-operation/256-bit IBC/512-bit IBC. Each round is ex-

ecuted for 10 simulated minutes and each data point represents an average of ten runs with identical traffic models. After simulations are executed, we analyze results from simulation logs and draw the figures below.

Figure 6.5 shows end-to-end delays of routing messages, measured by the difference between sending time and receiving time, in network size 10/20/40 and cryptographic settings no-cryptographic-operation/256-bit IBC/512-bit IBC.

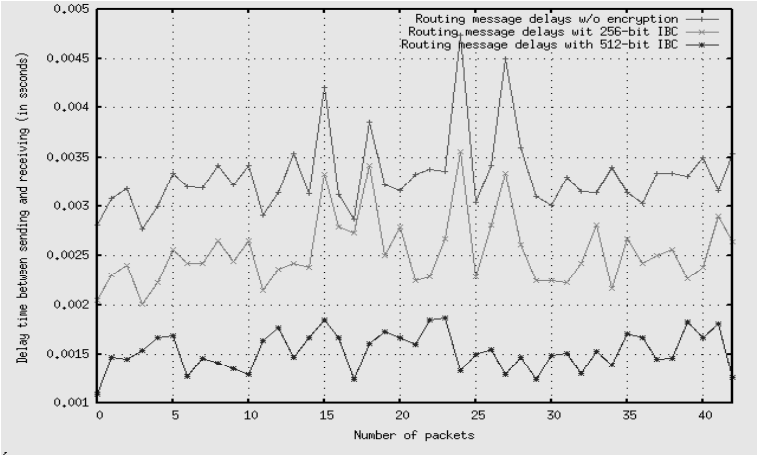
Figure 6.6 shows delays added by IBC cryptographic operations, i.e. delays with the 256-bit and 512-bit IBC minus delays with standard OLSR routing messages in same simulation setup. We compare the delays added by 256-bit IBC and 512-bit IBC in 10/20/40-node networks.

From these figures, we observe that:

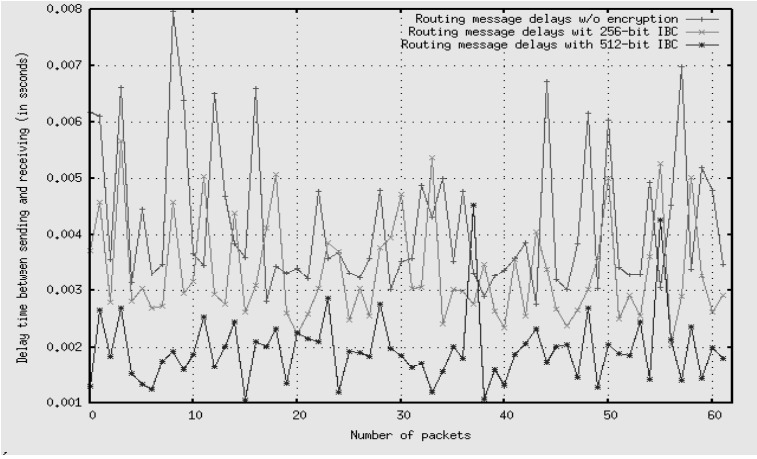
- There is a noticeable jitter in routing message delays, and this jitter increases when number of nodes increases. The reason of this is that when number of nodes increases, the number of routing messages a node receives also increases and the number of hops a routing message travels through may also increase, and the travelling time taken by a routing message from a 1-hop neighbor and the one by a routing message from a multi-hop node differ a lot. Because of the encryption/decryption and signature/verification operations on every hop, the secure routing scheme amplifies this difference.

- The proposed KM-SR framework does not cause any substantial degradation in the network performance. Communication and computational overhead is stable in all scenarios. For example, in the 40 node setting, the average delay of routing messages without cryptographic operation is about 0.002 second; the average delay of routing message with 256-bit IBC is about 0.004 second, and the average delay of routing messages with 512-bit IBC is about 0.006 second. Compared to the simulation results measured in many other schemes (mostly 0.01 second to 0.1 second) for example [94] and [28], the delay added by secure routing in our framework is acceptable.

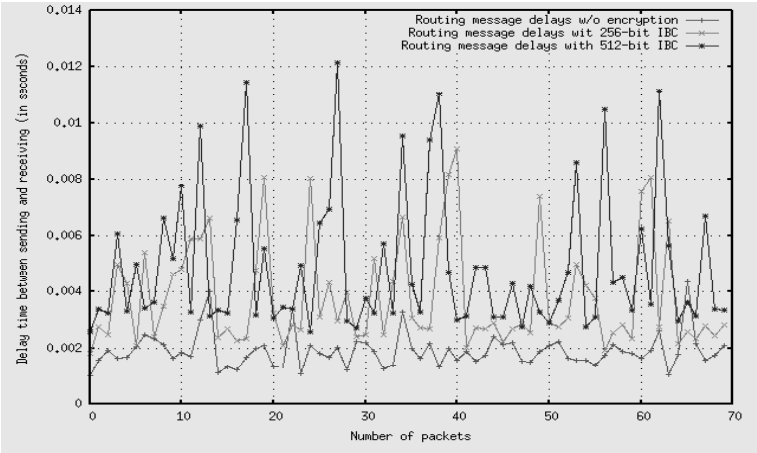
As a sidenote, we would like to show to interested readers the comparison between elliptic curve cryptographic operations and RSA: Generally, an elliptic curve whose order is a 160-bit prime offers approximately the same level of security as RSA with 1024-bit [71]. And the performance comparison is shown in Table- 6.3 [86].




(a) Delays in a 10-node network

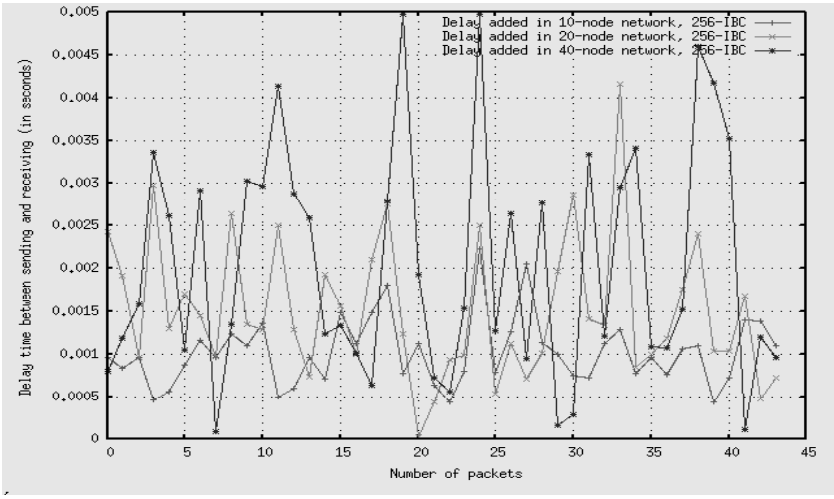



(b) Delays in a 20-node network

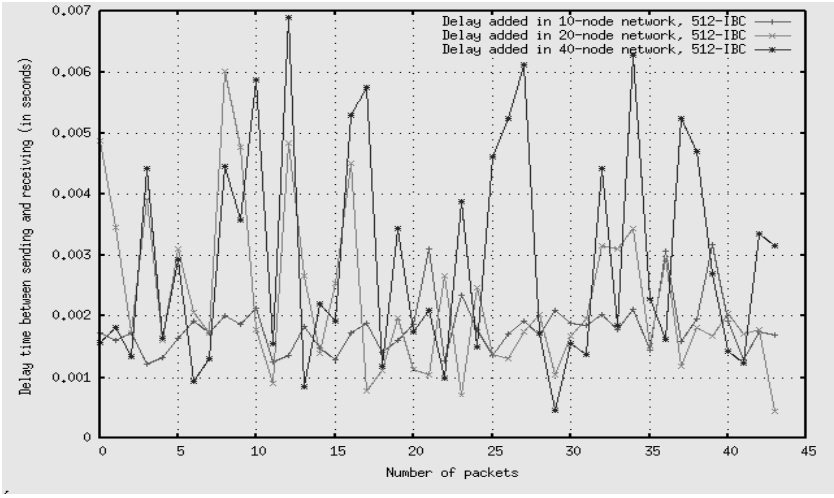



(c) Delays in a 40-node network

Figure 6.5: Comparison of End-to-end Delays of Routing Messages




(a) Delays added by 256-bit IBC operations




(b) Delays added by 512-bit IBC operations

Figure 6.6: Delays Added by IBC Operations

| **Systems** | **Key Generation** | **Signature** | **Verification** |
|---|---|---|---|
| **ECDSA** on $\mathbb{F}_{p192}$ | 5.5 | 6.3 | 26 |
| **DSA-1024** | 22.7 | 23.6 | 28.3 |
| **RSA-1024** | 1000 | 43.3 | 0.65 |

Table 6.3: Comparison of Performance of Elliptic Curve Cryptographic Operations, DSA and RSA (in milliseconds) [86]

## 6.5 Scalability Analysis

When the network size is increasing, there is more and more traffic to be transmitted and processed on each node, and thus performance will definitely decrease. This is concerned as scalability of a scheme. In the simulation of our KM-SR framework, we have simulated network size up to 40. It is infeasible to simulate a large-size network, and actually unnecessary if we can estimate and evaluate the impact of network size to the performance of the framework.

As our KM-SR framework is based on OLSR routing protocol, the computational and communication overhead brought by *HELLO* messages is proportional to the number of neighbors of a node, and not the number of nodes in the network. Only the computational and communication overhead brought by *TC* messages is proportional to number of nodes in the network. As is specified in OLSR protocol [23], *TC* message interval is 2.5 times of *HELLO* message interval. The number of neighbors of a node has a ceiling value (let's say $nb_{max}$), and does not change after the ceiling value is reached. The overall overhead changes much slower once $nb_{max}$ is reached.

