

國立交通大學

資訊工程學系

碩士論文

適用無線網路下抵擋路由攻擊
之路徑認證機制



A Path Authentication Scheme for Routing Disruption
Attack Prevention in Ad Hoc Network

研究生：林立洲

指導教授：謝續平 博士

中華民國九十三年六月

適用無線網路下抵擋路由攻擊之路徑認證機制

A Path Authentication Scheme for Routing Disruption

Attack Prevention in Ad Hoc Network

研究生：林立洲

Student: Li-Joe Lin

指導教授：謝續平 博士

Advisor: Dr. Shih-Pyng Shieh

國 立 交 通 大 學
資 訊 工 程 學 系
碩 士 論 文

A Thesis
Submitted to
Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
In Partial Fulfillment of the Requirements
For the Degree of
Master
In

Computer Science and Information Engineering

June 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年六月

適用無線網路下抵擋路由攻擊之路徑認證機制

研究生：林立洲

指導教授：謝續平

國立交通大學 資訊工程學系

摘要

由於 ad hoc 網路路由協定於設計時欠缺安全性的考量，任何假造竄改的錯誤路由訊息將擾亂整個路由系統，並使得目前運行於 ad hoc 網路路由協定遭受到嚴重的威脅。而目前所提出針對安全路由機制的研究，大多採行密碼學中非對稱式運算基礎運作，也因為非對稱式架構運算的高成本，使得這些機制不適用於移動式網路環境下。在本篇論文裡，我們將提出一個在路由路徑上的合作式攻擊模型，並用於鑑別目前所用來保護路由資訊方法之不足點，針對其不足點，我們提出一個有效的路徑認證機制，在特定的路由需求路徑上，確保路由資訊的完整性，並有助於目前的路由協定對抗路由攻擊。

A Path Authentication Scheme for Routing Disruption Attack Prevention in Ad Hoc Network

Student: Li-Joe Lin

Advisor: Shih-Pyng Shieh

Abstract

Ad hoc routing protocols are vulnerable due to the absence of security mechanism. Forged routing advertisement can disrupt the routing scheme. Research work has been proposed for securing the routing protocol in ad hoc networks. Some of them deployed the asymmetric cryptographic primitive, which are often infeasible in the mobile environment. In this paper, we discovered a strict, cooperative disruption attack behavior on the route path and identify the deficiency about present secure mechanisms for protecting the routing information. We proposed a path authentication scheme which relies on efficient symmetric cryptographic authentication approach. The Random Assignment Path Authentication (RAPA scheme) guarantees the integrity of a complete request route path in route discovery procedure and help the current on-demand routing protocol for resisting against the routing disruption attacks. Our scheme can be adjusted to meet different efficiency and security requirements for the various applications.

誌 謝

學生能得以完成本篇論文，最首要感謝的是我的指導老師謝續平教授。感謝老師在論文上的指導、幫助以及所給予的支持和鼓勵。再者，想感謝我的爸媽辛苦的栽培，而他們的支持，一直是令我有著克服任何困難的最大動力。也要感謝實驗室的學長姐、學弟們和佳燕助理，謝謝他們在生活課業上的幫助、在作論文期間的探討研究及協助。感謝好友陳志偉、學弟邱建熹在計中助教的工作期間熱心的協助及指教，辛苦你們偶而還必需聆聽已經重覆的問題。感謝好友小銘總是聆聽我在這兩年研究生活上的苦水，雖然他的建議我總是沒採納，但他總是給我從另一角度去思考的方向。感謝雲太在生活上給予的關心及隨時充當一個開心的種子，以及幾乎沒有拒絕的幫助。



Table of Content

Chapter 1. Introduction	1
1.1. Security Requirements	2
1.2. Related work	3
1.2.1. Setup Pair-wised Shared Keys and Public-key distribution.....	4
1.2.2. Broadcast Authentication Mechanisms.....	4
1.2.3. Secure On Demand Routing Protocols on Ad Hoc networks	5
1.2.4. Disruption Prevention in Ad Hoc Routing protocol	6
1.3. Contribution	7
1.4. Synopsis	8
Chapter 2. Preliminaries.....	9
2.1. Basics of Dynamic Source On-Demand Routing Protocol.....	9
2.2. Overview of Rushing Attack Prevention(RAP).....	10
2.3. Cooperative Attack Routing Disruption Threats.....	11
2.4. An extremely efficient authentication mechanism: HORS.....	13
2.5. Summary	15
Chapter 3. Proposed Scheme	16
3.1. Security Assumptions	16
3.2. Random Assignment Path Authentication(RAPA)	16
3.2.1. Notations	19
3.2.2. RAPA Protocol Description	20
3.2.3. RAPA Forwarder's Procedure.....	22
3.3. A reactive security adjusting strategy	25
3.4. Malicious nodes isolation feature	26
Chapter 4. Security Analysis:.....	28
4.1. Security strength of standard signature for HORS	28
4.2. Random Number Set Collision in Path Authentication	28
4.3. Comparison with Overall Cost for Forwarding Route Request.....	33
Chapter 5. Conclusions	37
Chapter 6. References	38

List of Figures

Figure 2-1 An example for Dynamic Source Routing protocol (a) node i broadcast the Route Request to discovery the path to target node T . (b) intermediate nodes help rebroadcast the Route Request message and append their own address. (c) target node T reversed the path list and send back to the initiator i	10
Figure 2-2 Secure Neighbor Detection for rushing attack prevention.....	11
Figure 2-3 Cooperative attackers to reveal the disruption attack threats.....	13
Figure 2-4 Key-pair generation for authentication in HORS.....	14
Figure 3-1 Basic processing flow chart for forwarder.....	17
Figure 3-2 Basic processing flow chart for initiator.....	19
Figure 3-3 A general RAPA protocol example.....	22
Figure 3-4 RAPA Forwarder’s Signing and Verification Algorithm.....	23
Figure 3-5 Protocol description the cooperative attack scenario.....	24
Figure 3-6 A reactive adjusting strategy for Route Discovery procedure.....	26
Figure 4-1 A algorithm for estimating rekeying interval with a given threshold probability.....	30
Figure 4-2 Variation of $\log_{10}(\Psi(n))$ in different parameters setting.....	32
Figure 4-3 Variation of $\log_{10}(\Psi(n))$ and the expected value for the same number of public slices.....	33
Figure 4-4 Variation of $\log_{10}(\Psi(n))$ and the expected value for the same number of private slices.....	33
Figure 4-5 Cost difference between RAPA and the straightforward way.....	35
Figure 4-6 Overall Cost comparison.....	36

Chapter 1. Introduction

Ad hoc wireless networks are essentially infrastructureless and adaptive. There are no any other additional fixed devices, such as base stations or a specific router, to be deployed in advance. Any couple of devices on the network can directly communicate if they are both within the radio transmission range. In general, ad hoc network is composed of several communication hosts sharing the same goal and all participants will play the intermediate nodes which help forward packets for others. Therefore, the important component of the ad hoc network is definitely the routing protocol. The design of routing protocol must meet several challenging factors, such like high mobility for dynamically changing topology, small and low computational power devices etc. As a whole, the routing protocol of ad hoc networks should take efficiency and low cost features. The proactive routing protocol which needs periodic updating packets caused either operation or bandwidth overhead constantly. In contrary, the reactive on-demand routing protocol is more feasible for wireless environment because the route discovery process is initiated only when the data transmission is on demand for communication. The secure routing protocol is even more challenging to design. For the reason that each host could be the role of forwarding the packets, the routing of ad hoc network will turn into a disaster without security consideration. Attackers can forge routing information to create an infinite routing loop. The malicious nodes disseminates artificial routing packets could also bring a black hole to attract most packets and drop all of them. Gray hole is a special case of black hole attack, in which the packets are selectively dropped. The above attacks can be detected by some probe mechanisms [4].

In this paper, the main issues we concern are the routing disruption attacks

caused by the intermediate nodes on the route path such as fabricating routing information to flood numerous routing control messages to dominate the forwarding resources [24], or modifying routing message to poison route cache and turn the routing protocol into a chaos.

1.1. Security Requirements

Due to the absence of security consideration, uncooperative intermediate nodes of ad hoc network would exhibit some Byzantine behavior to disrupt the whole routing system. In this paper, we consider the following requirements in a secure ad hoc routing protocol to prevent the routing disruption behavior.

Robustness:

The routing disruption attack could be a misbehavior which is caused by a selfish node tries to dominate most of forwarding resource. And the secure routing protocol should be robust to defend the impact from the misbehavior which is revealed from intermediate nodes either alone or in collusion with other nodes.

Lightweight authentication mechanism for route request messages:

Authentication could be used to avoid the propagation of the forged routing information from attackers. The authentication methods based on PKI infrastructure are not fit in ad hoc mobile network, since there is no always a trusted centralized CA available, and the RSA based or exponential computation based cryptographic operation is not preferred in the mobile environment. The mobile devices are often resource-constrained in computational power, battery capacity and so on. Therefore, the efficiency is a quite important consideration in choosing authentication mechanisms. Otherwise, the attacker could keep the forwarder busy verifying the bogus authentications and exhaust their resource if the inefficient authentication mechanism is deployed.

