ABSTRACT

IYER, VARAGUR K. Misbehavior Resistant Fair Scheduling in Wireless Backhaul Mesh Networks. (Under the direction of Dr. Peng Ning).

Wireless Mesh Networks (WMNs) are increasingly becoming popular as a means to realizing cost-effective and efficient public area networks. WMNs provide improved coverage with relatively cheaper infrastructure and are easy to deploy and expand as compared to Wireless LANs. This is possible due to the use of wireless multihop routing to forward traffic from nodes to a Gateway (GW) that has a wired connection to the Internet. However, there are several issues that need to be addressed to realize the practical deployment of large scale mesh networks serving a vast number of clients.

Fairness is one such key issue of concern in WMNs, however, WMNs based on existing 802.11 technology exhibit severe unfairness. Several TDMA based bandwidth scheduling schemes have been proposed as an alternative to ensure fairness in WMNs. Such schemes implicitly trust the nodes in the network and as a result, are vulnerable to threats due to misbehavior by nodes participating in the scheduling. These threats are further amplified in public area mesh networks where nodes have highly varying demand and are vulnerable to capture. While a lot of research has been done on securing routing and topology control protocols against misbehavior, the effects of misbehavior on fair scheduling have not been considered. In this thesis, we address the threats to fair scheduling in wireless backhaul mesh networks from node misbehavior and present a generic verification framework to detect such misbehavior. The verification framework is based on the argument that, while the actions of an individual node might not be trusted, the collective action of all the nodes in the network can be trusted since an attacker is assumed to be incapable of compromising every node in the network. We propose two verification schemes based on this framework, each designed for a particular deployment environment. We conduct an experimental evaluation of the verification schemes by using them to extend an existing fair scheduling scheme. The experimental results show that a fair scheduling scheme relying on the exchange of demand information can be extended by an instance of the verification framework to ensure misbehavior detection while incurring a relatively small amount of overhead.

Misbehavior Resistant Fair Scheduling in Wireless Backhaul Mesh Networks

by
Varagur Karthik Iyer

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fullfillment of the
requirements for the Degree of
Master of Science

Computer Science

Raleigh, North Carolina

2008

APPROVED BY:

Dr. Rudra Dutta

Dr. Ting Yu

Dr. Peng Ning
Chair of Advisory Committee

DEDICATION

To my family

BIOGRAPHY

Varagur Karthik Iyer was born on 9th April, 1985. He received his Bachelor of Technology degree, majoring in Information and Communication Technology, from Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT), Gandhinagar, Gujarat, India. He is currently a graduate student in the Department of Computer Science, North Carolina State University, Raleigh, USA.

ACKNOWLEDGMENTS

I am extremely grateful to my adviser Dr. Peng Ning for his invaluable guidance and support. He has been a constant motivator throughout my research and has imparted some precious lessons that I will remember for the rest of my life. I am also grateful to Dr. Ting Yu and Dr. Rudra Dutta for taking time off their busy schedules and agreeing to be on my thesis advisory committee, their feedback and suggestions have been very helpful in making my work better.

I would like to specially thank Zhibin Wu for providing the simulation code for CIRMA-MH and being very forthcoming in answering my questions regarding the same, his help has immensely aided in boosting the credibility of my research. I would also like to acknowledge Archana Rajagopal, my project partner with whom I had first begun to define my research direction. I would like to thank all my colleagues at the cyber defense lab for all the good times that we have had in the lab. Finally, I am grateful to my family for their constant love and encouragement.

TABLE OF CONTENTS

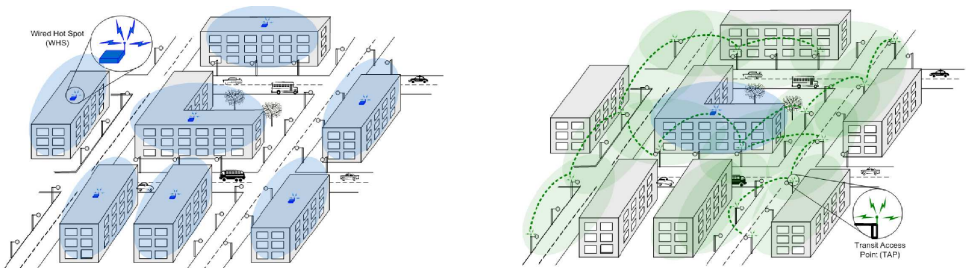
LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Wireless Mesh Networks (WMNs) are increasingly becoming popular as a means to realizing cost-effective and efficient public area networks. WMNs provide improved coverage with relatively cheaper infrastructure and are easy to deploy and expand as compared to Wireless LANs. This is possible due to the use of wireless multihop routing to forward traffic. As shown in Figure 1.1, a WMN with a single Gateway (with a wired connection to the Internet) and a number of Transit Access Points (TAPs) can cover the same area as a WLAN but at a much lower cost which makes them particularly attractive for modern public area wireless networks. In addition, WMNs also exhibit useful characteristics like self-healing and self-organization thus providing reliable and robust service.