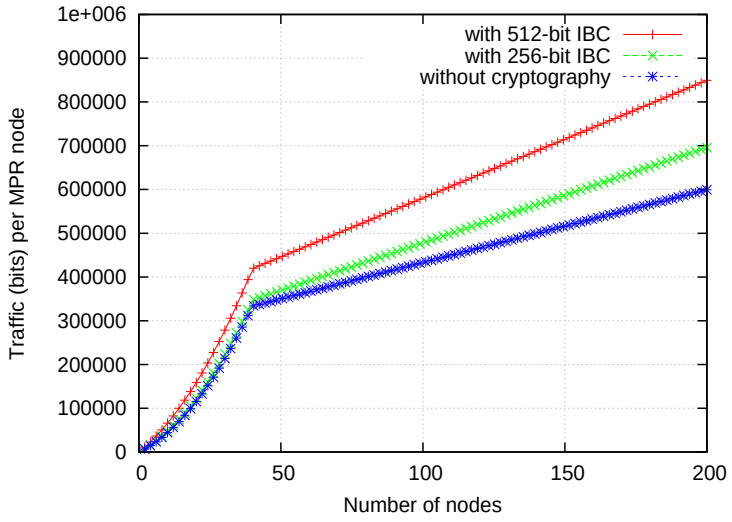



Figure 6.7: Traffic Model of the KM-SR Integrated Framework

We take the traffic amount to be received and processed by a MPR node to calculate computational and communication overhead of a node and to evaluate scalability of the scheme. This is the worst case because a MPR node forwards each *HELLO* message and *TC* message and has more traffic than normal nodes. When network size is less than $nb_{max}$, traffic to be received and processed by a MPR node is caused by both *HELLO*

messages and *TC* messages. Again we consider the worst case—assume all nodes to be neighbors. According to transmission overhead evaluation in Section 6.2, in a *TC* message interval, the traffic amount to be received and processed by a MPR node is about $(1514 + 40 \times n) \times n \times 2.5 + (1410 + 32 \times n) \times n = 5195 \times n + 132 \times n^2$ for 512-bit IBC, $(1002 + 40 \times n) \times n \times 2.5 + (898 + 32 \times n) \times n = 3403 \times +132 \times n^2$ for 256-bit IBC, and $(488 + 40 \times n) \times n \times 2.5 + (384 + 32 \times n) \times n$ = $3050 \times n + 132 \times n^2$ for standard OLSR without any cryptographic operation.

When network size is above $nb_{max}$, traffic to be received and processed by a MPR node comprises that from neighbor nodes (as calculated above) and that from non-neighbor nodes. Traffic from non-neighbor nodes is caused by only *TC* messages. The increased traffic amount from each non-neighbor node is $1410 + 32 \times nb_{max}$ for OLSR with 512-bit IBC, $898 + 32 \times nb_{max}$ for OLSR with 256-bit IBC, and $384 + 32 \times nb_{max}$ for standard OLSR. The number of non-neighbor nodes is $n - nb_{max}$. The overall traffic amount to be received and processed by a MPR node in a *TC* interval is therefore $C_1 + (1410 + 32 \times nb_{max}) \times (n - nb_{max})$ for OLSR with 512-bit IBC, $C_2 + (898 + 32 \times nb_{max}) \times (n - nb_{max})$ for OLSR with 256-bit IBC, and $C_3 + (384 + 32 \times nb_{max}) \times (n - nb_{max})$ for standard OLSR, where $C_1$, $C_2$ and $C_3$ are constants denoting the traffic amounts at $nb_{max}$ point for each case correspondingly.

Figure 6.7 shows the estimated traffic of a MPR node in a $TC$ interval of the framework for network size 0 to 200, with $nb_{max}$ set to 40. We collate this model with the previous simulation results and find that they match with each other. This model gives a rough picture of the scalability of the framework. From this model, the network designer can decide how many nodes are supported depending on bandwidth and processing power of nodes.

As a sidenote, we need to mention that all above calculation and estimation are based on IPv4 packets. When applied to IPv6, the length of a packet header and the length of a message header are increased by a constant, and message length is increased in proportion to number of neighbors advertised in the message. The transmission overhead model and traffic amount model do not change much. A curve just moves up a little as a whole.

## 6.6 Summary of the Chapter

This chapter analyzed performance of the proposed framework, demonstrated simulation results, and evaluated its scalability. In the next chapter we will present how to integrate solutions to limitations and weaknesses of IBC itself.

# Chapter 7

# Addressing Limitations and Weaknesses of IBC

So far, we have presented a novel IBC framework that addresses main issues of applying IBC to MANETs. IBC itself has some limitations and weaknesses. In this chapter, we propose solutions to these limitations and weaknesses. The reason we present these solutions in a separate chapter, instead of part of the framework, is mainly that although we start from the perspective of the KM-SR integrated framework, we aim to achieve solutions generic to all IBC schemes.

## 7.1 Addressing Identity Disclosure

We realize that the major vulnerability of MANETs, in contrast to wired networks, is lack of perimeter security due to dynamic topology and membership and wireless communication characteristics. In context of anonymous communication, the lack of perimeter security leads to the difficulty of specifying the anonymity set. For ad hoc networks that make use of use of an offline authority [65], we suggest using shared system secret to provide perimeter security which separate authentic nodes from adversary nodes, so that the anonymity set can be differentiated and protected. We propose some general-purpose identity (i.e. sender and receiver IP addresses in MANETs) hiding techniques that can be used in any routing protocol and also in any higher layer application in ad hoc networks. With an identity-protection key shared among authentic nodes, all real identities are hidden from adversaries. Network designers can apply these techniques in their routing protocols and applications with little modification and extension work.

If identities are simply encrypted using existing cryptosystems, the real identities are hidden to outsiders. However, adversaries can still apply traffic analysis and statistics to encrypted identities, and infer some information regarding the real identities. For example, an encrypted identity with a higher traffic in battlefield is more likely to be a commander than other nodes.

A working identity hiding scheme for MANETs should have the following properties:

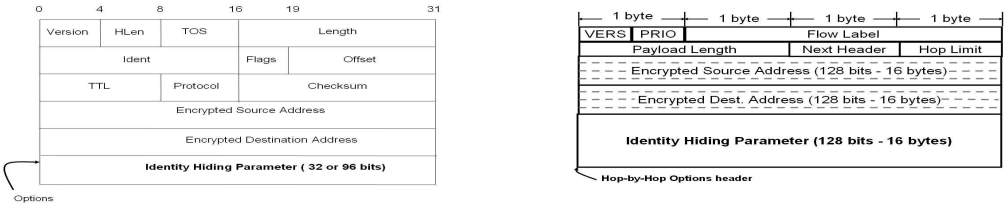
1. Identities in packets in the air should look totally random.
2. Encryption and decryption should be computable for authentic nodes: the sender, receiver, and nodes en route, but not computable for outsiders/eavesdroppers.
3. Packet format should not change, or the change should be compatible with existing routing protocols. In other words, it is better to have no extra field and no extra message added to a packet.

We adopt a commonly used assumption that the adversary in the network cannot attack below the network layer such as MAC (Medium Access Control) layer attack [56]. We consider this to be a basic feature of the wireless communication techniques used, for example, the IEEE 802.11 series. Many cryptosystems can be used to satisfy requirement 2, without breaking requirement 3 – just take the address as a binary input, and encrypt and decrypt it. But satisfying requirement 1 is not trivial, because regular encryption systems do not generate random output for the same input (the actual identity). Random output requires adding random portion to the fixed input.

The basic idea of our schemes is to encrypt the source and destination IP addresses with a random number using some popular cryptosystem, and transmit the random numbers in the IP header option field if there is no other space form them. We name the random number used “Identity Hiding Parameter”. In IPv4, the “Identity Hiding Parameter” can be placed in “options” field of the header. In IPv6, the “Identity Hiding Parameter” can be placed in “Hop-by-Hop Options header” of extension headers (value of the Next Header is ‘0’ in IPv6 header). Figure 7.1 illustrate the modified header and extension header in IPv4 and IPv6 packets.

### 7.1.1 AES-based Scheme

Standard AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits, and for specific input and key, the output is always the same. Consider applying the cryptosystem to IP addresses in MANETs, the input is fixed, and the key should also be fixed




(a) Identity Hiding in IPv4 Packets (b) Identity Hiding in IPv6 Packets

Figure 7.1: Identity Hiding in IPv4 and IPv6 Packets

and pre-distributed; but we want the output to be random. Thus, we need some revision or reorganization to AES cryptosystem.

**A Scheme for IPv4 Addresses:**

We distribute a 128-bit key $k$ to all authentic nodes when dispatching. An IPv4 address $m$ is 32 bits long. We generate a 96-bit random number $r$ for each packet. We append the random number to the IP address and encrypt it with a 128-bit AES cryptosystem: $c = AES_E(m + r, k)$. The first 32 bits of resulted 128-bit output are placed in the IP address, and the rest in the option field. An authentic node decrypts the original 128-bit plaintext as $m + r = AES_D(c, k)$, and gets the 32-bit address ($r$ is of no use now).

For example, we choose the key $k = a3b2c3d6f2c6e8561278164546cd3515$. Assume the IPv4 address to be encrypted is $m = 126.140.216.213$ ($7e8cd8d5$ in hex). In one packet, choose a random number as $r = 11000000a1c3b8d54e34156f$. Then,

$$\begin{aligned} c &= AES_E(m + r, k) \\ &= AES_E(7e8cd8d511000000a1c3b8d54e34156f, \\ &\qquad\qquad a3b2c3d6f2c6e8561278164546cd3515) \\ &= b7fcc8cdbdac86d7d8947729fbc6e919. \end{aligned}$$

The first 4 bytes $B7FCC8CD$ (i.e. 183.252.200.205) is placed in the IP address field, and

place $BDAC86D7D8947729FBC6E919$ in the option field of the header. To decrypt,

$$\begin{aligned} m + r &= AES_D(c, k) \\ &= AES_D(b7fcc8cdbdac86d7d8947729fbc6e919, \\ &\qquad a3b2c3d6f2c6e8561278164546cd3515) \\ &= 7e8cd8d511000000a1c3b8d54e34156f. \end{aligned}$$

The first 32 bits are the original IP address.