Source authentication:

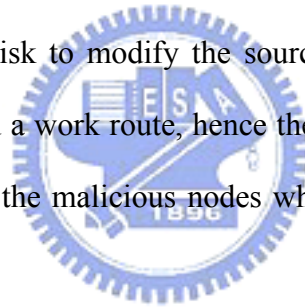
If the intermediate node on route path could not verify the identity of initiator who originated route request message, then the malicious node can impersonate and fabricate the route request to cause the rushing attack.

Path authentication capability:

An uncooperative node could inject counterfeit routing record into a route request message and prevent the legal nodes from finding a work route if the routing protocol has no capacity for checking the integrity of a source route record. It means that the polluted routing information could disrupt the whole routing system by poisoning the route cache.

Isolation and Announcement:

The attacker tries to take a risk to modify the source route record, and caused the legitimate nodes failed to find a work route, hence the secure ad hoc routing protocol should have ability to isolate the malicious nodes when they could be caught due to their misbehavior.

**Adjustable secure level for efficiency trade-off:**

The system which provided higher security often has more computational overhead or lower performance. Therefore, a reactive strategy to rectify the computational cost of the overall system is a plus in a secure routing protocol for more efficiency and flexibility.

1.2. Related work

There are several secure components which are deployed to provide the routing security for ad hoc networks. First, we will introduce the shared key setup mechanisms. It is key component for many security services. Further, we review and compare several efficient broadcast authentication mechanisms which are more practical than the asymmetric cryptographic methods in the resource-constrained

environment. Then we examine some current secure routing protocols on the ad hoc networks. We also present the disruption attacks and some countermeasures for the ad hoc routing protocol.

1.2.1. Setup Pair-wised Shared Keys and Public-key distribution

Many authentication mechanisms assume the pairwise shared secrets or the prior-dispensed authentic public values for verification. They must be distributed in advance in order to make sure the authentication mechanisms workable. The concept of the key distribution center [3] can be used to bootstrap the shared key, its role also could be the network access control system. Zhu et al [20] proposed a distributed protocol for establishing a shared key without a centralized infrastructure, which combine probabilistic key sharing and threshold secret sharing techniques. The most well-known solution of public-key distribution problem bases on third party public-key certificates. Another approach presented by Hubaux et al [9][18] provides a self-organized infrastructure like Pretty Good Privacy (PGP) system but without depending on the certificate directories for public-key distribution. Every user maintains a small local certificate repository, and merges others to find the certificate chains to each other with high probability. Numerous of distribution techniques have been proposed [9][12].

1.2.2. Broadcast Authentication Mechanisms

The broadcast authentication protocol is used to confirm the source identification of sender nodes and the one-time authentication mechanism is used for broadcast authentication recently due to their efficiency. However, another approach for broadcast authentication, named signature amortization, which relies on RSA-like cryptographic primitives is still cost-expensive for computing the signature and not suitable on resource-constrained mobile network environment. One-time authentication applies the symmetric cryptographic operation but brilliantly achieves

asymmetric signature property like the public-key system. The primitive idea most put the onewayness of the hash function in applying the signature verification. Timed Efficient Stream Loss-tolerant Authentication (TESLA) protocol [1] which is presented by Perrig et al is a light-way source authentication approach. TESLA integrates a time element and the concept of the hash-chain into the authentication process. It achieves the asymmetric property by delaying signature-verified time and possesses the loss-tolerant characteristic. The necessary for time synchronization and temporally buffering received messages are primary constrains for TESLA. BiBa [2] also proposed Perrig, it takes use of the hash collision event and discloses the partial private values as the message signatures. The security strength against which the adversary has to take is getting easier with more disclosed partial private values. In contrary, the HORS [13] requires the subset-resilient hash function to generate different combination values for signatures. HORS also is so far most efficient broadcast authentication mechanism and our scheme also integrates it for source authentication. We will depict it in details on next chapter.

1.2.3. Secure On Demand Routing Protocols on Ad Hoc networks

There are several secure routing protocols which have been developed on ad hoc networks. In this paper, we focus on securing on-demand routing protocol and there are several works [10][17] show its better performance than that proactive routing protocols could provide. Generally speaking, the ad hoc routing protocol needs no confidentiality and privacy. Actually the routing privacy is also difficult to accomplish in wired network. The most significant security consideration for secure ad hoc routing protocol is the ability to prevent the routing information from impersonation and modification. Dahill et al [11] proposed a secure routing protocol, named ARAN which required a centralized trusted certificate server which initiated all certifications for communication nodes and also deployed the asymmetric cryptography for digital

signature. However computing such signature is expensive on resource-constrained mobile devices. Papadimitratos and Haas proposed the Secure Routing Protocol (SRP) [16] to apply on the existing routing protocols, particularly for Dynamic Source Routing protocol [7][6]. It assumed that any pair of initiator and target node in route discovery shared the security association for authentication or other secure communication scheme. Basically, SRP point the concept about end-to-end security and ignore the possible disruption behaviors revealed by the intermediate nodes on the route path.

Ariadne is a secure on demand routing protocol proposed by Perrig et al, which uses MAC with a shared secret key between any couple of communication nodes for end-to-end authentication. The efficient authentication protocol TESLA is also deployed for providing integrality of a work route between the intermediate nodes. The requirement of the time synchronization is the main drawback for Ariadne. Secure Ad Hoc On-Demand Vector Routing protocol (SAODV) [15] provides a security framework with IPsec for AODV protocol. Zapata and Asokan [14] also identify the security flaws of AODV, then deploying the hash chain to protect the hop count field of AODV from modifying by malicious nodes.

1.2.4. Disruption Prevention in Ad Hoc Routing protocol

Although the black hole and gray hole [23] problem is difficult to defend by current secure routing protocol, the additional detection techniques [21] have been proposed and possibly mitigate this kind of disruption behaviors. This work devises two methods: *watchdog* is taken for detecting the uncooperative nodes, *pathrater* is taken for managing the most reliable path, whereas the main problem of this approach is still hard to detect some ambiguous misbehavior such as receiver collisions, collusion and etc. The disruption misbehavior about flooding the routing control

messages and fabricating the routing information are main concern issues in this paper. The corrupted routing information might cause Route Cache Poison attack in DSR. Since DSR allows nodes to learn new routes from any promiscuously received packets and also implicitly gives the chance to exploit this flaw for the malicious nodes. Consequently any behavior about the injection or alteration with false routing information should be exhibited on secure ad hoc routing protocol. The rushing attack is a malicious attacks introduced by Hu and Perrig [24], and which exists in on-demand routing protocol of the ad hoc network, such as AODV[5], SAODV[15], DSR[7], Ariadne[23], ARAN[11] and etc. In the Router Discovery phase, the normal nodes initiate a Route Request message to request a route path for communicating with the target. Except for target node, all the nodes received the Route Request will help forwarding the Route Request. However in present on-demand routing protocol, the intermediate nodes only forward the first arrived Route Request then discard any further Route Request arrived later. This serious flaw could be exploited by attackers. Under this attacks scenario, attacker can block the legitimate Route Request by quickly resending Route Request to result an effective denial-of-service phenomenon. The Rushing Attack Prevention (RAP) solution [24] is adequate to solve Acitve-1-1 attacker model presented in Ariadne for rushing behavior. However, the cooperative attackers could turn the routing architecture into a mess, which is detailed in Chapter 2.

1.3. Contribution

We presented a cooperative attack behavior and identify the inadequacy of current secure on-demand routing protocol. We also proposed a efficient path authentication scheme which focuses on securing on-demand routing protocol and preventing the routing disruption behavior under the cooperative attack model, where the consecutive attackers can collaborate to disrupt the routing system suchlike

fabricating, or altering the contents of the routing control packets. Our scheme provided the source more flexibility for both efficiency and security consideration in tuning the authentication cost and security level. We also suggest a plain strategy to demonstrate this property the scheme could provide. Our path authentication mechanism requires lower overall cost for signature generation and verification than a straightforward method needs. Finally, the strategy we provided is possible to locate and isolate the malicious nodes.

1.4. Synopsis

The architecture of this paper will be presented as the follows: Chapter 2 introduces the preliminaries needed in our scheme. The proposed path authentication mechanism is detailed in Chapter 3. The analytical security and cost analysis is performed in Chapter 4. Finally we make conclusions about our work in last chapter.



Chapter 2. Preliminaries

In this Chapter, we will introduce the background before presenting our path authentication scheme. First we take an overview of basic operation in the Dynamic Source Routing protocol. And further, we illustrate the RAP component more clearly than as we mentioned in related works, and demonstrate the threats which it has to face under our proposed attack model. Thereafter, we describe the HORS which is the efficient broadcast authentication mechanism deployed in our path authentication scheme.

2.1. Basics of Dynamic Source On-Demand Routing Protocol

DSR[7] is a reactive source routing protocol which has lower overhead than proactive routing protocol. On-demand behavior denotes that the route path is discovered when the source node wants to send packets to the destination node. The on-demand behavior causes the less overhead packet when all needed route have been constructed. DSR is mainly composed of two functions: route discovery and route maintenance.

In DSR, the route discovery will be triggered when the node tries to send some data packets to the destination node. We use Figure 2-1 to illustrate the route discovery procedure. The node **i** try to communicate with the node **T**, and then broadcast the Route Request with specified target address and the unique identifier. When the other node received this Route Request, it will discard the request packet if it has seen the identifier before. Otherwise, it appends its own address to the hops list and rebroadcast the Route Request. This procedure will repeat until the Route Request has reached the target node **T**. When target node **T** sees the Route Request packet, node **T** will reverse the routing path list to send back the Route Reply if the

bi-directional link exists or initiate a new route discovery back to initiator node **i** with piggybacked routing path. The thick line means the reversed route path $\{T, F2, F1, i\}$ chosen by the node **T**. Obviously, there are many routes from the node **i** to node **T**. The intermediate nodes also cache the route path for future use.