(a) (b)

Figure 1.1: Two approaches to providing Internet connectivity. (a) WiFi Network: several Wireless Hot Spots (WHSs) are needed to offer good coverage of a given area; (b) Wireless Mesh Network (WMN): by using one WHS/GW and several Transit Access Points (TAPs), it is possible to cover the same area as in (a); the TAPs rely on the WHS to transmit their traffic to and from the Internet. [1]

1.1 Background

A generic WMN is made up of two types of nodes, Transit Access points (TAPs) which act as intermediate nodes that forward traffic and client nodes which are the end points of the network that generate and receive traffic. The client nodes in a WMN are also capable of forwarding traffic. The WMN architecture can be broadly classified into three categories [5] (i) infrastructure/backbone mesh networks (ii) client mesh networks and (iii) hybrid mesh networks. These categories differ in the ability of the component nodes to forward traffic. The focus of this work is on infrastructure mesh networks which are used to provide wireless backhaul service to clients. An infrastructure mesh network shown in Figure 1.2, consists of TAPs which form a static wireless backbone connecting the client nodes to a wireless Gateway (GW). Client nodes connect to this backbone by associating themselves with the closest TAP and at any given time, a client is assumed to be associated with one and only one TAP. The mesh network may have one or more wireless Gateways, however in this thesis, we consider an infrastructure wireless mesh network with a single Gateway. All traffic in the network is assumed to flow between the client nodes and the gateways with the TAPs acting as intermediate nodes which only forward the traffic.

A WMN offers a variety of services to its users such as Broadband Internet access. It can be ob-

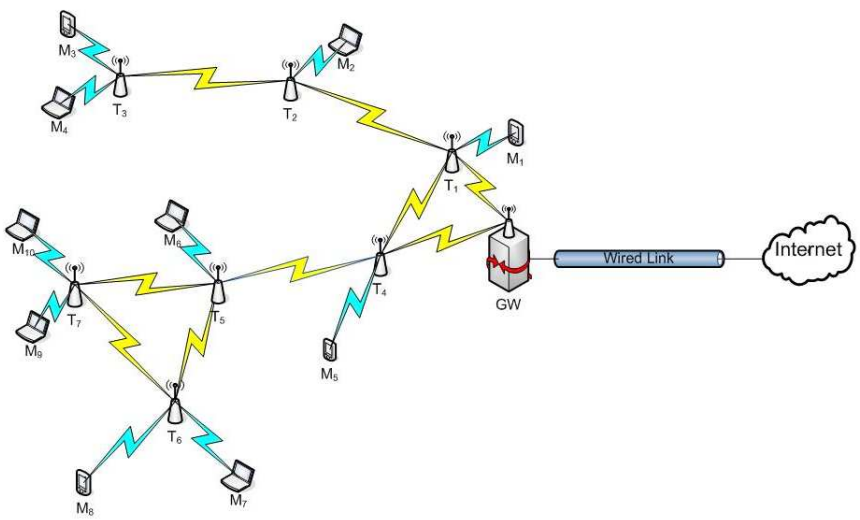


Figure 1.2: Infrastructure Wireless Mesh Network

served that such a deployment should be accompanied by certain fairness guarantees. In this thesis, a public area wireless network (PAWN) [6] realized using a WMN is considered. Such a network is

characterized by (i) a large number of clients, (ii) presence or absence of billing and accounting, (iii) highly varying demand with respect to topology and time, and (iv) a highly transient relationship between the client and the access points that make up the network. The network should guarantee that the throughput received by clients is maximized while ensuring fair allocation of bandwidth. However, mesh networks using 802.11 technology suffer from severe spatial unfairness as the throughput received by nodes farther from the Gateway is far less than the throughput received by nodes which are closer. TDMA based fair scheduling schemes which regulate the amount of bandwidth being allocated to the nodes to achieve fairness while maximizing the throughput of the network have been considered as an alternative to contention based solutions. Both centralized [3] [7] [8] and distributed [9] [10] [8] fair scheduling schemes that operate on different deployment environments and under different fairness granularities have been proposed. These scheduling schemes depend on traffic load information or in other words demand information for fair allocation of bandwidth among the nodes in the network. As a result, control messages containing traffic demand information have to be exchanged between nodes in the network in order to enable these schemes. Hence, it is very critical that these control messages be accurate and reflect the true demand in the network.