In another packet, choose a random number as $r = 11000000a1c3b8d54e34156e$. Then,

$$\begin{aligned} c &= AES_E(m + r, k) \\ &= AES_E(7e8cd8d511000000a1c3b8d54e34156e, \\ &\qquad a3b2c3d6f2c6e8561278164546cd3515) \\ &= 2f8748eb04b57c2fd89cdcf39075572c. \end{aligned}$$

The first 4 bytes $2f8748eb$ (i.e. 47.135.72.235) is placed in the IP address field, and place $04B57C2FD89CDCF39075572C$ in the option field of the header. Note that although the random numbers selected are only 1 bit different, the resulted encrypted addresses are different on every bit. To decrypt,

$$\begin{aligned} m + r &= AES_D(c, k) \\ &= AES_D(2f8748eb04b57c2fd89cdcf39075572c, \\ &\qquad a3b2c3d6f2c6e8561278164546cd3515) \\ &= 7e8cd8d511000000a1c3b8d54e34156f. \end{aligned}$$

The first 32 bits are the original IP address.

**A Scheme for IPv6 Addresses:**

We distribute a 128-bit key $k$ to all authentic nodes when dispatching. An IPv6 address $m$ is 128 bits long. We encrypt it for a packet in the following steps:

1. Generate a 128-bit random number $r$, and place it in the option field of extension header.

2. Using AES as blackbox, encrypt $k_1 = AES_E(r, k)$;

3. Using $k_1$ as key and AES as blackbox, calculate encrypted address $c = AES_E(m, k_1)$, and place $c$ in the address field.

Accordingly, we decrypt an encrypted address in the following steps:

1. Take the 128-bit random number $r$ from the option field of extension header.

2. Using AES as blackbox, encrypt $k_1 = AES_E(r, k)$;

3. Using $k_1$ as key and AES as blackbox, calculate decrypted address $m = AES_D(c, k_1)$.

For example, we use the same key as in Section 7.1.1. Assume the IPv6 address to be encrypted is $5F1B{:}DF00{:}CE3E{:}E200{:}0020{:}0800{:}2078{:}E3E3$. In one packet, choose a random number as $r = a115862311000000a1c3b8d54e34156f$. Then,

$$\begin{aligned} k_1 &= AES_E(r, k) = 18d1995807fd0ea5fc47717ef907a1c3, \\ c &= AES_E(m, k_1) \\ &= AES_E(5f1bdf00ce3ee200002008002078e3e3, \\ &\qquad 18d1995807fd0ea5fc47717ef907a1c3) \\ &= d7ae18429f9bb5e60914f5f092bafb45. \end{aligned}$$

$D7AE{:}1842{:}9F9B{:}B5E6{:}0914{:}F5F0{:}92BA{:}FB45$ is placed in the IP address field. To decrypt, a node calculates $k_1$ in same way as above.

$$\begin{aligned} m &= AES_D(c, k_1) \\ &= AES_D(d7ae18429f9bb5e60914f5f092bafb45, \\ &\qquad 18d1995807fd0ea5fc47717ef907a1c3) \\ &= 5f1bdf00ce3ee200002008002078e3e3. \end{aligned}$$

In another packet, we choose a different random number as $r = a115862311000000a1c3b8d54e34156e$. Then,

$$\begin{aligned} k_1 &= AES_E(r,k) = 3512de51fd9903c26f03c86e2dfa507b, \\ c &= AES_E(m,k_1) \\ &= AES_E(5f1bdf00ce3ee200002008002078e3e3, \\ &\qquad 3512de51fd9903c26f03c86e2dfa507b) \\ &= 8cd1b2cd2a33b98bd9d4d78bc89e7c71. \end{aligned}$$

$8CD1{:}B2CD{:}2A33{:}B98B{:}D9D4{:}D78B{:}C89E{:}7C71$ is placed in the IP address field.

To decrypt,

$$\begin{aligned} m &= AES_D(c,k_1) \\ &= AES_D(8cd1b2cd2a33b98bd9d4d78bc89e7c71, \\ &\qquad 3512de51fd9903c26f03c86e2dfa507b) \\ &= 5f1bdf00ce3ee200002008002078e3e3. \end{aligned}$$

### 7.1.2 RSA-based Scheme

**Algorithm:**

The standard RSA algorithm produces fixed output for a fixed input, so we modify it as follows:

1. The system administrator chooses secret primes $p$ and $q$ and computes $n = pq$, $\phi(n) = (p-1)(q-1)$. $n$ and $\phi(n)$ are distributed to authentic nodes before dispatching.

2. For each packet, node $A$ chooses a random $e$ with $gcd(e, \phi(n)) = 1$, and encrypts IP address $m$ in the packet as $c \equiv m^e \pmod{n}$. $c$ and $e$ are sent in the packet.

3. An authentic node computes $d \equiv e^{-1} \pmod{\phi(n)}$, upon receiving the packet, and decrypts $c$ by $m \equiv c^d \pmod{n}$.

The security foundation of the algorithm is the same as the RSA, namely the *Integer Factorization* problem: the modulus $n$ and encryption key $e$ are made public; only those who know the factorization of $n$ can calculate $\phi(n)$ and the decryption key $d$. In a network system, the factorization of $n$ is a system wide secret (in case $n$ is not large enough, even

$n$ is secret), but $e$ changes for each packet, thus only authentic nodes can calculate $d$ and get the real identities.

**Applying to IPv4 Packets:**

An IPv4 address is 32-bit long, thus the transformed address is also 32-bit. For simplicity, we show the case in which the modulus $n$ is a 32-bit integer. Accordingly, the largest length of $e$ is also 32 bits, or 4 bytes. In the IPv4 header options, one byte of "type" and one byte of "length" are required, and 2 bytes padding is required for alignment, so the total length of the option is 8 bytes.

For example, let $p = 65479$ (a prime number), $q = 64591$ (a prime number), then $n = p \cdot q = 65479 \cdot 64591 = 4229354089$, $\phi(n) = (p-1) \cdot (q-1) = 65478 \cdot 64590 = 4229224020$. Assume the IPv4 address to be encrypted is $224.0.0.0$ ($m = 3758096384$ in decimal). In one IP packet, we choose the encryption key $e$ randomly as $e = 9007$. The encrypted address is $c \equiv m^e \bmod n \equiv 3758096384^{9007} \equiv 188128951 \pmod{4229354089}$, i.e. $11.54.158.183$. The encrypted address is placed in the IP address field of the packet. $\phi(n)$ is shared secret among authentic nodes, $e$ is transmitted in the header option of the IP packet; so an authentic node can calculate the decryption key $d \equiv e^{-1} \bmod \phi(n) \equiv 9007^{-1} \equiv 1317553303 \pmod{4229224020}$, and then decrypt the IP address as $m \equiv c^d \bmod n \equiv 188128951^{1317553303} \equiv 3758096384 \pmod{4229354089}$.

In another packet, $e$ is chosen another number. Assume $e = 9011$, then the encrypted address is $c \equiv m^e \bmod n \equiv 3758096384^{9011} \equiv 2953896956 \pmod{4229354089}$, i.e. $176.16.227.252$. An authentic node can calculate the decryption key $d \equiv e^{-1} \bmod \phi(n) = 9011^{-1} \equiv 829324031 \pmod{4229224020}$, and then decrypt the IP address as $m \equiv c^d \bmod n \equiv 2953896956^{829324031} \equiv 3758096384 \pmod{4229354089}$.

**Applying to IPv6 Packets:**

An IPv6 address is 128 bits long, thus the transformed address is also 128-bit. For simplicity, we show the case in which the modulus $n$ is a 128-bit integer. Accordingly, the largest length of $e$ is also 128 bits, or 16 bytes. We place $e$ in the Hop-by-hop option header immediately after IPv6 header (Next header = 0 in IPv6 header). One byte of "type" and one byte of "length" are required, and 6 bytes padding is required for alignment, so the total length of the option is 24 bytes.

For example, let $p = 18446744073709551557$ (a prime number), $q = 18446744073709$

551533 (a prime number), then

$$\begin{aligned} n = p \cdot q &= 340282366920938460843936948965011886881 \text{(less than } 2^{128}\text{)}, \\ \phi(n) = (p-1) \cdot (q-1) &= 18446744073709551556 \cdot 18446744073709551532 \\ &= 340282366920938460807043460817592783792. \end{aligned}$$

Assume the IPv6 address to be encrypted is $5F1B{:}DF00{:}CE3E{:}E200{:}0020{:}0800{:}2078{:}E3E3$ ($m = 126421374655918995273183870066405991395$ in decimal). In one IP packet, we choose the encryption key $e$ randomly as $e = 9007$. The encrypted address is

$$\begin{aligned} c \equiv m^e &\text{mod } n \equiv 126421374655918995273183870066405991395^{9007} \\ &\equiv 259104061544288900162192625529409559801 \\ &(\text{mod } 340282366920938460843936948965011886881), \end{aligned}$$

i.e. $C2ED{:}A088{:}948E{:}4635{:}0589{:}880B{:}F6E8{:}DCF9$. The encrypted address is placed in the IP address field of the packet. $\phi(n)$ is shared secret among authentic nodes, $e$ is transmitted in the extension header of the IP packet. An authentic node can calculate the decryption key

$$\begin{aligned} d \equiv e^{-1} &\text{mod } \phi(n) \equiv 9007^{-1} \\ &\equiv 106010027931625771180699894643517858479 \\ &(\text{mod } 340282366920938460807043460817592783792), \end{aligned}$$

and then decrypt the IP address as

$$\begin{aligned} m \equiv c^d &\text{mod } n \\ &\equiv 259104061544288900162192625529409559801\hat{} \\ &\quad 106010027931625771180699894643517858479 \\ &\equiv 126421374655918995273183870066405991395 \\ &(\text{mod } 340282366920938460843936948965011886881). \end{aligned}$$

In another packet, $e$ is chosen another random number. Assume $e = 9011$, then the

encrypted address is

$$\begin{aligned} c \equiv m^e \bmod n &\equiv 126421374655918995273183870066405991395^{9011} \\ &\equiv 218386055390514149206611613922146001358 \\ &\pmod{340282366920938460843936948965011886881}, \end{aligned}$$

i.e. $A44B{:}9FD5{:}7CF4{:}ECE5{:}339C{:}B448{:}E0CA{:}21CE$. An authentic node can calculate the decryption key

$$\begin{aligned} d \equiv e^{-1} \bmod \phi(n) &\equiv 9011^{-1} \\ &\equiv 263208089827870499592175443557720752 75 \\ &\pmod{340282366920938460807043460817592783792}, \end{aligned}$$

and then decrypt the IP address as

$$\begin{aligned} m \equiv c^d \bmod n & \\ \equiv 218386055390514149206611613922146001358^{\wedge} & \\ 263208089827870499592175443557720752 75 & \\ \equiv 126421374655918995273183870066405991395 & \\ \pmod{340282366920938460843936948965011886881}. & \end{aligned}$$

### 7.1.3 ElGamal-based Scheme

**Algorithm:**

In ElGamal algorithm, there is a random number used for each message, and it is an essential requirement that the random number used each time be different. This means the ElGamal is already very close to the requirements of an identity hiding scheme. The following algorithm is very similar to standard ElGamal algorithm, the only difference being in the key distribution:

1. The system administrator chooses a large primes $n$, a primitive root $g$, and a random integer $k$ (less than $n$). $n$, $g$ and $k$ are distributed to authentic nodes before dispatching.