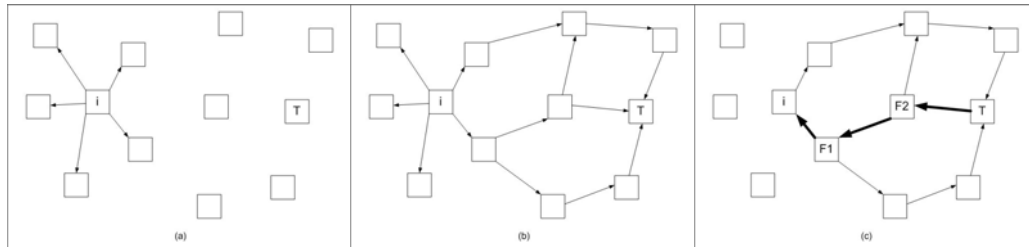


Figure 2-1 An example for Dynamic Source Routing protocol (a) node **i** broadcast the Route Request to discover the path to target node **T**. (b) intermediate nodes help rebroadcast the Route Request message and append their own address. (c) target node **T** reversed the path list and send back to the initiator **i**

The route maintenance is used for detecting broken links. When the intermediate nodes find the occurrence of the broken link it will send Route Error back to the source for notifying the non-existence about the traveling route path. In this paper, we focus on securing the Route Discovery phase.

2.2. Overview of Rushing Attack Prevention(RAP)

Rushing Attack Prevention (RAP) is a defense component, which is purposed by Hu et al, to aim at the weakness of duplicate-suppression routing request in the on-demand protocol of ad hoc network. This prevention mechanism is mainly composed of two components, which are Secure Neighbor Detection and Randomized Message Forwarding; Secure Neighbor Detection is a component which used to defend a special case of wormhole attack [22] and Active-1-1 attack model which have been defined in Ariadne. It mainly adopted the mutual authentication between both neighbor forwarders in the request route path. As the illustration with the Figure 2-2 below, the redundant messages 1, 2 are used to protect the routing system from

the repeater attack which is a special case of wormhole attack. In this attack scenario, the attacker can introduce the two nodes which actually are not in each others' communication range as neighbor nodes. It can be applied to generate the rushing behavior. Message 1, 2 can be taken to estimate a maximum bound of the distance between two communication nodes by calculating the delay when the message 2 returns. Message 3 is the route request message of traditional forwarding process and RAP component ask the F1 to piggyback his previous signature in this message. Therefore, any one compromised node of the path could not modify the route request messages and the one-hop-far rushing behavior can be avoided under this scene. Our proposed idea mainly concentrates on enhancing this route request forwarding process. Another component of RAP, Randomized Message Forwarding is to choose one request to forward from its collected N route request for the purpose of mitigating the chance that the adversary can dominate the forwarder's resource by rapidly sending request messages. RAP achieves what it claimed indeed but seems to be vulnerable in our attack model. In next section, we will present this attack model cooperated by several malicious nodes.

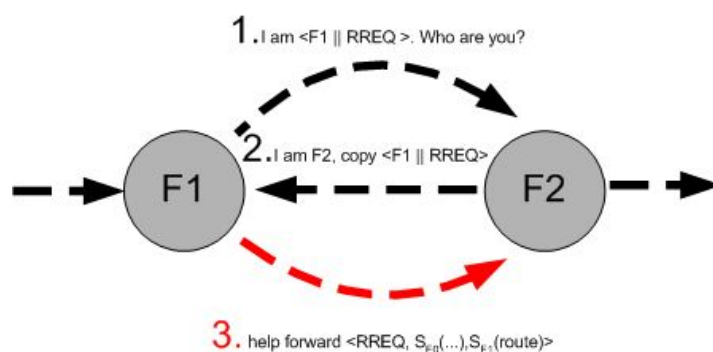


Figure 2-2 Secure Neighbor Detection for rushing attack prevention

2.3. Cooperative Attack Routing Disruption Threats

We are going to present a specific attack model to which the RAP is still

vulnerable. This type attack is a special subset cases of Active-x-y attacker model mentioned in Ariadne. We need to more precisely separate this one from Active-n-n attacker model because it is helpful to evaluate the robustness of the routing protocol and to discuss the problem the RAP might miss. We denote the Cooperative-n Attack model as that the attackers were positioned consecutively in the same forwarding route path and they can cooperatively disrupt the routing protocol, like the rushing attack or fabrication to the routing information. Figure 2-3 shows the Cooperative-2 attack model and we use this model to identify the security concern which is absent in RAP. In this attack scenario, node M1 and node M2 are malicious nodes who try to mislead the Route Discovery phase. Due to the lack of the ability to verify the initiator's identity for node F in the same route request path, the cooperative attackers can take advantage of this weakness to fabricate a faked route request to rush the successive forwarder F. Secure Neighbor Detection provided the forwarding node F the hopping authentication ability by concatenating the piggyback signature which was two hops far from the forwarding node F itself. However, in RAP, node M1 and node M2 can cooperatively cheat the node F with forwarding a lot of forged route requests from nodes f1~f3. Node F is still under the rushing attack threat. Moreover, cooperative attackers can inject some faked hops into the route request message and this could poison the node F's or other nodes' route cache to mislead the whole routing system.

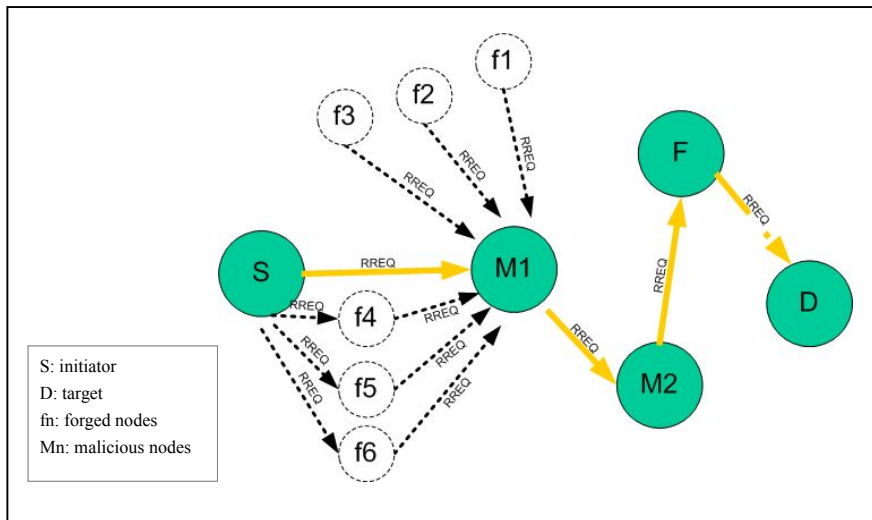


Figure 2-3 Cooperative attackers to reveal the disruption attack threats

2.4. An extremely efficient authentication mechanism: HORS

As the above section 1.2.3 mentioned, we need an instantly-verifiable authentication mechanism for path authentication. We apply a so far most efficient broadcast authentication mechanism to our scheme. We first review the detail of the HORS then we will make a little change in our proposed scheme.

HORS keep the fast signing and verifying speed which benefited from the hash operation of symmetric authentication mechanisms, such like RIPEMD-160 and SHA-1 etc. It also possesses the wonderful feature of asymmetric authentication as the public-key system. We are going to explain the three phases, key-pair generation, signing and verification in HORS. Before generating the signature for a particular message, sender must have its own private-public key pair; the key-pair generation is described in Figure 2-4.

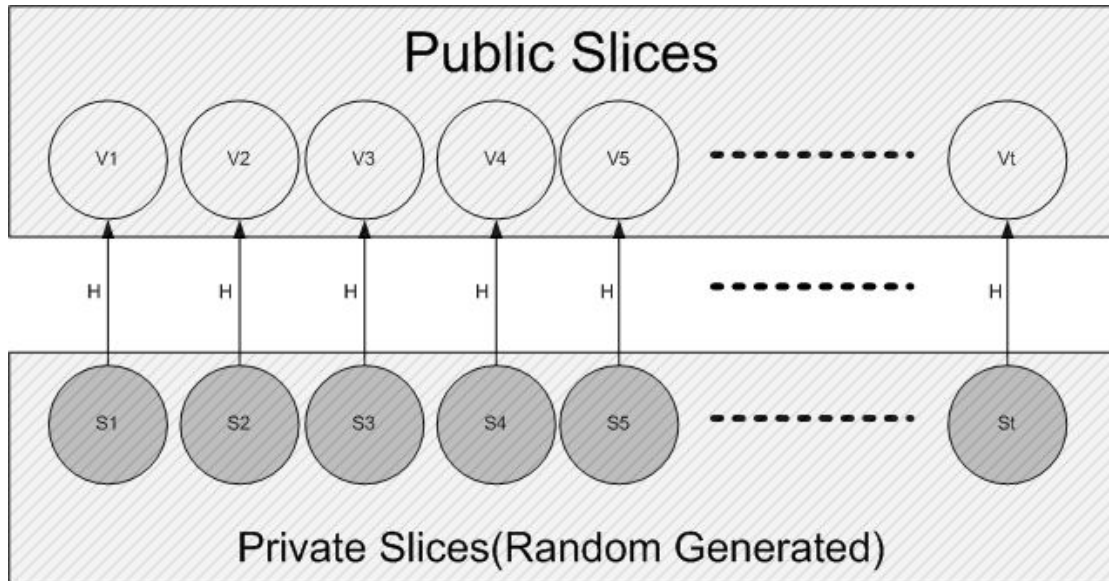


Figure 2-4 Key-pair generation for authentication in HORS

Sender first generate a total t random number as its own secret keys: $SK=(S_1, S_2, \dots, S_t)$, we called SK as private slices. In the Figure 2-4, “H” means the public hash operation which is used to verify the signature as follows: $PK = (V_1, V_2, \dots, V_t)$, $V_1=H(S_1), \dots, V_t=H(S_t)$. PK is named public slices. Considering a simple phenomenon about the sender-receiver message authentication procedure and PK is assumed available for every communication partners with distribution in advance; if a sender wants to transmit a message M to the receiver and attach its signature to message M , the sender will do the follows:

- (1) Split the hash value of M into k substrings of length $\log(t)$, $k < t$
- (2) Interpret k substrings into k integers as index values
- (3) Select a k -elements subset from t private slices as signature of M according to k index values of (2)

When the receiver obtained the k private slices, the receiver can verify the signature by checking if the hash values of k private slices equal the correspondingly public slices. Through the security consideration, the sender should also determine when to republish its new public slices which named rekeying procedure. In general, HORS

apply a fabulous feature that is releasing the partial private slices to achieve approximately the asymmetric authentication characteristic in public key system. Besides, HORS possesses extremely efficient signing and verifying capacity, requiring only a couple of simple cryptographic hash operations.

2.5. Summary

Even the secure version of DSR [23] which provided the properties of end-to-end authentication for using a shared MAC key was still vulnerable to routing disruption attacks. Ariadne only supplied the initiator with the ability to check the integrity of route path until the route reply message received. However, the intermediate nodes in Ariadne could not instantly authenticate the previous hop in route discovery because TESLA authentication mechanism requires clock synchronization, Zapata [14] even point that it might be an unrealistic requirement for ad hoc network. Without instantly-authenticating ability for each entry of the path, any routing protocol could be vulnerable under modification, impersonation and fabrication attack to the routing information. Instead of TESLA, the instantly-verifiable authentication protocol HORS which is fastest in current one-time authentication schemes can be applied well between hops in the route path for path authentication. HORS also keep efficient processing speed for the signing and verifying procedure with the simple hash operation. RAP brought a piggyback signature concept to defend one compromised node against the routing disruption attacks in the route path. However, we will present a more strict attack model which could reveal the rushing attack or spurious routing messages again.

Chapter 3. Proposed Scheme

We proposed a path authentication mechanism to enhance RAP and develop a cost-flexible feature with HORS. We named this scheme as the Random Assignment Path Authentication (RAPA). The RAPA uses the random numbers set issued from initiator as the indexes of generating the authenticator for path authentication. Besides, we will demonstrate the way to isolate a malicious node who tries to modify the routing information in our proposed strategy. The common security assumptions are depicted first in next section.

3.1. Security Assumptions

The underlying data link layer provides reliable transmission on a link and which is assumed to be bidirectional. Each transmission is received by all neighbors, which are assumed to operate in promiscuous mode. The physical transmission media in ad hoc network has a fundamental Denial-Of-Service vulnerability such as jamming attack. The jamming effect could be mitigated by some coding spread spectrum mechanisms. This attack issue was excluded in our work. We also ignored the attacks in Medium Access Control layer. In 802.11, attackers could block his neighbors by flooding CTS signal.

We assume that the necessary shared keys for broadcast authentication have been configured in advance as the techniques mentioned in related work. Our goal aims to defense the disruption misbehavior of intermediate nodes on route path. In our work, the initiator and target node are assumed to be trusted without losing generality.

3.2. Random Assignment Path Authentication(RAPA)

Our design goal is extending RAP as a general securing routing component

named Random Assignment Path Authentication (RAPA) to defend the cooperative-n attack model by applying the efficient authentication mechanism HORS. For conveniently presenting our idea, RAPA will focus on the behavior of delivering the Route Request and ignore the first and second messages about the wormhole detection of SND in RAP. To defend the cooperative attackers as the above we presented, we need a secure-designed approach to take in charge of the forwarding Route Request procedure. This approach should contain two features; except target node, the forwarder should be able to authenticate the identity of initiator in the Route Request. And source authenticity is easy-accomplished due to the asymmetric nature of HORS. Each forwarder must confirm the correctness of source route field in Route Request. The forwarder should verify that each entity in the source route field indeed deliver this Route Request. The forwarder should append its own signature signing the source route field to the route request message before broadcasting it. The below Figure 3-1 illustrates the basic process flow for the forwarder in RAPA.

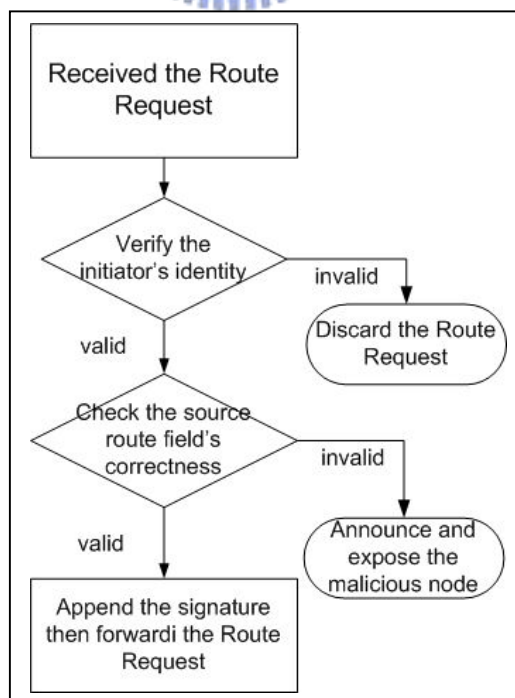


Figure 3-1 Basic processing flow chart for forwarder

With our observation, the forwarder just needs to attach the evidence instead of signature to prove that the request message has passed through itself. Therefore, we modify a little the usage of HORS to help us save the signing and verifying cost in our proposed scheme. The proposed scheme named Random Assignment Path Authentication which provides the initiator the right to adjust the security level of request route path. The higher security level can reduce the chance of that the attacker can inject any bogus routing information into the route request message, and it also can interfere the performance of the entire system. For the reason, we suggest a strategy to adjust the cost corresponding to the stability of constructing a route for path authentication. Our strategy will raise the security level when the stability of route path construction is doubtful and sacrifice the routing performance. The Figure 3-2 below shows the processing flow diagram about the initiator in RAPA. The security level determined by the initiator also provides some flexibility. RAPA also provides the ability to announce the attacker's misbehavior and non-repudiation feature to resist the blackmail attack, in which the malicious users can incriminate deliberately a well-behaved guy. We are going to detail RAPA protocol in the next section.

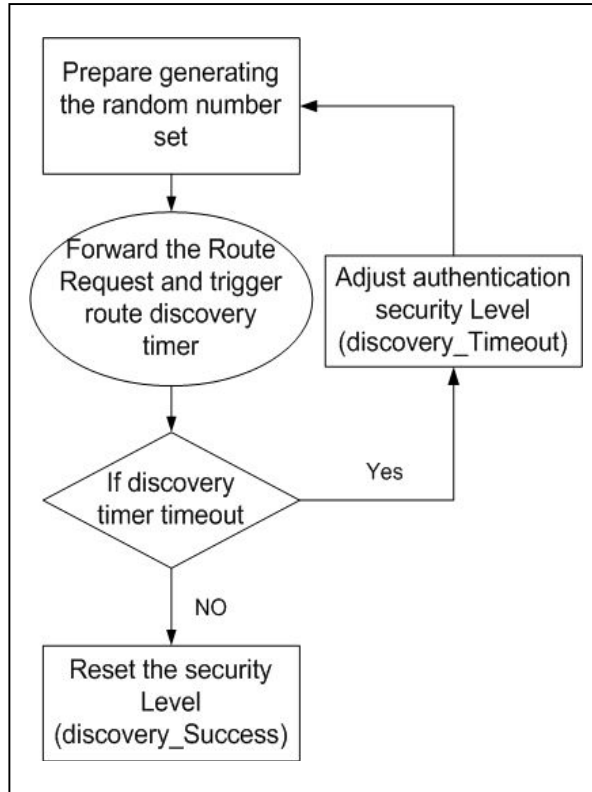


Figure 3-2 Basic processing flow chart for initiator


3.2.1. Notations

We use the following notation as similar in RAP to describe security protocols and some operations:

- S, D : Source(initiator) node and Destination(target) node
- $A \rightarrow B [M]$: node A sends node B the message M
- $A \rightarrow * [M]$: node A broadcast message M
- $A [H(M)]$: A generate the hash value of message M
- $\langle \rangle$: A empty list of data messages or a empty data field
- $SK_A(\mathcal{R})$: A's private slices indexes by a random set \mathcal{R}
- Σ_i : node i's signature
- σ_i : node i's path signature assigned by initiator
- \xleftarrow{R} : Random Generating Process
- \xleftarrow{S} : Signing Process

3.2.2. RAPA Protocol Description

We now describe in detail the RAPA. We assume that every communicating nodes have already know all communicating nodes' public keys of HORS which are distributed in advance; there are a lot of strategy for this purpose have been proposed [9]. RARP integrate HORS authentication mechanism into the path authentication procedure as a secure routing component. RAPA provides the following properties: (1) forwarder can authenticate each entry of the path in the Route Request; (2) every participant in the forwarding route path can authenticate the initiator and thus no cooperative-n attackers can impersonate a fabricated Route Request. (3) no intermediate forwarders can inject or modify the source route in the Route Request under max security level have been set.