1.2 Motivation

The nature of WMNs makes them vulnerable to a number of threats, one of them being node (TAP/Client) misbehavior. Since the use of cheap infrastructure is one of the attractive features of a WMN, it cannot guarantee that all the components of the network are physically secure against tampering by an adversary. The TAPs in a PAWN are usually deployed in public places like roof tops where they are easily accessible and do not have the security mechanisms in place to protect against capture, tampering and replication. As a result, they become active targets of any adversary looking to compromise and gain entry into the network and the same can be said about the client machines which access the WMN. The compromised TAPs and client nodes can pose a serious threat to the fairness guarantees of a WMN. Most scheduling schemes assume that the TAPs in the mesh network report the correct demand in their respective neighborhoods in order to enable the computation of an efficient schedule. For example, in the fair scheduling scheme proposed in [9], every TAP distributes local demand information to all other TAPs in the network thus enabling every TAP to have a global picture of demand in the network. In the IFA scheme proposed in [10], a TAP propagates local demand and link capacity information along the branch to the GW to enable computation of bandwidth allocation for each TAP. These schemes implicitly trust the nodes to provide

the accurate demand information and do not make provisions for the situation where one or more of the nodes in the network might be compromised and could manipulate this demand information. As a result, these scheduling schemes open up to a number of threats in the presence of compromised nodes.

1.3 Contributions

We have made the following contributions in this thesis.

- We address the threat of node misbehavior on fair scheduling in Wireless Mesh Networks for the first time in this thesis. We outline the threat model and discuss several attacks that can affect the performance of a fair scheduling scheme in the presence of compromised nodes. We also illustrate the impact of one type of node misbehavior on an existing fair scheduling scheme called CIRMA-MH proposed in [3] and show that with relative ease an adversary can cause a severe reduction in the performance of a network.

- We present a generic verification framework for the detection of node misbehavior during fair scheduling and outline the security properties that must be satisfied by a scheme that implements this framework. We also define the performance metrics that are used to evaluate such an implementation. We propose two concrete misbehavior detection schemes based on the verification framework each suited for a particular deployment environment.

  - The first scheme is a digital signature based scheme that is designed for a distributed authentication environment where, digital signatures are used to generate verifiable commitments and demand claims.
  - The second scheme is a HAFS [11] based verification scheme implemented using symmetric key cryptographic primitives. It is designed for a centralized authentication environment and makes use of symmetric key encryption and Message Authentication Codes (MAC) to generate verifiable commitments and demand claims.

- We perform an experimental evaluation of both the verification schemes by extending CIRMA-MH [3] with these solutions and measuring the performance metrics defined in the verification framework. Based on the experimental results, we demonstrate that a fair scheduling scheme

based on exchange of demand information can be augmented with an instance of the verification framework to guarantee detection of misbehavior while incurring minimal overhead.