2. For each packet, node $A$ chooses a random integer $r$, and encrypts IP address $m$ in the packet as $c \equiv m \cdot g^{k^r} \pmod{n}$. $c$ and $g^r$ are sent in the packet.

3. An authentic node, upon receiving the packet, decrypts $c$ by $m \equiv c/g^{rk} \pmod{n}$.

**Applying to IPv4 Packets:**

Again, for simplicity, we show the case in which the modulus $n$ is a 32-bit integer. Accordingly, the largest length of $k$ is also 32 bits, or 4 bytes. The total length of the option in the IP header is 8 bytes, taking consideration of one byte of "type", one byte of "length", and 2 bytes of padding.

For example, let $n = 4294967291$ (a prime number less than $2^{32}$), and $g = 2$ (a primitive root of $n$). Choose the key as $k = 59$. Assume the IPv4 address to be encrypted is $126.140.216.213$ ($m = 2123159765$ in decimal). In one IP packet, we choose the random number $r$ randomly as $r = 1798435485$ (less than $n$). The encrypted address is $c \equiv m \cdot g^{k^r} \bmod n \equiv 2123159765 \cdot 2^{59^{1798435485}} \equiv 4201974492 \pmod{4294967291}$, i.e. $250.117.10.220$. The encrypted address is placed in the IP address field of the packet. $n, g$ and $k$ are shared secrets among authentic nodes, $g^r$ is transmitted in the header option of the IP packet. So an authentic node can decrypt the IP address as $m \equiv c/g^{rk} \bmod n \equiv 4201974492/2^{17984354859^{59}} \equiv 2123159765 \pmod{4294967291}$.

In another packet, we choose the random number $r$ as $r = 7586249853$ (less than $n$). The encrypted address is $c \equiv m \cdot g^{k^r} \bmod n \equiv 2123159765 \cdot 2^{59^{7586249853}} \equiv 1275954819 \pmod{4294967291}$, i.e. $76.13.134.131$. The encrypted address is placed in the IP address field of the packet. An authentic node can decrypt the IP address as $m \equiv c/g^{rk} \bmod n \equiv 1275954819/2^{758624985359} \equiv 2123159765 \pmod{4294967291}$.

**Applying to IPv6 Packets:**

Again, for simplicity, we show the case in which the modulus $n$ is a 128-bit integer. Accordingly, the largest length of $k$ is also 128 bits, or 16 bytes. We place $k$ in the Hop-by-hop option header immediately after IPv6 header (Next header = 0 in IPv6 header). The total length of the option is 24 bytes, considering one byte of "type", one byte of "length", and 6 bytes padding is required for alignment.

For example, let $n = 340282366920938463463374607431768211297$ (a prime number less than $2^{128}$), and $g = 5$ (a primitive root of $n$). Choose the key as $k = 59$ (less than $n$). Assume the IPv6 address to be encrypted is $5F1B{:}DF00{:}CE3E{:}E200{:}0020{:}0800{:}2078{:}E3E3$ ($m = 126421374655918995273183870066405991395$ in decimal). In one IP packet, we choose the random number $r$ randomly as $r = 1798435485$ (less than $n$).

The encrypted address is

$$\begin{aligned} c &\equiv m \cdot g^{k^r} \bmod n \\ &\equiv 126421374655918995273183870066405991395 \cdot \\ &\quad 5^{59^{1798435485}} \\ &\equiv 113272238072654985588029568175975497 80 \\ &\pmod{340282366920938460843936948965011886881}, \end{aligned}$$

i.e. $0885{:}8B40{:}6B7B{:}1D07{:}13B3{:}B3F9{:}5B19{:}4CD4$. The encrypted address is placed in the IP address field of the packet. $n, g$ and $k$ are shared secrets among authentic nodes, $g^r$ is transmitted in the header option of the IP packet. An authentic node can decrypt the IP address as

$$\begin{aligned} m &\equiv c/g^{rk} \bmod n \\ &\equiv 113272238072654985588029568175975497 80/ \\ &\quad 5^{1798435485^{59}} \\ &\equiv 126421374655918995273183870066405991395 \\ &\pmod{340282366920938460843936948965011886881} \end{aligned}$$

In another packet, we choose the random number $r$ randomly as $r = 126346546799798$. The encrypted address is

$$\begin{aligned} c &\equiv m \cdot g^{k^r} \bmod n \\ &\equiv 126421374655918995273183870066405991395 \cdot \\ &\quad 5^{59^{126346546799798}} \\ &\equiv 119086206211554693843012791568616584700 \\ &\pmod{340282366920938460843936948965011886881}, \end{aligned}$$

i.e. $5997{:}2B46{:}E6F1{:}FB19{:}4FAE{:}A0D6{:}26EE{:}F5FC$. The encrypted address is placed in the IP address field of the packet. An authentic node can decrypt the IP ad-

dress as

$$\begin{aligned} m &\equiv c/g^{rk} \bmod n \\ &\equiv 11908620621155469384301279156861658470 0/ \\ &\quad 5^{126346546799798^{59}} \\ &\equiv 126421374655918995273183870066405991395 \\ &\pmod{340282366920938460843936948965011886881} \end{aligned}$$

### 7.1.4 Comparison of the Schemes

The above *RSA*-based and *ElGamal*-based schemes require a modulus number. Because there are some requirements for choosing this modulus number specified by the algorithms, it cannot be exactly equal to the maximum IP address. If this modulus number is greater than the maximum IP address, the length of the resulted address may also be longer than standard length, and there will be definitely extra field for extra bits. If this modulus number is less than the maximum IP address, it cannot cover all IP addresses. Fortunately, large addresses are reserved for future use in IP standards, and are not really used currently. If for any reason these addresses are to be used, we can extend the address field by 1 extra byte—using 40-bit modulus for IPv4 and 136-bit for IPv6 addresses—and accommodate the transformed address in the "enlarged" address field—putting the extra byte to the "Identity Hiding Parameter" field. In contrast, the *AES*-based scheme does not have this limitation, because the last step is not modulus operation, and thus all values representable by the address field can be encrypted.

Another concern for compatibility is the option field or extension header that requires implementation on all hosts and routers in a network. We notice that for *RSA*-based and *ElGamal*-based scheme, it is not trivial to avoid the option field or extension header, because in both of them, a random number is needed to accompany a packet to decrypt the hidden address. In *RSA*-based scheme, the security level is dependent on the length of the random number. If the random number $e$ in the packet is small, it is subject to cryptoanalysis. In *ElGamal*-based scheme, the random number $g^r$ in the packet is always of length defined by modulus $n$.

For *AES*-based scheme, security level is not dependent on the length of the random number; so we can integrate a short random number (we name it "*scrambler*") into the address field. This is applicable because a MANET is not directly connected to Internet, so we really do not need a 32-bit address field. In most cases, a 16-bit or 24-bit address

is big enough for a MANET. For example a 16-bit IP address can accommodate 65,536 hosts; so we have space for a 16-bit *scrambler*. In this way, the IP address itself contains a random number, but the randomness only happens on fixed bits and we need to diffuse it to the whole field. If we pass the transformed IP address to an *AES* cryptosystem, the output is a random number on every bit. For a 128-bit IPv6 address, we can pass it to a standard 128-bit *AES* block cipher. For a 32-bit IPv4 address, we can use *AES* in *MR_CFB* (Cipher Feed-Back) mode as a stream cipher to diffuse the scrambler bits.

Figure 7.2 shows the simulation result of processing IPv4 addresses with *AES* using 8-bit *scrambler* in 250 packets. An original IP address 192.168.0.1 is scrambled randomly to different addresses in the range [0.0.0.0, 255.255.255.255].

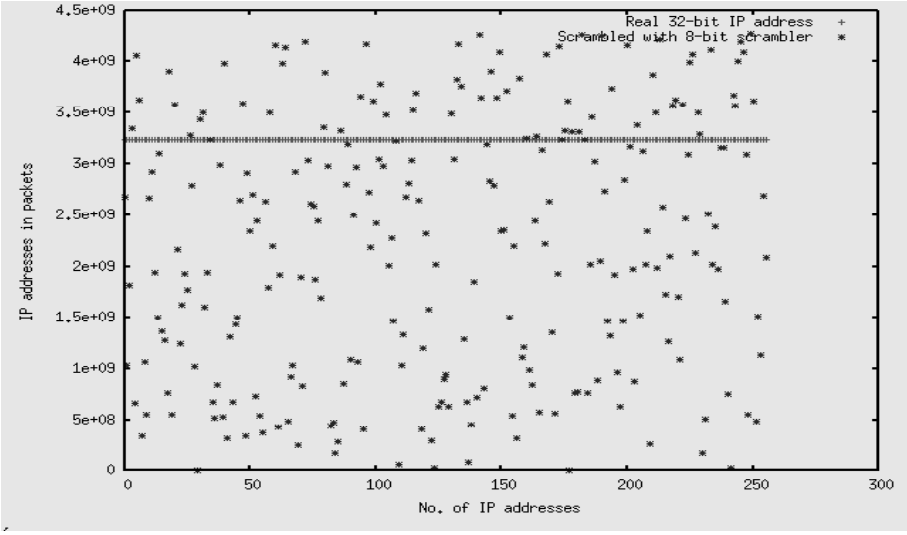



Figure 7.2: Scrambled IP Addresses

Compared to previous proposals in literature, our scheme has the following advantages: while previous proposals seem to focus on certain routing protocols, our scheme can be applied not only in any routing protocol, but also in any protocol or application on higher layers; while previous proposals use either pairwise keys or a large number of pseudonyms, our scheme only relies on a small number of pre-distributed parameters. The idea of this scheme is based on our notion of achieving all-around perimeter security of MANETs by pre-distributed system wide secrets, and is applicable to MANETs with offline authority, such as communication systems for public safety, emergency and disaster applications.