A Route Request in RAPA contains primary eight fields: *<Route Request, initiator, target, assigned random list, initiator's signature, source route, path authenticator list, forwarder's signature>*. The *initiator* and *target* are placed the address of initiator node and target node respectively. The *assigned random list* is a set of random number generated by initiator. RAPA requires that each communicating node i possesses two different secret private keys $\{SK_i^s, SK_i^p\}$, SK_i^s for standard signature procedure and another SK_i^p for path authentication. Before disseminating a Route Request to the target node, the initiator has to first randomly assign k random number $\{\pi_1, \pi_2, \dots, \pi_k\}$. These random numbers are used to index π th private slice of SK_i^p which is append to the *path authenticator list* by the forwarder for the purpose of path authentication. The *initiator's signature* is partial slice of the SK_i^s , which signed the *initiator, target* and *assigned random list* fields.

The *source route* field contains the all previous participant nodes' address in the path. Finally, the *forwarder's signature* is applied to sign the whole route request message and replaced by next forwarder's signature. As in DSR, the target node received the Route Request will return the Route Reply message after it confirmed correctness of the signatures and authenticators list. A Route Reply message format in RAPA can be a condensed route request message but removing the *assigned random list*, *path authenticator list* and *forwarder's authenticator*. The Route Reply message also contains the reversed *source route* records and appends the target node's own signature. Figure 3-3 exhibits an example of our RAPA protocol; RREQ and RREP mean the Route Request and Reply respectively. When any node A received the Route Request for which it is not the target node, except of checking the recent Route Request Table in DSR, the node will first testify whether the three authentications of RAPA is valid. If the node finds the *path authenticator list* is not valid, the node can discard the packet. If the forwarder nodes determine the request packet is valid, they will generate the necessary signatures and piggyback these signatures which are including the authenticator for path authentication to the route request packet. If the target node settles the Route Request is legitimate, it returns a Route Reply to the initiator. Next section will detail the forwarder's behavior in RAPA.

$S : \mathfrak{R} \xleftarrow{R} [\pi_1 \dots \pi_i]$ $S : \Sigma_S \xleftarrow{S} [RREQ, \langle S, D \rangle, \mathfrak{R}]$ $S \rightarrow * : [RREQ, S, D, \mathfrak{R}, \Sigma_S, \langle SourceRoute \rangle, \langle pathAuthList \rangle, \langle forwarderSign \rangle]$ $A : \sigma_A \xleftarrow{\quad} [SK_A(\mathfrak{R})]$ $A : \Sigma_A \xleftarrow{S} [\Sigma_S, \langle S, A \rangle, \langle \sigma_A \rangle]$ $A \rightarrow * : [RREQ, S, D, \mathfrak{R}, \Sigma_S, \langle S, A \rangle, \langle \sigma_A \rangle, \langle \Sigma_A \rangle]$ $B : \sigma_B \xleftarrow{\quad} [SK_A(\mathfrak{R})]$ $B : \Sigma_B \xleftarrow{S} [\Sigma_S, \langle S, A, B \rangle, \langle \sigma_A, \sigma_B \rangle]$ $B \rightarrow * : [RREQ, S, D, \mathfrak{R}, \Sigma_S, \langle S, A, B \rangle, \langle \sigma_A, \sigma_B \rangle, \langle \Sigma_B \rangle]$ $D : \Sigma_D \xleftarrow{S} [RREP, D, S, \langle B, A, S \rangle]$ $D \rightarrow B : [RREP, D, S, \langle B, A, S \rangle, \Sigma_D]$ $B \rightarrow A : [RREP, D, S, \langle B, A, S \rangle, \Sigma_D]$ $A \rightarrow S : [RREP, D, S, \langle B, A, S \rangle, \Sigma_D]$

Figure 3-3 A general RAPA protocol example

3.2.3. RAPA Forwarder's Procedure

RAPA mainly denotes the behavior of the intermediate nodes after received a Route Request as the two phases, they are separately *RAPA_Verify* and *RAPA_Sign* and formally expressed as the follows; a forwarder node will first apply *RAPA_Verify* to check whether the Route Request is valid. If the output of *RAPA_Verify* is true then the forwarder will append the necessary signatures to the Route Request by using *RARP_Sign* and continue to deliver the Route Request, else the forwarder discards the Route Request in contrary.

Parameters: $t, k, n, \{SK_i^s, SK_i^p\}, f', f$
 t^s : t public slices for standard signature
 k^s : choose k private slices as signature
 n : n private slices assigned by initiator
 SK_i^s : node i 's private keys for standard signing process
 SK_i^p : node i 's private keys for path authentication process
 f' : previous hop node's identity
 f : current forwarder's identity
init : initiator's identity

RAPA_Verify {

Input: a route request message **RREQ**

$\{j_1 \dots j_k\} = \text{split Hash}(\mathbf{RREQ}[\text{initiator}, \text{target}, \text{assigned random list}])$ into k indexes of index' length is $\log_2 t$ bits

IF $PK_{init}^s \neq \text{Hash}(SK_{init}^s)$, or ... $PK_{init}^s \neq \text{Hash}(PK_{init}^s)$

THEN Output: FALSE

End-IF

$\{i_1 \dots i_k\} = \text{split Hash}(\mathbf{RREQ}[\text{initiator's signature}, \text{source route}, \text{hops authenticator list}])$ into k indexes of index' length is $\log_2 t$ bits

IF $PK_{f'}^s \neq \text{Hash}(SK_{f'}^s)$, and ... $PK_{f'}^s \neq \text{Hash}(PK_{f'}^s)$

THEN select all nodes i in $\{\text{source route}\}$

IF $PK_i^p \neq \text{Hash}(SK_i^p)$, and ... $PK_i^p \neq \text{Hash}(SK_i^p)$

THEN Output: TRUE

ELSE

Broadcast **RREQ** to announce the previous node f' as malicious nodes

End-IF

ELSE

Output: FALSE

End-IF

}

RAPA_Sign {

Input: a route request message **RREQ**

$\{\pi_1 \dots \pi_n\} = \mathbf{RREQ}[\text{assigned random list}]$

$\sigma_f = \{SK_f^p, \dots, SK_f^p\}$

Append σ_f to **RREQ**

split Hash(**RREQ**) into k indexes and interpret as an integer i_j for $1 \leq j \leq k$

$\Sigma_f = \{SK_f^s, \dots, SK_f^s\}$

replace Σ_f in **RREQ**

Forward **RREQ**

}

Figure 3-4 RAPA Forwarder's Signing and Verification Algorithm

Defending Cooperative Attackers

We review the attack phenomenon and demonstrate why the RAPA can protect the routing protocol from cooperative attack threats. The Figure 3-5 illustrates the attack scenario in protocol description. If the cooperative attackers M1 and M2 try to impersonate a Route Request then they can not be successful, because that HORS provides the asymmetric public key system- like property to allow communication nodes authenticating each others by applying their public key rather than the MAC sharing different secret key-pair between distinct authentication participants. In RAP protocol, the piggyback one-hop-far signature is not sufficient to prevent the cooperative attackers from inserting counterfeit route records which resulted in the pollution of the other's routing cache. The routing cache poison might cause overall performance of routing system down. In RAPA, we ask each forwarder node M1, M2, F to piggyback their identifier which is another private key in HORS for this route request message. Hence, if M1 and M2 try to modify the route request packets, such like inserting the node f1's address into the source route field, they must have f1's private slices first. If the M1 and M2 try to guess the f1's private slices, they have to counter the security strength guarded by HORS. We will discuss this security analysis of our scheme in Chapter 3.