1.4 Thesis Organization

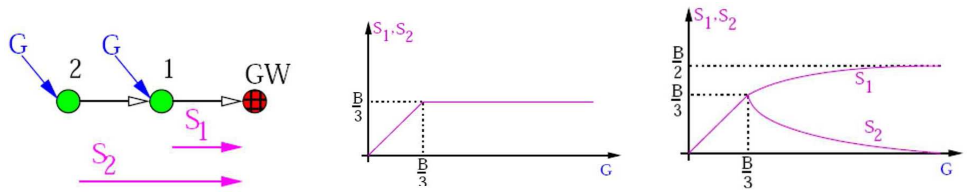
The rest of this thesis is organized as follows, in Chapter 2, we discuss the background and related works pertaining to fair scheduling in mesh networks and the security challenges to scheduling in wireless mesh networks. In Chapter 3, we present the assumptions about the network environment, the threat model and discuss several threats to fair scheduling as a result of node misbehavior. We also illustrate the impact of a certain type of misbehavior on an existing scheduling algorithm proposed in [3]. In Chapter 4, we outline a generic verification framework for detection of misbehavior during fair scheduling. We outline the different stages of the verification framework, the security properties and the performance metrics relevant to this framework. In Chapter 5, we propose a digital signature based verification scheme that is designed for a network with distributed authentication and discuss two variants of this scheme under different assumptions about the misbehavior in the network. We perform a theoretical analysis of the security and performance of this scheme and propose a solution to optimize the overhead. In Chapter 6, we propose a symmetric key based verification scheme for detecting misbehavior in a network with centralized authentication and perform a theoretical analysis of the security and performance of this scheme. In Chapter 7, we conduct an experimental evaluation of the schemes proposed in Chapters 5 and 6 by extending the CIRMA-MH scheduling algorithm to incorporate each of these schemes. We analyze the verification schemes with respect to the performance metrics of the verification framework and discuss their merits and demerits. Finally in Chapter 8, we present our conclusions and propose future directions for research on this topic.

Chapter 2

Background and Related Work

2.1 Fairness in Wireless Mesh Networks

It has been shown that in 802.11 based networks, the throughput received by a flow is inversely proportional to the number of hops it traverses as shown in Figure 2.1. One reason for this is that, in a contention based MAC protocol such as 802.11, a flow traversing a larger number of hops has to contend for the channel more than a flow traversing a lesser number of hops. Several works including [2] [10] [12] [13] have addressed the problem of unfairness present in 802.11 based mesh networks. These works look at the problem from the perspective of maximizing the throughput of the network while preserving fairness guarantees to prevent starvation of nodes more than one hop away from the gateway. While these works present solutions and highlight the improvements to the throughput received by the different nodes in the mesh network, they work under an ideal assumption that all nodes in the network are trusted and do not consider the robustness of the solution in the presence of malicious entities in the network.



(a) (b) (c)

Figure 2.1: (a) Fairness study of a two-node network forwarding packets to a gateway GW. The ideal (b) and real (c) throughputs of nodes 1 and 2 as a function of the offered load G [2]

Fairness in mesh networks can be achieved at two levels (i) per TAP fairness and, (ii) per client fairness. While per TAP fairness guarantees that all TAPs in the network get an equal share of the bandwidth, per client fairness guarantees that all clients in the network receive an equal share of the bandwidth. Either of these fairness guarantees is applicable depending on the deployment environment. The per client fairness guarantee would be applicable in a deployment environment such as a public area network where clients are provided free or subsidized access to the network. All clients are assumed to pay the same flat rate for access to the network and each client has an independent contract with the network. In other deployment environments like community networks, the per TAP fairness guarantee is applicable. In such a network, a cluster of entities in the community might have a single contract with the network and are assigned a separate TAP, and all clusters pay the same flat rate for the access to the network.

2.2 Fair Scheduling in Wireless Mesh Networks

Scheduling based on Time Division Multiple Access (TDMA) is considered as an alternative to ensure fairness in wireless mesh networks since it precludes some of the problems inherent to contention based multiple access schemes which cause unfairness. In a TDMA based MAC protocol, time slots are assigned to nodes for data transmission and this assignment is specified in the form of a transmission schedule. A specific variant of TDMA called Spatial TDMA (STDMA) [14] is used for scheduling in wireless multihop networks to enable spatial reuse. STDMA produces a more efficient transmission schedule compared to TDMA since it assigns the same timeslot to multiple nodes which do not interfere with each other. A scheduling scheme in which the schedule is computed in such a way as to ensure an equal share of bandwidth to the nodes in the network is referred to as a fair scheduling scheme. Several proposed approaches to fair scheduling based on the different levels of fairness discussed above and affected by the problem that is addressed in this work are outlined below.

In [9] Salem and Hubaux propose a distributed scheduling algorithm that ensures per client fairness. The scheme employs spatial reuse as it uses a STDMA approach to assign transmission rights to links in the network. The scheme involves the computation of a compatibility matrix to determine links in the network that can be activated simultaneously, the matrix is used to form cliques made up of links that can be active simultaneously. A transmission schedule is calculated which assigns timeslots to each of the cliques based on the maximum amount of traffic forwarded

by a link in the clique. The proposed solution requires the scheduling algorithm to be executed every time clients enter or leave the network, this requires that demand information to be exchanged whenever there is a change in demand in the network. It is implicitly assumed that all nodes advertise accurate demand information such that all TAPs in the network arrive at a schedule consistent with the existing demand in the network. The authors do not address the robustness of the scheme in the event that the demand information being exchanged is inconsistent.