## 7.2 Addressing Key Revocation Difficulty

Combined with the previous identity-hiding scheme, we propose a key revocation scheme. To facilitate identity revocation, we divide IP addresses into two categories: *Long-term addresses* that do not expire and are implicitly valid until explicit revocation, and *Short-term addresses* that expire after a while until explicit renewal. The first type of address ends with bit $0$; the second type ends with $1$. A long-term address has 16 bits scrambler. A short-term address has 8 bits scrambler and 8 bits *validity counter*. (This is a general scheme. The length can be adjusted as will be discussed shortly).

*Validity counter* can be used in several ways, for example:

- Indicating the count of packets sent from this address: *Validity counter* of an address decrements each time when a packet is sent from the address. Correspondingly, each node en route maintains in the routing table the value of *validity counter* for each short-term address, and checks this value in each packet. If this value is not decremented or reaches 0, the packet is discarded. Note that in a packet, the encrypted address is used, and is decrypted upon reception.

- Indicating time span of validity of this address: *Validity counter* indicates number of days, or number of hours, etc. the address is valid after first use. Other nodes calculate the expiry time when they first receive a packet from this address, store the expiry time for each node, and check against the expiry time each time they receive a packet from this address. If the expiry time is reached, no packet is accepted any more.

When a short-term identity expires, the administrator has the right to renew its validity by broadcasting an *Identity Renewal Message*. The *Identity Renewal Message* contains the addresses to be renewed, the new *validity counter* of each address, the timestamp, and is finally signed by the administrator (Sign and verification processes are similar as those for *entrance ticket* explained earlier in Section 5.2). The network nodes update their *validity counter* table according to the *Identity Renewal Message*.

When a node with short-term or long-term identity is compromised or dismissed, or a node with short-term has finished its task ahead of schedule, the administrator can revoke its identity at any time needed by broadcasting an *Identity Revocation Message*. The *Identity Revocation Message* contains the addresses to be revoked, the timestamp, and is signed by the administrator. Each network node maintains a revocation list according to

the *Identity Revocation Message*. The node checks the identity of each packet it receives against the revocation list and discards the packet with an identity in the list.

Although the *Identity Renewal Message* and *Identity Revocation Message* mechanism is no better than existing schemes, the use of *Validity counter* minimizes the need of identity renewal and identity revocation, which provides a higher reliability and saves much traffic overhead. Since all identities are encrypted, this decreases the possibility of an identity being stolen or impersonated by adversaries. For short-term identities, the length of *validity counter* field can be adjusted according to specific task so that the validity expiration keeps step with task progress. With the limitation of *validity counter*, the benefit an adversary stealing an identity is limited. Thus the significance of revocation is also decreased—even the identity is stolen by adversaries, there is very limited room they can do with it. For example, in a network with 1000 nodes, we assume 200 of them are short-term nodes and 800 are long-term nodes, among long-term nodes 10% are detected compromised later, and there is no synchronization mechanism. With previous schemes, 280 revocation messages need to be sent. With our scheme, if the *validity counter* is set to proper length, the number of revocation messages is only 80.

## 7.3 Addressing Key Escrow

The scheme we proposed for prevention of mobile attacks in Section 5.2 also addresses the key escrow issue. With the KM-SR integrated framework, after initialization phase, the nodes can generate pairwise secret key that is key-escrow free, and contribute to a dynamic part of master key. The new master key is comprised of a dynamic part and a static part, and thus is unknown both to the offline administrator and to online nodes, since the offline administrator does not know the dynamic part which is kept secret by a threshold number of online nodes, and the static part is kept secret to online nodes. The new private key of a node is known only to the node. The new system public key is only known to the online nodes. The administrator does not have the master key without cooperation of other nodes. Thus, the online nodes can communicate with each other confidential to the offline administrator, but online nodes can still communicate with offline administrator using the initial static keys.

## 7.4 Summary of the Chapter

This chapter presented our solutions to address limitations and weaknesses of IBC itself, namely the identity disclosure problem, key revocation difficulty problem, and key escrow problem. These solutions can be used in other IBC schemes, as well as the proposed KM-SR integrated framework.

# Chapter 8

# Conclusions and Future Work

## 8.1 Conclusions

Security of MANETs is still a challenging problem due to specific features of MANETs discussed in Section 2.2. This thesis reviews requirements and solutions of this problem. Traditional cryptographic solutions use either symmetric or asymmetric cryptography. Both of them have many limitations when used in MANETs. Identity-based Cryptography, as a new asymmetric cryptography technology, has many advantages when considered in the context of a MANET, and is the most promising technology for MANET security (Section 2.3). Concentrated on IBC solutions, this research studies existing applications of IBC in MANETs, and points out issues of applying this technology to MANET security (Chapter 3). The biggest issue preventing these solutions from being used in practical applications is the interdependency cycle between secure routing and security services, for instance key management. Other issues include using IBC in the same way as traditional CBC and disabling advantages of IBC, insecurity of key generation process, vulnerability to Sybil attacks, mobile attacks, proximity-caused insecurity etc.

In light of such issues, a novel KM-SR integrated framework for MANET security is proposed in this thesis (Chapter 4). The proposed framework addresses key management and secure routing interdependency cycle problems of IBC. This framework brings these contributions: compared to symmetric key solutions, it has more functionality derived from asymmetric keys, and is more secure due to using 1-to-$m$ broadcasting key instead of only 1 group broadcasting key, and has less keys to store per node due to using asymmetric keys instead of pairwise symmetric keys; compared to CBC solutions, the storage and communication requirements are lower due to IBC properties; compared to previous IBC

solutions, it has no KM-SR interdependency cycle problem, and is immune to insider attacks and mobile attacks and many other routing attacks.

We implemented the framework and simulated it in NS-2. Simulation results show that performance of the framework is comparable to, or better than, existing schemes (Chapter 6).

We prove the security of the encryption/signature scheme used in the framework in the random oracle model (Chapter 5). Based on the security of the scheme, these security features of the framework are ensured: confidentiality, integrity, authentication, freshness, and non-repudiation. These security features lend the framework capability to defend against most known attacks on network layers, such as spoofing and Sybil attacks, eavesdropping and traffic analysis, modifying routing packets, etc. With some extra enhancements, the framework can counter some other attacks, such as record and replay attacks, wormhole attacks, blackhole attacks, and mobile attacks.

IBC itself has some limitations and weaknesses, such as key escrow, key revocation difficulty, and identity disclosure. We propose solutions to address these limitations and weaknesses in this framework (Chapter 7).

The result of this work presents a feasible security solution to a wide range of MANETs where there is an administrator that generates and distributes initial system parameters to all nodes, and the administrator can authenticate the identity of a node and assign initial private key to it. Basically this includes all MANETs where IBC is applicable, with an extra requirement—a controlled deployment phase. Examples of this type of MANETs include but are not limited to: sensor networks, wearable computer systems in military, public safety networks, and emergency and disaster rescue teams.

## 8.2 Limitations and Future Work

The framework provides a security solution for MANETs mainly on network layer, on the assumption that link/physical layer security are already provided. If that is not the case, the framework may fail to work. Security on transport layer and application layer is not considered in this framework. This means that while secure routing assures secure applications to run in the network, malicious applications can also enjoy this service without any penalty. Security on application layer needs to be considered on a per-application basis, and network layer security does not provide any help on this.

We notice that the framework is not immune to denial-of-service attacks and selfishness of inside nodes. These availability related issues cannot be addressed with a crypto-

graphic solution, but need a monitoring and accusation system. The framework provides a solid bedrock for this system, but itself does not implement the functionality.

Identity-based cryptography itself is under continuing study and development. On the one hand, algorithms and implementations of higher efficiency and security level are being developed; while on the other hand, security flaws are being discovered—some of them are fixed and some others remain open. The selection of curves on which IBC is implemented is very subtle and critical to efficiency and security. There are some controversial discussions on the security of this cryptography, and it is not yet widely accepted. We need to keep track of the latest developments in IBC.

We intend to consider these topics as future work in our research.

# Bibliography


[1] ABUSALAH, L., KHOKHAR, A., AND GUIZANI, M. A survey of secure mobile ad hoc routing protocols. *Communications Surveys & Tutorials, IEEE 10*, 4 (2008), 78–93.

[2] ADJIH, C., RAFFO, D., AND MUHLETHALER, P. Attacks against olsr: Distributed key management for security. In *Proc. OLSR Interop and Workshop* (2005), INRIA, pp. 1–6.

[3] ANJUM, F., AND MOUCHTARIS, P. *Security for Wireless Ad Hoc Networks*. Wiley Inter-Science, 2007.

[4] BALFANZ, D., SMETTERS, D. K., STEWART, P., AND WONG, H. C. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proc. Symp. Network and Distributed Systems Security* (2002), Internet Society, pp. 23–35.

[5] BOBBA, R., ESCHENAUER, L., GLIGOR, V., AND ARBAUGH, W. Bootstrapping security associations for routing in mobile ad-hoc networks. In *Proc. Global Telecommunications Conference* (2003), IEEE, pp. 1511–1515.

[6] BOHIO, M., AND MIRI, A. Authenticated secure communications in mobile ad hoc networks. In *Proc. Canadian Conference on Electrical and Computer Engineering* (2004), IEEE, pp. 1689–1692.

[7] BOHIO, M. J., AND MIRI, A. An authenticated broadcasting scheme for wireless ad hoc network. In *Proc. CNSR 2004* (2004), IEEE Computer Society, pp. 69–74.