$S \rightarrow * : [RREQ, S, D, \mathfrak{R}, \Sigma_S, \langle \rangle, \langle \rangle]$ $M1 : \sigma_{M1} \leftarrow [SK_{M1}(\mathfrak{R})]$ $M1 : \Sigma_{M1} \xleftarrow{S} [\Sigma_S, \langle S, A \rangle, \langle \sigma_A \rangle]$ $M1 \rightarrow * : [RREQ, S, D, \mathfrak{R}, \Sigma_S, \langle S, M1 \rangle, \langle \sigma_{M1} \rangle, \Sigma_{M1}]$ $M2 : \sigma_{M2} \leftarrow [SK_{M2}(\mathfrak{R})]$ $M2 : \Sigma_{M2} \xleftarrow{S} [\Sigma_S, \langle S, M1, M2 \rangle, \langle \sigma_{M1}, \sigma_{M2} \rangle]$ $M2 \rightarrow * : [RREQ, S, D, \mathfrak{R}, \Sigma_S, \langle S, M1, M2 \rangle, \langle \sigma_{M1}, \sigma_{M2} \rangle, \Sigma_{M2}]$ $F : RAPA_Verify$

Figure 3-5 Protocol description the cooperative attack scenario

3.3. A reactive security adjusting strategy

The routing disruption attacks we introduce in this paper prevent the normal nodes from finding a working route. The concept of authentication thwarts the impact of this misbehavior but also brings additional computational overheads which depress the performance of the routing system. If a secure routing protocol could lower the redundant security cost when the routing protocol is stable for constructing a workable route, it implies that the system might perform better than fixed-costs system even while there is no any disruption attacks happened. Since the quantity of the random number selected by initiator decides the security strength of this request route path. RAPA allows path authenticators which the forwarders should append to route request message to be randomly assigned by initiator. In other words, we provide the routing protocol with the capacity of adopting a strategy to dynamically settle the security level of request path before starting the route discovery phase. Along with the ability RAPA presented, we suggest a simple flexible strategy for initiator as the follows; *discovery_Timeout*, *discovery_Success* procedures would be triggered separately when the Route Discovery procedure is timeout, or successfully finding a route. In *RAPA_Init*, the initiator has to assign a set of random number for path authentication, it determine the amount of the random numbers according to the accumulated times of timeout in Route Discovery. The successive timeout in waiting a Route Reply will higher the security level and the cost. The *interval* can be decided depending on the demand but it supposes not to be over short, or it probably reacts slackly to consume more redundant cost.

Parameters: *acc_failTimes*, *max_level*, *interval*
acc_failTimes : accumulated failed times of routing discovery
max_level : max allowed security level of the routing system
interval : step for increasing security level
MaxFailTimes : max tolerant max failed discovery times

RAPA_Init {

Input:RREQ

$\mathfrak{R} = \text{MIN}(\text{acc_failtimes} * \text{interval}, \text{max_level})$

append the signature $\text{sign}(\text{RREQ})$ and random numbers set \mathfrak{R}

do original Route Discovery procedure

}

discovery_Timeout {

acc_failTimes ← **acc_failTimes** + 1

IF **acc_failTimes** < **MaxFailTimes**

THEN trigger RAPA_Init procedure

}

discovery_Success {

acc_failTimes ← 1

Prepare sending data procedure

}

Figure 3-6 A reactive adjusting strategy for Route Discovery procedure

3.4. Malicious nodes isolation feature

In our attack scenario, the malicious nodes try to add some forged routing record to route request message. We hope that the malicious nodes can be isolated from the network system by the RAPA after its disruption behaviors have been detected. In RAPA, each node is suggested to maintain a blacklist for blocking the malicious node. We consider a detection phenomenon as that the well-behaved forwarder can detect the malicious node by recognizing the forged records in path authenticators list of the polluted request message. Therefore in Figure 3-5, a well-behaved node F can expose the malicious node M2 by rebroadcast the polluted request message. This announcement message has the nonrepudiation property about the malicious node so that, after receiving this announcement message, every node will check the

correctness of the incrimination and then to add the malicious node to their own blacklists. In the RAPA protocol detailed in section 3.2.2, the forwarder's signature field is necessary for achieving the nonrepudiation for isolation method mentioned above. Without adding this signature, the forwarder node F is still able to discard the polluted request message but losing the ability to announce other nodes the information about the malicious nodes.



Chapter 4. Security Analysis

We discuss the security issues and properties of RAPA in this chapter. We divided it into three assessment phase: 1) source authentication 2) path authentication 3) cost comparison with the straightforward mechanism.

4.1. Security strength of standard signature for HORS

There are two important factors about the security considerations for HORS. They are separately the special feature for the hash operation on messages and the probability to forge a signature. The requirement for hash function on messages must have “Subset-Resilient” feature which make the indexes generated by this hash function on two messages impossible to conflict in the similar k_s -element subset of $\{1 \dots t_s\}$. The “Subset-Resilient” characteristic is formally defined and detailed in [HORS]. If the attacker tries to forge a signature after obtaining signatures on R messages, then the probability is trivial at most $(\frac{Rk_s}{t_s})^k$. With the parameters setting $R=4$, $k_s=1024$, $t_s=16$ the security strength is 2^{-64} . The security strength of HORS might need to initiate the rekeying procedure more frequently due to the significant diminution of the security strength.

4.2. Random Number Set Collision in Path Authentication

In general, we did not integrate directly the HORS into path authentication part of our scheme for which the security issues is related to the adaptive chosen message attack. The attacker try to invert the one way function H (which appeared in section 2.4) in the public slices for which the corresponding secret slices has not released in this attack. The security strength is reduced to the one-wayness and the collision-resistance of H. The security parameter of H is the bit length L of its input string. If attacker tries to fabricate the authenticator of a forwarder have to spend $k_p * 2^L$ guessing cost, k_p is the number of private slices for one authenticator in path

authentication. The parameters chosen k_p and L should make invert of H infeasible or the one-time signature system including HORS will be crashed. We use the term “authenticator” rather than “signature” for the staff appended by each forwarder of our path authentication mechanism because it needs no signing operation. As for non-adaptive chosen message attack, the attacker tries to forge the forwarder’s authenticator by choosing r released private slices. In general r is quite related the average number of neighbor nodes, the total number of nodes and the density of the network. In this regard, the path authenticators do not link to any messages, the partial released private slices is already proving the identity of the forwarder and the one-time signature also implicitly assume that the random number set issued by initiator can not be duplicated. Therefore, the forwarder nodes could refuse to append the corresponding authenticators and report back the initiator if the duplicated random number set has been detected. The two normal initiators might accidentally issue the same random number set that resulted the collision in the distributed phenomenon. We are interested in this collision probability under security concern. With the follow parameters; n for average number of neighbor nodes, Therefore, the average route request incoming rate for any intermediate nodes can be estimated as n , consider the probability $W(i, j)$ that no two packets out of i received route requests will match the same random number set from j different sets. Explicitly,

$$\begin{aligned}
 W(i, j) &= \frac{(j-1) \times (j-2) \times \dots \times (j-(i-1))}{j^i} \\
 &= \frac{j!}{(j-i)! j^i}
 \end{aligned}$$

,so the collision random set probability $P(i, j) = 1 - W(i, j) = 1 - \frac{j!}{(j-i)! j^i}$ this is similar as the birthday problem analysis, then we can estimate the collision probability as [19]

$$P(i, j) \approx 1 - e^{-i(i-1)/2j}$$

$$\approx 1 - \left(1 - \frac{i}{2j}\right)^{i-1}, \varepsilon < \frac{i^3}{6(j-i+1)}$$

This approximation probability is helpful for the purpose of determining the suitable time of rekeying with given parameters, t_p public slices, k_p private slices as authenticator in the path authentication, and also imply the possibility that the attacker can take use these disclosed authenticators for modifying the route request message. The collision probability should be raised with that the accumulated number of released private slices is increasing. Given a threshold value $P_{threshold}$, use the following algorithm to estimate average rekeying interval λ :

Input Parameters: $t_p, k_p, r, P_{threshold}$

t_p : t_p public slices for path authentication
 k_p : chose k_p private slices as authenticator
 r : r private slices out of t_p have been released
 n : average number of neighbor nodes
 $P_{threshold}$: max tolerated collision probability
 λ : average rekeying interval estimation

Estimate_RekeyInterval {
initialize: $\lambda \leftarrow 0$
loop do
 $\lambda \leftarrow \lambda + 1$
until $P(n, C_{k_p}^{t_p} - C_{k_p}^{n \times (\lambda - 1)r}) > P_{threshold}$
End-loop
Output: λ
}

Figure 4-1 A algorithm for estimating rekeying interval with a given threshold probability

In *Estimate_RekeyInterval* procedure, k_p is the max bound for number of average disclosed random number sets for each request forwarding path.

Consider this regular example in [7], the network with two hundred nodes, the communication area is approximately 400 meter square and the communication range is around one meter for each node, we estimated three nodes as the average number of neighbors which is obtained from this statistical study [8]. We take a example security

parameters setting in HORS as $t_p=1024$, $k_p=16$ and , observing a particular probability $\Psi(n)$ that one initiator's random number set conflicts with others in the intermediate node is as the following:

$$\begin{aligned}\Psi(n) &= \frac{n \times C_{k_p}^r}{C_{k_p}^{t_p}} \\ &= n \times \frac{r!(t-k)!}{t!(r-k)!}\end{aligned}$$

Substitute factorial of N as the follows by using Stirling's approximation,

$$N! \approx \sqrt{(2N + \frac{1}{3}) \times \pi} \times N^N \times e^{-N}$$

to simplify the equation $\Psi(n)$, and obtain,

$$\Psi(n) = n \times \sqrt{\frac{r \cdot (t-k)}{t \cdot (r-k)}} \times \left(\frac{r}{r-k}\right)^r \times \left(\frac{t-k}{t}\right)^t \times \left(\frac{r-k}{t-k}\right)^k$$

With the above example parameters set above, the probability is 9.8301e-006 and 0.015 after 512, 768 private slices have been disclosed. This probability is directly related to the rejecting request probability for a forwarder and also reflecting the adaptability of our proposed scheme. Therefore, we can take account of this factor for the rekeying procedure. Besides it also strongly related the performance of our proposed scheme because higher collision probability caused tremendous rejecting ratio for request forwarding process. The following figure will show more clearly the variation of $\Psi(n)$ when different security parameter is set. Figure 4-2 plots this probability versus the x-axis which presents the accumulated private slices disclosed by one initiator. From the gradient of the two curves of the Figure 4-2, It indicates that the higher security parameter set indeed generate the lower collision probability but the higher $\{t_p, k_p\}$ setting also have more cost on either computational or transmission requirement.