In [10] Gambiroza, Sadeghi and Knightly propose a distributed layer 2 fair scheduling scheme called the inter TAP fairness algorithm (IFA) that ensures per TAP fairness. They provide a general design of the layer 2 scheme briefly outlining the different stages such a scheme must implement. The stages of the scheme are (i) Measurement of Offered Load and Capacity (ii) Message Distribution (iii) Aggregate Fair Share Computation and (iv) Ingress Rate Limiting. The fair share computation phase in IFA relies on the exchange of offered loads and link capacities among the nodes in a branch in order to enable TAPs to compute their fair shares and limit the traffic from client nodes. The IFA scheme does not address the issue of node misbehavior since it assumes that all TAPs in the network are trusted and are able to accurately estimate the offered load.

In [3] Wu, Ganu and RayChaudhuri propose IRMA, an Integrated Routing and MAC scheduling protocol for wireless mesh networks which implements a cross layer scheduling scheme that maximizes throughput and ensures fairness in the network. They argue that the 802.11 MAC protocol does not perform well with adhoc routing protocols such as AODV, DSR or DSDV and propose both a centralized and a distributed version of the scheduling algorithm that allocates resources based on the traffic flow specifications and the network compatibility graph. Two algorithms (i) Link scheduling with Min-Hop routing (IRMA-MH) and (ii) Link scheduling with Bandwidth aware routing (IRMA-BR) are proposed and the authors show through simulation results that both these schemes produce a significant improvement in throughput in the network. However, the authors do not evaluate the fairness or performance of the schemes in the presence of inconsistent traffic information from the nodes in the network.

In [8] Li suggests that while the End-to-end fair scheduling schemes for wireless adhoc networks which break down multihop flows as single hop flows may preserve the basic fairness constraint, they might not maximize spatial reuse in the network. The author proposes an algorithm that maximizes the spatial reuse while still preserving the fairness guarantees of the network. A centralized version of this algorithm is first outlined, where a central processing node collects per-flow information about flows originating from all the other nodes in the network. Based on this information, the node computes a weighted subflow contention graph. The node then computes the

optimum allocation strategy for all the nodes in the network and broadcasts this strategy. The author also describes a distributed version of the algorithm where nodes exchange per-flow information locally to arrive at a local optimum allocation strategy. While both these schemes depend on per-flow information originating from the nodes in the network, they do not have any means of verifying the accuracy of the information that is presented.

In [7] Sarkar and Tassiulas also consider the problem of End-to-end fairness guarantees in wireless adhoc networks and propose a centralized algorithm. The algorithm consists of a link scheduling phase which requires global coordination among the nodes in the network. The fair rate computation is based on a token generation process at each node for the flows that traverse through the node. The transmission of packets belonging to a flow are scheduled according to the token generation rate for the respective flow, this rate is adjusted so as to finally converge to the token generation rate of the bottle-neck node in the path of the flow. The scheme requires exchange of scheduling state information among one hop neighbors in order to compute the token generation rate of a session.

In [15] Salonidis and Tassiulas present the framework for a dynamic distributed TDMA scheduling scheme for End-to-end rate guarantees in wireless adhoc networks. The framework requires a distributed coordination mechanism among nodes in a contention region to adapt to the changing demand conditions. The authors outline a dynamic link scheduling algorithm called STABLE_TREE that makes use of the distributed coordination among nodes to compute an optimum link schedule. During the distributed coordination phase the nodes participate in link rate adjustment when local demand at the nodes and the capacity of the links change. As part of link rate adjustment both endpoints of a link exchange the current schedule related information using the SC_INFO packets and compute a new slot assignment which satisfies the demand for the given link undergoing adjustment. The nodes then propagate this new assignment to all the affected neighbors in the contention region using the SC_UPD packets, the neighbors use the information from this packet to perform the link rate adjustment on links that are affected by this adjustment.