[8] BOHIO, M. J., AND MIRI, A. Efficient identity-based security schemes for ad hoc network routing protocols. *J. Ad Hoc Networks 2*, 3 (2004), 309–317.

[9] BOLDYREVA, A. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In *Proc. 6th International Work-*

*shop on Theory and Practice in Public Key Cryptography* (2003), Springer, pp. 31–46.

[10] BONEH, AND FRANKLIN. Identity-based encryption from the weil pairing. In *Proc. Crypto 2001* (2001), LNCS, Springer, pp. 213–219.

[11] BONEH, D., AND BOYEN, X. Efficient selective-id secure identity based encryption without random oracles. In *Proc. EUROCRYPT* (2004), LNCS, Springer, pp. 223 – 238.

[12] BONEH, D., GENTRY, C., LYNN, B., AND SHACHAM, H. Aggregate and verifiably encrypted signatures from bilinear maps. In *Proc. 22nd International Conference on Theory and Applications of Cryptographic Techniques* (2003), LNCS, Springer, pp. 416–432.

[13] BONEH, D., LYNN, B., AND SHACHAM, H. Short signatures from the weil pairing. In *Proc. ASIACRYPT* (2001), LNCS, Springer, pp. 514–532.

[14] BONEH, D., LYNN, B., AND SHACHAM, H. Short signatures from the weil pairing. *J. Cryptol. 17* (September 2004), 297–319.

[15] BOYD, MAO, AND PATERSON. Key agreement using statically keyed authenticators. In *Proc. International Conference on Applied Cryptography and Network Security* (2004), LNCS, Springer, pp. 248–262.

[16] BOYEN, X. Multipurpose identity-based signcryption: A swiss army knife for identity-based cryptography. In *Proc. Crypto 2003* (2003), LNCS, Springer, pp. 383–399.

[17] BUCHEGGER, S., AND LE BOUDEC, J.-Y. Performance analysis of the confidant protocol. In *Proc. 3rd ACM international symposium on Mobile ad hoc networking & computing* (2002), MobiHoc '02, ACM, pp. 226–236.

[18] CHA, J., AND CHEON, J. An identity-based signature from gap diffie-hellman groups. In *Proc. International Workshop on Practice and Theory in Public Key Cryptography* (2003), LNCS, Springer, pp. 18–30.

[19] CHAKERES, I. D., AND MACKER, J. P. Mobile ad hoc networking and the IETF. *J. SIGMOBILE Mob. Comput. Commun. Rev. 10* (2006), 58–60.

[20] CHEN, L., AND KUDLA, C. Identity based authenticated key agreement protocols from pairings. Tech. Rep. HPL-2003-25, Hewlett Packard Laboratories, Feb. 12 2003.

[21] CHIEN, H.-Y., AND LIN, R.-Y. Identity-based key agreement protocol for mobile ad-hoc networks using bilinear pairing. In *Proc. Sensor Networks, Ubiquitous, and Trustworthy Computing* (2006), IEEE, pp. 520–529.

[22] CHIEN, H.-Y., AND LIN, R.-Y. Improved id-based security framework for ad hoc network. *Ad Hoc Netw. 6*, 1 (2008), 47–60.

[23] CLAUSEN, T., AND JACQUET, P. RFC3626 - Optimized Link State Routing Protocol (OLSR), 2003.

[24] COCKS, C. An identity based encryption scheme based on quadratic residues. In *Proc. IMA Conference on Cryptography and Coding* (2001), LNCS, Springer, pp. 360–363.

[25] CORSON, S., AND MACKER, J. RFC2501 - Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, 1999.

[26] CREPEAU, C., AND DAVIS, C. A certificate revocation scheme for wireless ad hoc networks. In *Proc. 1st workshop on Security of ad hoc and sensor networks* (2003), SASN '03, ACM, pp. 54–61.

[27] DELERABLÉE, C. Identity-based broadcast encryption with constant size ciphertexts and private keys. In *Proc. ASIACRYPT'07* (2007), Springer, pp. 200–215.

[28] DENG, H., AND AGRAWAL, D. P. TIDS: threshold and identity-based security scheme for wireless ad hoc networks. *Ad Hoc Networks 2*, 3 (2004), 291–307.

[29] DENG, H., MUKHERJEE, A., AND AGRAWAL, D. P. Threshold and identity-based key management and authentication for wireless ad hoc networks. In *Proc. ITCC* (2004), IEEE, pp. 107–111.

[30] DESMEDT, AND QUISQUATER. Public-key systems based on the difficulty of tampering (is there a difference between DES and RSA?) (extended abstract). In *Proc. Crypto* (1987), Springer, pp. 111–117.

[31] DESMEDT, Y. Threshold cryptography. *European Transactions on Telecommunications 5,* 4 (July – Aug. 1994), 449–457.

[32] DESMEDT, Y., AND FRANKEL, Y. Threshold cryptosystems. In *Proc. Advances in cryptology* (1989), CRYPTO '89, Springer, pp. 307–315.

[33] DU, X., WANG, Y., GE, J., AND WANG, Y. Id-based authenticated two round multi-party key agreement. Cryptology ePrint Archive, Report 2003/247, 2003. `http://eprint.iacr.org/`.

[34] DUTTA, R., BARUA, R., AND SARKAR, P. Pairing-based cryptographic protocols: A survey. Cryptology ePrint Archive, Report 2004/064, June 24 2004. `http://eprint.iacr.org/`.

[35] FANG, Y., ZHU, X., AND ZHANG, Y. Securing resource-constrained wireless ad hoc networks. *Wireless Commun. 16,* 2 (2009), 24–29.

[36] FELDMAN, P. A practical scheme for non-interactive verifiable secret sharing. In *28th Symposium on Foundations of Computer Science* (1987), IEEE, pp. 427–437.

[37] FREEBERSYSER, J. A. A DoD perspective on Mobile Ad Hoc Networks. *J. Ad hoc networking* (2001), 29–74.

[38] GENTRY. Certificate-based encryption and the certificate revocation problem. In *Proc. EUROCRYPT* (2003), LNCS, Springer, pp. 272–293.

[39] GENTRY, G., AND SILVERBERG, A. Hierarchical ID-based cryptography. In *Proc. ASIACRYPT* (2002), LNCS, Springer, pp. 548–566.

[40] GOLDWASSER, S., MICALI, S., AND RIVEST, R. L. A digital signature scheme secure against adaptive chosen-message attacks. *J. SIAM Comput. 17* (April 1988), 281–308.

[41] GUAN, Y., FU, X., XUAN, D., SHENOY, P., BETTATI, R., AND ZHAO, W. NetCamo: Camouflaging Network Traffic for QoS-Guaranteed Mission Critical Applications. *J. Transactions on Systems, Man, and Cybernetics 31,* 4 (2001), 253–265.

[42] HANSEN, S., WARREN, G., SEGRAVES, B., AND BANE, C. Modeling and simulation analyses of the Joint Tactical Radio System(JTRS). In *Proc. Aerospace Conference* (2001), IEEE, pp. 1065–1074.

[43] HEGLAND, A. M., WINJUM, E., MJOLSNES, S. F., RONG, C., KURE, O., AND SPILLING, P. A survey of key management in ad hoc networks. *J. Communications Surveys & Tutorials, IEEE 8*, 3 (2006), 48–66.

[44] HESS, F. Efficient identity based signature schemes based on pairings. In *Proc. SAC: Annual International Workshop on Selected Areas in Cryptography* (2003), LNCS, Springer, pp. 310–324.

[45] HOEPER, K., AND GONG, G. Identity-based key exchange protocols for ad hoc networks. In *Proc. Canadian Workshop on Information Theory* (2005), CWIT, pp. 127–130.

[46] HOEPER, K., AND GONG, G. Bootstrapping security in mobile ad hoc networks using identity-based schemes with key revocation. Report 2006-04, University of Waterloo, 2006.

[47] HOEPER, K., AND GONG, G. Key revocation for identity-based schemes in mobile ad hoc networks. In *Proc. ADHOC-NOW* (2006), LNCS, Springer, pp. 224–237.

[48] HUANG, Y.-M., AND LIN, H.-Y. Information service on scalable ad-hoc mobile wireless networks. In *Proc. Computer Networks and Mobile Computing* (2003), IEEE, pp. 190–196.

[49] HUANG, Y.-M., LIN, H.-Y., AND WANG, T.-I. Inter-cluster routing authentication for ad hoc networks by a hierarchical key scheme. *J. Comput. Sci. Technol. 21*, 6 (2006), 997–1011.

[50] JIANG, X. F. Berkeley Wireless AC Meter/Switch. At `http://smote.cs.berkeley.edu:8000/tracenv/wiki/ACME`, 2011. referenced 20.June.2011.

[51] JOUX, A. A one round protocol for tripartite diffie-hellman. In *Proc. ANTS IV* (2000), LNCS, Springer-Verlag, pp. 385–394.

[52] KHALILI, A., KATZ, J., AND ARBAUGH, W. A. Toward secure key distribution in truly ad-hoc networks. In *Proc. SAINT Workshops* (2003), IEEE, pp. 342–346.

[53] KONG, J., LUO, H., XU, K., GU, D. L., GERLA, M., AND LU, S. Adaptive security for multilevel ad hoc networks. *J. Wireless Communications and Mobile Computing 2*, 5 (2002), 533–547.

[54] LEE, B., BOYD, C., DAWSON, E., KIM, K., YANG, J., AND YOO, S. Secure key issuing in ID-based cryptography. In *Proc. 2nd Australasian Information Security Workshop* (2004), ACS, pp. 69–74.

[55] LEE, W., AND SRIBORRIRUX, W. Optimizing authentication mechanisms using ID-based cryptography in ad hoc wireless mobile networks. In *Proc. Information Networking, Networking Technologies for Broadband and Mobile Networks* (2004), LNCS, Springer, pp. 925–934.

[56] LEE, Y.-H., KIM, H., CHUNG, B., LEE, J., AND YOON, H. On-demand secure routing protocol for ad hoc network using id based cryptosystem. In *Proc. 4th ICPD-CAT* (2003), IEEE, pp. 211–215.

[57] LEINER, B., RUTHER, R., AND SASTRY, A. Goals and challenges of the DARPA GloMo program. In *Proc. Personal Communications* (1996), IEEE, pp. 34–43.