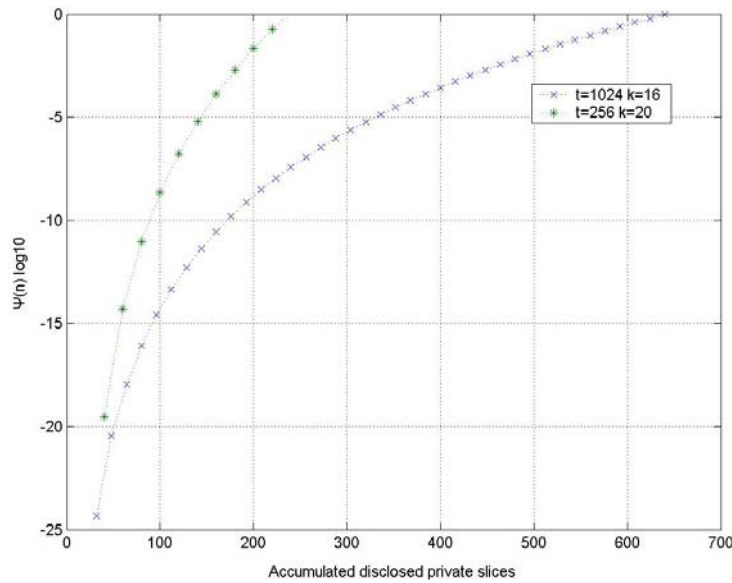


Figure 4-2 Variation of $\log_{10}(\Psi(n))$ in different parameters setting

We use Figure 4-3 and Figure 4-4 to discuss the variation of $\Psi(n)$ between different setting for t_p, k_p . In Figure 4-3, the left side show the quantity scale of $\Psi(n)$ and the right side is the expected value of collision with all the same amounts of public slices setting. The expected value of collision reflects the reference number of disclosed private slices after which the rekeying procedure could be initiated. The double k_p number is helpful for slowing the growth rate of the collision probability. For $k_p=16$, the collision would be expected to reveals after 448 private slices disclosed and this tolerated number of exposed private slices can increase 200 private slices when the k_p parameter is double. The Figure 4-4 fixes k_p as 16, it shows that the tolerated number of exposed private slices only increase 100 scale from $t_p=1024$ to $t_p=1280$.

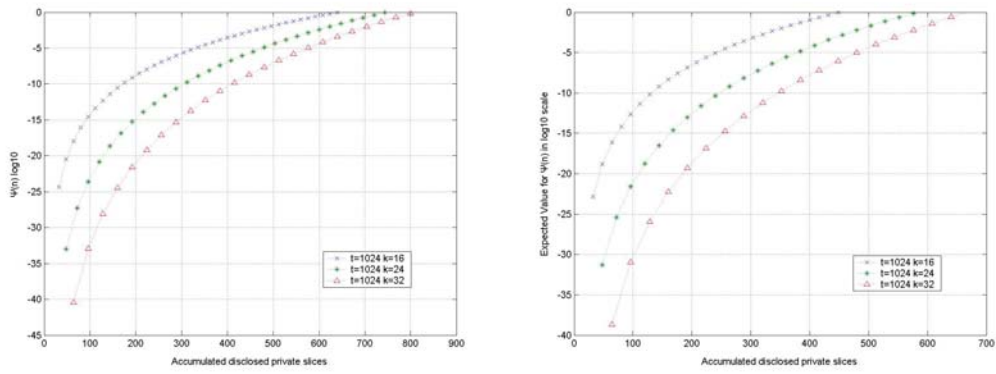


Figure 4-3 Variation of $\log_{10}(\Psi(n))$ and the expected value for the same number of public slices

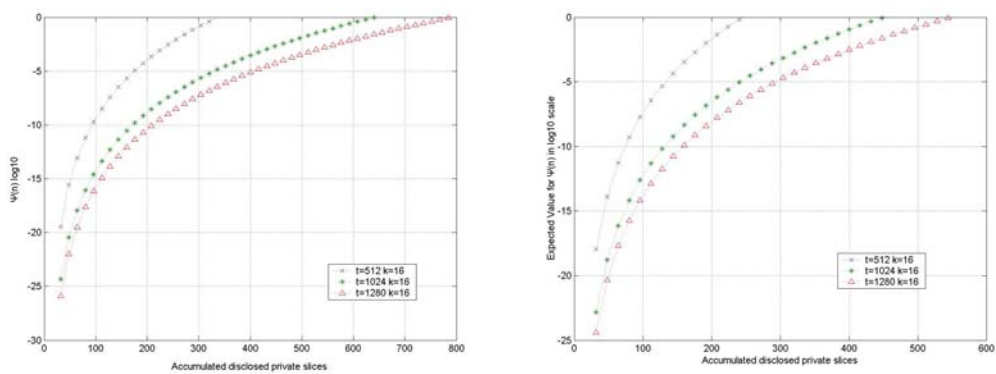


Figure 4-4 Variation of $\log_{10}(\Psi(n))$ and the expected value for the same number of private slices

The above analysis exhibits that the collision probability of the random number set chosen by an initiator is not only ignorable but also growing slowly with the disclosed private slices accumulated. We see the insignificant performance impact brought by our path authentication scheme and also show its practicability on the disclosed number of private slices.

4.3. Comparison with Overall Cost for Forwarding Route Request

In this chapter, we compare the verification and signing cost between our proposed scheme and a straightforward method. Based on the harsh attack model, a straight-forward solution with piggybacking all signatures of each hop could bring a lot of either bandwidth or verification cost for the routing system. We consider a

straightforward way for path authentication which asks each forwarder to piggyback their signatures rather than the “authenticators” in our scheme. So the forwarder should apply a hash operation on the forwarding route request message to get the index of private slices for signing the request. We consider the amounts of hash operation to evaluate the comparative cost. In RAPA, we conserve more verification cost than the straightforward way with that the request route path increases, because our method needs no the hash operation on messages for indexing the private slices.

Normally speaking, the Table 4-1 generalized the verification and signing cost of the forwarders for a route request path of length L in both our scheme and the straightforward method. The symbol σ_i denoted the number of previous hops through which a route request packet has passed at the i th intermediate node on route request path.

	RAPA Scheme	A Straightforward Method
Signing cost of i th intermediate node	1	1
Verification cost of i th intermediate node	$(k_p \times \sigma_i) + (k_s + 2)$	$((k_p + 1) \times \sigma_i) + (k_s + 1)$
Overall Cost	$(k_p \times \sum_2^L \sigma_i) + (L - 1) \times (k_s + 3)$	$((k_p + 1) \times \sum_2^L \sigma_i) + (L - 1) \times (k_s + 2)$

Table 4-1 Generalized hash operation costs on a route request path of length L .

The overall cost is calculated as the summation of signing and verification cost for the whole intermediate nodes. As for initiators, The Figure 4-5 illustrate the cost difference with a given parameters $k_p=k_s=16$ for lower curve with dot markers. Comparison with the following scenario under the same parameters $t_p=1024$, $k_p=16$, if the disclosed private slices are half amounts of total public slices, the security strength (forging signature probability) HORS can provide is $\frac{1}{2^{-16}}$ but the collision

probability presented by our scheme is $\frac{1}{2^{-115}}$. Therefore we can reduce the scale of k_p in our scheme but achieve the same security strength with the HORS. The estimation for appropriate k_p can apply the equation appeared in *Estimate_RekeyInterval* procedure of section 3.2. We could assume $P(3, C_{k_p}^{t_p} - C_{k_p}^{t_p/2}) = \frac{1}{2^{-16}}$, $t_p=1024$ and solve k_p . The k_p is at most 4 and our scheme can lower the collision probability if we settle higher k_p for higher cost. The line marking asterisks of Figure 4-5 shows the cost difference between our scheme and HORS with the same above security assumption.

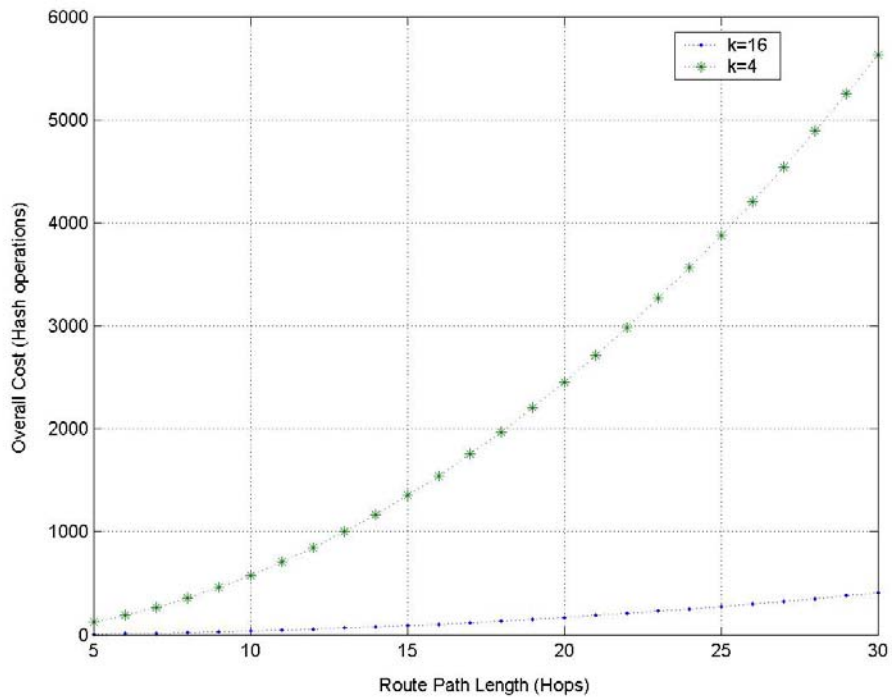


Figure 4-5 Cost difference between RAPA and the straightforward way

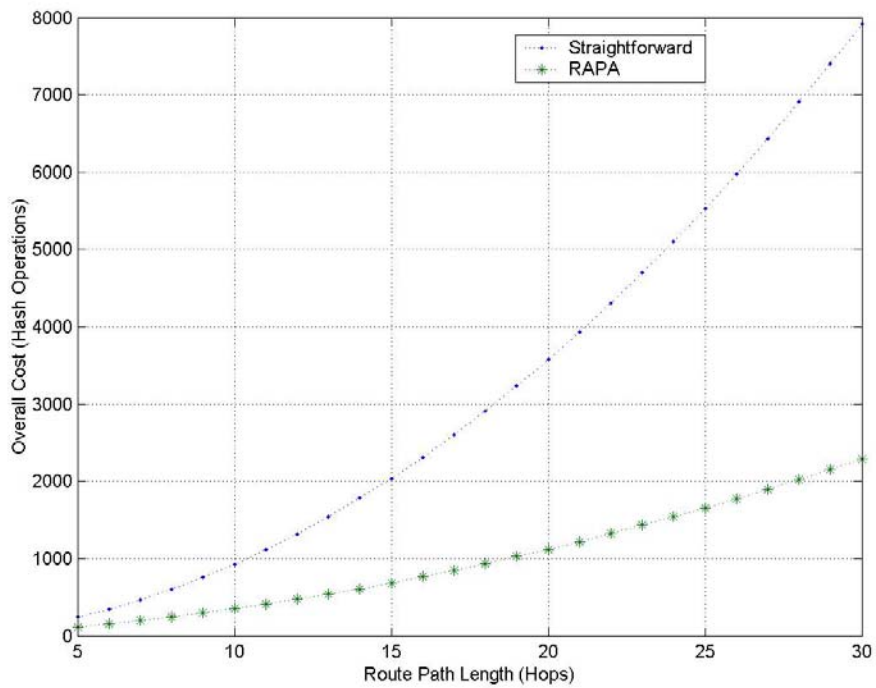


Figure 4-6 Overall Cost comparison

Obviously in Figure 4-6, the difference grows increasingly when request route is longer. It also implies that our scheme can get better performance with larger network scalability. Comparison to a straightforward method, we did economize on the computational cost of the overall system with the increased length of route path.

Chapter 5. Conclusions

Secure routing protocol is one of the most challenging areas in ad hoc network security. The routing mechanism is harder to efficiently secure and more vulnerable in ad hoc network than wired networks because every device could perform as the forwarding router. Any misbehavior like disseminating the false routing information could paralyze the whole routing system and the cooperative attacks can result in the serious threats for networking. Since the resource-constrained factor for mobile environment, security mechanisms should respect both the efficiency and cost consideration.

In this paper, we describe a specific cooperative attack model against the current secure mechanisms for DSR routing protocol. Under this attack phenomenon, adversaries can perform various routing disruption misbehaviors and bring a serious threat on routing system. We also proposed a solution, named Random Assignment Path Authentication, which provides lower computational cost than a straightforward way for protecting the routing information on an entire route path from spitefully alternation. Due to the non-repudiation property provided from the adopted source authentication mechanism, we suggest to use blacklist for prohibiting the malicious nodes from joining the network. Our scheme also provides a flexible feature to adjust security level for a route discovery phase. It did present the capability for developing the possible strategy in balancing the performance and security strength according to deployed policy.

Chapter 6. References

- [1] Adrian Perrig, Ran Canetti, Dawn Song, and J. D. Tygar. “Efficient and Secure Source Authentication for Multicast.” In *Network and Distributed System Security Symposium* (NDSS '01), pages 35-46, February 2001.
- [2] Adrian Perrig. “The BiBa one-time signature and broadcast authentication protocol.” In *Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 28-37, November 2001.
- [3] Asad Amir Pirzada, Chris McDonald. “Kerberos assisted Authentication in Mobile Ad-hoc Networks.” In *Proceedings of the 27th conference on Australasian computer science*, Volume 26, pages 41-46, January 2004.
- [4] Baruch Awerbuch, David Holmer, Cristina Nita-Rotaru and Herbert Rubens. “An on-demand secure routing protocol resilient to byzantine failures.” In *Proceedings of the ACM workshop on Wireless security* (WiSe'02), pages 21-30, September 2002.
- [5] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir R. Das. “Ad Hoc On Demand Distance Vector (AODV) Routing.” Internet-Draft, draft-ietf-manet-aodv-13.txt, February 2003. Work in progress.
- [6] D. B. Johnson, D. A. Maltz, and J. Broch, “DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks.” In *Ad Hoc Networking*, ch. 5, pp. 139--172. Addison-Wesley, 2001.
- [7] David B. Johnson, David A. Maltz, Yih-Chun Hu, and Jorjeta G. Jetcheva. “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks.” Internet-Draft, draft-ietf-manet-dsr-09.txt, April 2003.
- [8] Hui Li and Dan Yu. “A statistical study of neighbor node properties in ad hoc network.” In *Proceedings of the International Conference on Parallel Processing*

Workshops (ICPPW'02), pages 103-108, August 2002.

- [9] Jean-Pierre Hubaux, Levente Buttyán and Srdan Capkun. “The Quest ofr Security in Mobile Ad Hoc Networks.” In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking and computing (MobiHOC'01)*, pages 146-155, October 2001.
- [10] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta G. Jetcheva. “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols.” In *Proceedings of the Fourth ACM/IEEE International Con-ference on Mobile Computing and Networking (MobiCom'98)*, pages 85–97, Oc-tober 1998.
- [11] Kimaya Sanzgiri, Bridget Dahill, Brian Neil Levine, Clay Shields, Elizabeth M. Belding-Royer. “A *Secure Routing Protocol for Ad Hoc Networks.*” In *Proceedings of 10th IEEE International Conference on Network Protocols (ICNP'02)*, pages 78-87, November 2002.
- [12] Laurent Eschenauer, Virgil D. Gligor. “Key management and key exchange: A key-management scheme for distributed sensor networks.” In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 41-47, November 2002.
- [13] Leonid Reyzin and Natan Reyzin. “Better than BiBa: Short One-Time Signatures with Fast Signing and Verifying.” In *Proceedings of the 7th Australian Conference on Information Security and Privacy*, pages144-153, July 2002.
- [14] Manel Guerrero Zapata and N. Asokan. “Securing Ad hoc Routing Protocols.” In *Proceedings of the 2002 ACM workshop on Wireless security (WiSe'02)*, pages 1-10, September 2002.
- [15] Manel Guerrero Zapata. “Secure Ad hoc On-Demand Distance Vector (SAODV) Routing.” Internet-Draft, draft-guerrero-manet-saodv-00.txt August 2002. First

published in the IETF MANET Mailing List (Oct. 8th 2001).

- [16] P. Papadimitratos and Z. J. Hass. “Secure Routing for Mobile Ad hoc Networks.” In *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS’02)*, January 2002.
- [17] Per Johansson, Tony Larsson, Nicklas Hedman, Bartosz Mielczarek, and Mikael Degermark. “Scenario-based Performance Analysis of Routing Protocols for Mo-bile Ad-hoc Networks.” In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom’99)*, pages 195–206, August 1999.
- [18] S. Capkun, L. Buttyan and J.-P. Hubaux. “Self-organized public-key management for mobile ad hoc networks.” In *IEEE Transactions on Mobile Computing*, pages 52-64, March 2003.
- [19] Sayrafiezadeh, M. “The Birthday Problem Revisited.” *Mathematics Magazine* 67, pages 220-223, 1994.
- [20] Sencun Zhu, Shouhuai Xu, Sanjeev Setia and Sushil Jajodia. “Establishing Pairwise Keys for Secure Communication in Ad Hoc Networks: A Probabilistic Approach.” In *Proceedings of the 11th IEEE International Conference on Network Protocols*, pages 326, November 2003.
- [21] Sergio Marti, T. J. Giuli, Kevin Lai and Mary Baker. “Mitigating routing misbehavior in mobile ad hoc networks.” In *Proceedings of the 6th annual international conference on Mobile computing and networking (MOBICOM 2000)*, pages 255-265, August 2000.
- [22] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. “Wormhole detection in wireless ad hoc networks,” Department of Computer Science, Rice University, Tech. Rep. TR01-384, June 2002
- [23] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. “Ariadne: A Secure

On-Demand Routing Protocol for Ad Hoc Networks.” In *Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking (MobiCom’02)*, pages 12-23, September 2002.

- [24] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. “Rushing attacks and defense in wireless ad hoc network routing protocols.” In *Proceedings of the 2003 ACM workshop on Wireless security (WiSe’03)*, pages 30-40, September 2003.