[58] LI, G., AND HAN, W. A new scheme for key management in ad hoc networks. In *Proc. 4th International Conference on Networking Proceedings* (2005), LNCS, Springer, pp. 242–249.

[59] LIBERT, B., AND QUISQUATER, J. A new identity based signcryption schemes from pairings. In *Proc. Information Theory Workshop* (2003), IEEE, pp. 155–158.

[60] LIN, H.-Y., HUANG, Y.-M., AND WANG, T.-I. Resilient cluster-organizing key management and secure routing protocol for mobile ad hoc networks. In *Proc. IEICE Transactions on Communications* (2005), IEICE, pp. 3598–3613.

[61] LIU, D., NING, P., AND SUN, K. Efficient self-healing group key distribution with revocation capability. In *Proc. 10th ACM conference on Computer and communications security* (2003), ACM, pp. 231–240.

[62] LUO, H., ZERFOS, P., KONG, J., LU, S., AND ZHANG, L. Self-securing ad hoc wireless networks. In *ISCC* (2002), IEEE Computer Society, pp. 567–574.

[63] MARTI, S., GIULI, T. J., LAI, K., AND BAKER, M. Mitigating routing misbehavior in mobile ad hoc networks. In *Proc. 6th annual international conference on Mobile computing and networking* (2000), MobiCom '00, ACM, pp. 255–265.

[64] MAURER, AND YACOBI. Non-interactive public-key cryptography. In *Proc. 10th annual international conference on Theory and application of cryptographic techniques* (1991), EUROCRYPT'91, Springer, pp. 498–507.

[65] MERWE, J. V. D., DAWOUD, D., AND MCDONALD, S. A survey on peer-to-peer key management for mobile ad hoc networks. *ACM Comput. Surv. 39*, 1 (2007), 1–45.

[66] MICHIARDI, P., AND MOLVA, R. Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proc. the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security* (2002), pp. 107–121.

[67] OH, J., LEE, K., AND MOON, S.-J. How to solve key escrow and identity revocation in identity-based encryption schemes. In *Proc. ICISS 2005* (2005), LNCS, Springer, pp. 290–303.

[68] PANAOUSIS, E. A., NAZARYAN, L., AND POLITIS, C. Securing AODV against wormhole attacks in emergency manet multimedia communications. In *Proc. 5th International ICST Mobile Multimedia Communications Conference* (2009), pp. 1–7.

[69] PARK, B.-N., AND LEE, W. ISMANET: A secure routing protocol using identity-based signcryption scheme for mobile ad-hoc networks. *J. IEICE Transactions on Communications E88-B*, 6 (2005), 2548–2556.

[70] PARK, B.-N., MYUNG, J., AND LEE, W. ISSRP: A secure routing protocol using identity-based signcryption scheme in ad-hoc networks. In *Proc. 5th International Conference on Parallel and Distributed Computing* (2004), LNCS, Springer, pp. 711–714.

[71] PARK, B.-N., MYUNG, J., AND LEE, W. LSRP: A lightweight secure routing protocol with low cost for ad-hoc networks. In *Proc. International Conference on Convergence in Broadband and Mobile Networking* (2005), LNCS, Springer, pp. 160–169.

[72] PATERSON, K. G. ID-based signatures from pairings on elliptic curves. Report 2002/004, Cryptology ePrint Archive, 2002. http://eprint.iacr.org/2002/004.ps.gz.

[73] PFITZMANN, A., AND HANSEN, M. Anonymity, unlinkability, unobservability, pseudonymity, and identity management— a consolidated proposal for terminology, 2005. "http://www.freehaven.net/anonbib/cache/terminology.pdf".

[74] RAFFO, D. Security Schemes for the OLSR Protocol for Ad Hoc Networks. PhD thesis in the University of Paris 6, 2005.

[75] REN, Y., WANG, J., ZHANG, Y., AND FANG, L. Identity-based key issuing protocol for ad hoc networks. In *Proc. International Conference on Computational Intelligence and Security* (2007), IEEE, pp. 917–921.

[76] RHEE, K. H., PARK, Y.-H., AND TSUDIK, G. A group key management architecture for mobile ad-hoc wireless networks. *J. Inf. Sci. Eng 21*, 2 (2005), 415–428.

[77] RUPPE, R., AND GRISWALD, S. Near Term Digital Radio (NTDR) System. In *Proc. MILCOM'97* (1997), IEEE, pp. 1282–1287.

[78] SAKAI, R., OHGISHI, K., AND KASAHARA, M. Cryptosystems based on pairing. In *Proc. Symposium on Cryptography and Information Security* (2000), SCIS, pp. 26–28.

[79] SAXENA, N. Public key cryptography sans certificates in ad hoc networks. In *Proc. 4th International Conference on Applied Cryptography and Network Security* (2006), LNCS, Springer, pp. 375–389.

[80] SHAMIR, A. How to share a secret. *Communications of the ACM 22*, 11 (1979), 612–613.

[81] SHAMIR, A. Identity-based cryptosystems and signature schemes. In *Proc. Crypto* (1984), LNCS, Springer, pp. 47–53.

[82] SONG, J., KIM, H., LEE, S., AND YOON, H. Security enhancement in ad hoc network with id-based cryptosystem. In *Proc. the 7th International Conference on Advanced Communication Technology* (2005), IEEE, pp. 372–376.

[83] TANAKA, H. A realization scheme for the identity-based cryptosystem. In *Proc. CRYPTO '87* (1987), LNCS, Springer, pp. 340–349.

[84] TOH, C.-K. *Ad Hoc Mobile Wireless Networks Protocols and Systems*. Prentice Hall, NJ,USA, 2002.

[85] TSUJII, S., AND ITOH, T. An ID-based cryptosystem based on the discrete logarithm problem. *IEEE J. on Selected Areas in Communications 7*, 4 (1989), 467–473.

[86] WIN, E. D., MISTER, S., PRENNEL, B., AND WIENER, M. On the performance of signature based on elliptic curves. In *Proc. 3rd Intern. Symp. Algorithmic Number Theory* (1998), LNCS, Springer, pp. 252–266.

[87] XIA, P., WU, M., WANG, K., AND CHEN, X. Identity-Based Fully Distributed Certificate Authority in an OLSR MANET. In *Proc. 4th Wireless Communications, Networking and Mobile Computing* (2008), IEEE, pp. 1–4.

[88] XU, S., AND ČAPKUN, S. Distributed and secure bootstrapping of mobile ad hoc networks: Framework and constructions. *ACM Trans. Inf. Syst. Secur. 12*, 1 (2008), 1–37.

[89] YANG, H., LUO, H., YE, F., LU, S., AND ZHANG, L. Security in mobile ad hoc networks: challenges and solutions. *J. IEEE Wireless Communications 11*, 1 (2004), 38–47.

[90] ZHANG, F., SAFAVI-NAINI, R., AND SUSILO, W. An efficient signature scheme from bilinear pairings and its applications. In *Proc. 7th International Workshop on Theory and Practice in Public Key Cryptography* (2004), Springer, pp. 277–290.

[91] ZHANG, L., HU, Y., AND MU, N. An identity-based broadcast encryption protocol for ad hoc networks. In *Proc. International Conference for Young Computer Scientists* (2008), IEEE, pp. 1619–1623.

[92] ZHANG, P., YE, C., LI, X., CHENG, Y., AND MA, X. Constant-round contributory group key agreement for ad hoc networks. In *Proc. IEEE Wireless Communications, Networking and Mobile Computing* (2005), IEEE, pp. 1199–1202.

[93] ZHANG, Y., LIU, J., WANG, Y., HAN, J., WANG, H., AND WANG, K. Identity-based threshold key management for ad hoc networks. In *Proc. Pacific-Asia Workshop on Computational Intelligence and Industrial Application* (2008), IEEE, pp. 797–801.

[94] ZHANG, Y., LIU, W., LOU, W., AND FANG, Y. MASK: anonymous on-demand routing in mobile ad hoc networks. In *Wireless Communications* (2006), IEEE, pp. 2376–2385.

[95] ZHANG, Y., LIU, W., LOU, W., AND FANG, Y. Securing mobile ad hoc networks with certificateless public keys. *J. Trans. Dependable Secur. Comput. 3,* 4 (2006), 386–399.

[96] ZHANG, Y., LIU, W., LOU, W., FANG, Y., AND KWON, Y. AC-PKI: Anonymous and certificateless public-key infrastructure for mobile ad hoc networks. In *Proc. International Conference on Communications* (2005), IEEE, pp. 3515–3519.

[97] ZHAO, S., AND AGGARWAL, A. Against mobile attacks in ad-hoc networks. In *Proc. International Conference on Information Theory and Information Security* (2010), IEEE, pp. 499–502.

[98] ZHAO, S., JASKIEWICZ, D., AND KARVO, J. A deployment tool for public safety ad-hoc networks. In *Proc. Communication System Software and Middleware* (2006), IEEE, pp. 1–6.

[99] ZHOU, L., AND HAAS, Z. J. Securing ad hoc networks. *J. IEEE Network 13,* 6 (1999), 24–30.

# Appendix A: List of Publications (Year 2006-2011)

**Journal Papers**

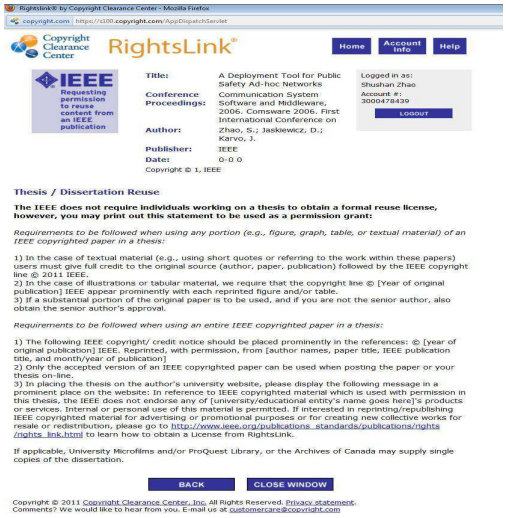
1. Shushan Zhao and Akshai Aggarwal “PAPA-UIC: A Design Approach and a Framework for Secure Mobile Ad-hoc Metworks”, Security and Communication Networks, Special Issue: Security in Ad Hoc Networks and Pervasive Computing, John Wiley and Sons, Volume 3 Issue 5, pp. 371-383, September-October 2010
2. Shushan Zhao, Akshai Aggarwal, Richard Frost and Xiaole Bai “A Survey of Applications of Identity-Based Cryptography in Mobile Ad-Hoc Networks”, Communications Surveys & Tutorials, IEEE, Volume PP Issue 99, pp. 1-21, ISSN: 1553-877X, March 2011
3. Shuping Liu, Shushan Zhao and Weirong Jiang “A Framework for Security-Enhanced Peer-to-Peer Applications in Mobile Cellular Networks”, Int’l J. of Communications Network and System Sciences, Volume 4 Issue 7, pp. 456-463, July 2011
4. Shushan Zhao, Robert D. Kent and Akshai Aggarwal “An Improved and Integrated Key Management and Secure Routing Scheme for Mobile Ad-hoc Networks”, submitted to IEEE Transactions on Communications

**Conference Papers**

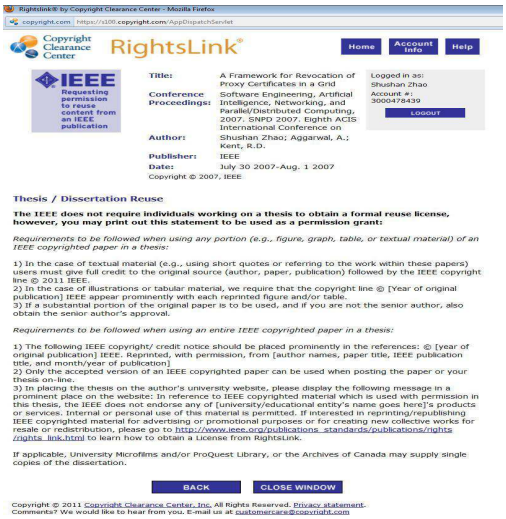
5. Shushan Zhao, Daniel Jaskiewicz and Jouni Karvo “A Deployment Tool for Public Safety Ad-hoc Networks”, In Proc. 1st Communication System Software and Middleware (2006), IEEE
6. Shushan Zhao, Jouni Karvo and Henri Koskinen “Connectivity Enhancement in Deployable Ad-Hoc Networks with Moving Nodes”, 15th IST Mobile & Wireless Communications Summit (2006), IST

7. Shushan Zhao, Akshai Aggarwal and Robert D. Kent “A Framework for Revocation of Proxy Certificates in a Grid”, In Proc. 8th International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (2007), IEEE
8. Shushan Zhao, Akshai Aggarwal and Robert D. Kent “PKI-Based Authentication Mechanisms in Grid Systems”, In Proc. International Conterence on Networking, Architecture, and Storage (2007), IEEE
9. Shushan Zhao, Akshai Aggarwal and Shuping Liu “Building Secure User-to-user Messaging Channels in Mobile Telecommunication Networks”, 7th Wireless Telecommunications Symposium (2007), IEEE
10. Shushan Zhao, Akshai Aggarwal, Shuping Liu and Huapeng Wu “A Secure Routing Protocol in Proactive Security Approach for Mobile Ad-hoc Networks”, In Proc. Wireless Communications & Networking Conference (2008), IEEE
11. Shuping Liu, Jianwei Chen, Shushan Zhao and Fan Na, “Peer-to-Peer Application in Mobile Cellular Systems”, 5th International Conference on Information Technology: New Generations (2008), IEEE
12. Shushan Zhao and Akshai Aggarwal “General-purpose Identity Hiding Schemes for Ad-hoc Networks”, In Proc. International Symposium on Intelligent Ubiquitous Computing (2009), IEEE
13. Shushan Zhao and Akshai Aggarwal “Against mobile attacks in Mobile Ad-hoc Networks”, In Proc. Information Theory and Information Security (2010), IEEE

# Appendix B: Permissions of Reused Publications





copyright.com | https://s100.copyright.com/AppDispatchServlet

Copyright Clearance Center RightsLink® Home | Account Info | Help

IEEE Requesting permission to reuse content from an IEEE publication

| | |
|---|---|
| Title: | A Secure Routing Protocol in Proactive Security Approach for Mobile Ad-Hoc Networks |
| Conference Proceedings: | Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE |
| Author: | Shushan Zhao; Aggarwal, A.; Shuping Liu; Huapeng Wu |
| Publisher: | IEEE |
| Date: | March 31 2008-April 3 2008 |

Logged in as: Shushan Zhao Account #: 3000478439 LOGOUT

Copyright © 2008, IEEE

**Thesis / Dissertaion Reuse**

**The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:**

*Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:*

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

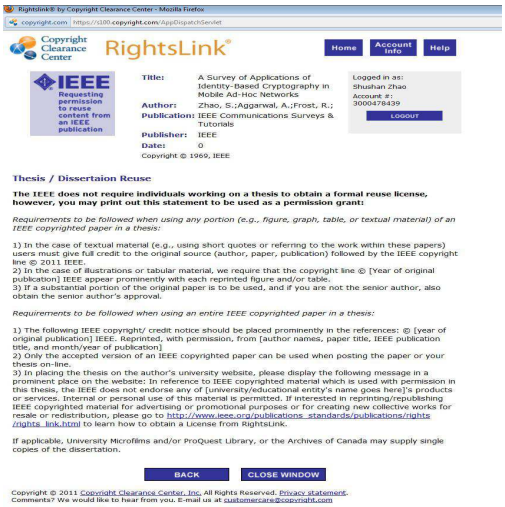
*Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:*

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW





RightsLink® by Copyright Clearance Center - Mozilla Firefox

copyright.com | https://s100.copyright.com/AppDispatchServlet

Copyright Clearance Center RightsLink® Home | Account Info | Help

IEEE Requesting permission to reuse content from an IEEE publication

| | |
|---|---|
| Title: | A Survey of Applications of Identity-Based Cryptography in Mobile Ad-Hoc Networks |
| Author: | Zhao, S.;Aggarwal, A.;Frost, R.; |
| Publication: | IEEE Communications Surveys & Tutorials |
| Publisher: | IEEE |
| Date: | 0 |

Logged in as: Shushan Zhao Account #: 3000478439 LOGOUT

Copyright © 1969, IEEE

**Thesis / Dissertaion Reuse**

**The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:**

*Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:*

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

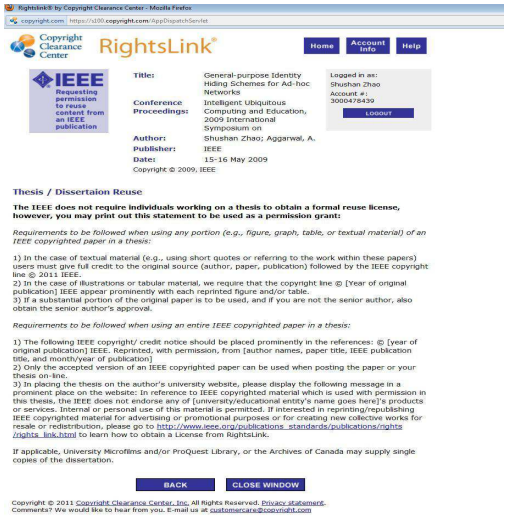
*Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:*

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW





RightsLink® by Copyright Clearance Center - Mozilla Firefox

copyright.com | https://s100.copyright.com/AppDispatchServlet

Copyright Clearance Center RightsLink® Home | Account Info | Help

IEEE Requesting permission to reuse content from an IEEE publication

| | |
|---|---|
| Title: | General-purpose Identity Hiding Schemes for Ad-hoc Networks |
| Conference Proceedings: | Intelligent Ubiquitous Computing and Education, 2009 International Symposium on |
| Author: | Shushan Zhao; Aggarwal, A. |
| Publisher: | IEEE |
| Date: | 15-16 May 2009 |

Logged in as: Shushan Zhao Account #: 3000478439 LOGOUT

Copyright © 2009, IEEE

**Thesis / Dissertaion Reuse**

**The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:**

*Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:*

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

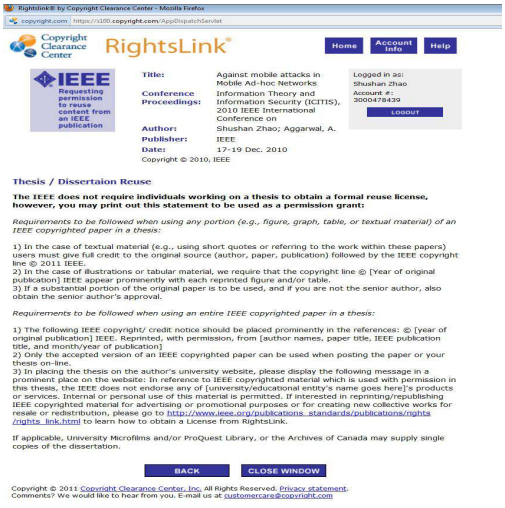
*Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:*

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW





RightsLink® by Copyright Clearance Center - Mozilla Firefox

copyright.com | https://s100.copyright.com/AppDispatchServlet

Copyright Clearance Center RightsLink® Home | Account Info | Help

IEEE Requesting permission to reuse content from an IEEE publication

| | |
|---|---|
| Title: | Against mobile attacks in Mobile Ad-hoc Networks |
| Conference Proceedings: | Information Theory and Information Security (ICITIS), 2010 IEEE International Conference on |
| Author: | Shushan Zhao; Aggarwal, A. |
| Publisher: | IEEE |
| Date: | 17-19 Dec. 2010 |

Logged in as: Shushan Zhao Account #: 3000478439 LOGOUT

Copyright © 2010, IEEE

**Thesis / Dissertaion Reuse**

**The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:**

*Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:*

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

*Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:*

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

# Vita Auctoris

Shushan Zhao was born in Shandong China. He graduated from Shandong University in 1992, Jinan China, where he obtained Bachelor of Science degree in Computer Software. He obtained Master of Science degree in Telecommunication Software from Helsinki University of Technology in Finland in 2005. Since September 2006, he has been pursuing his doctoral research in the school of Computer Science at the University of Windsor, Ontario, Canada.