2.3 Centralized Integrated Routing and MAC scheduling with Min Hop Routing (CIRMA-MH)

In this section, we discuss the centralized version of the Integrated Routing and MAC scheduling with Min hop routing (CIRMA-MH) scheme that has been proposed in [3]. This scheme

is of particular interest since it is very similar to the type of fair scheduling scheme referred to in this thesis and exhibits the problems that our solution looks to address. A slightly modified version of this scheme can be extended to include an instance of the verification framework proposed in this thesis and as a result CIRMA-MH is a potential candidate scheduling scheme that can be used to evaluate the security and performance of our verification framework.

CIRMA-MH is a scheduling scheme for adhoc wireless mesh networks where there exists a centralized node called the master node that collects bandwidth requests and generates a TDMA schedule for the different traffic flows in the network. The scheduling scheme assumes the existence of a global control plane on which the topology and traffic information are exchanged between the nodes and the master node, and the scheduling algorithm depends on this signaling information to establish routing and generate TDMA schedules. The mesh network considered is referred to as a Control-Based Wireless Mesh Network (CB-WMN) and its architecture is presented in Figure 2.2.

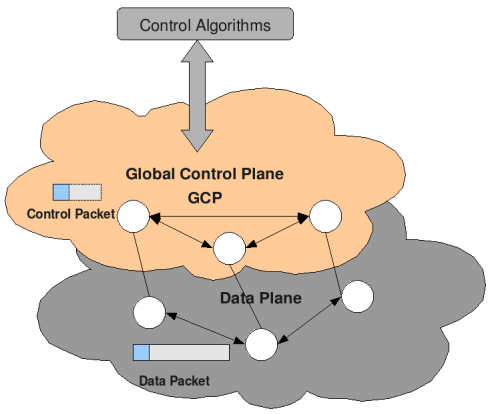


Figure 2.2: CB-WMN Network Architecture [3]

In CIRMA-MH, a control agent is associated with each node in the network and is responsible for monitoring traffic flow and exchanging topology information in order to set up routes both in the control and data plane. During network initialization, all nodes are expected to check in with the master node. The control agents then begin the routing and neighbor discovery phase during which they send periodic control route updates to the master node. The routing for the data plane is done separately by a routing agent designed to coordinate with the control agent and begins after an initial bootstrap period when the control routing has stabilized and the routes have been

established in the control plane. The control agents also send data plane route updates to the master node through the control channel, this is done to enable the master node to schedule the corresponding links in the path of a flow. When the control agent detects a flow originating at its node, it sends out a traffic update to the master node. The master node collects all such traffic updates in a given time period and computes a TDMA schedule based on the algorithm that is outlined in [3]. The master node then sends out a TDMA slot assignment to each node in the network containing the transmission schedule for the radio of the corresponding node.

2.4 Security Challenges in Wireless Mesh Networks

WMNs are increasingly becoming popular as a means of achieving broadband ubiquitous access to the Internet. However, there are several issues that concern WMNs which need to be addressed in order to realize their commercial wide scale deployment. Security is one such major area of concern. Salem and Hubaux discuss some challenges to the security of WMNs in [1]. They lay stress on three fundamental operations that must be employed to make the WMN secure, these are (i) detection of corrupt TAPs (ii) securing the routing mechanism and (iii) definition of a proper fairness metric to ensure a certain level of fairness in the WMN. While Salem and Hubaux discuss the significance of these operations in isolation and also operations (i) and (ii) in conjunction, they fail to address the implications of considering operations (i) and (iii) together. In other words they do not address the effect corrupt nodes might have on fairness in the mesh network.

In [4], Zhang, Luo and Lu dedicate a chapter to security in wireless mesh networks. They discuss the security challenges that exist at the different layers of the network stack and also outline both existing and new security technologies that can be used to tackle them. The authors emphasize that, while a lot of the security issues addressed in adhoc networks are relevant in the context of wireless mesh networks, the nature of the network architecture and deployment scenario introduce new security issues in wireless mesh networks. This difference makes solutions proposed for adhoc networks inadequate in the context of wireless mesh networks. While discussing the security challenges at the MAC layer they consider the impact of node misbehavior on the 802.11 MAC protocol. They list node compromise and misbehavior as a major security challenge in wireless mesh networks due to the open nature of their deployment and the difficulty of detecting and mitigating threats due to misbehavior. Some of the attacks they outline are very similar to the threats discussed in this thesis, however, the focus of this work is on node misbehavior in TDMA based fair scheduling schemes which rely on coordination and information exchange among nodes.

The fair scheduling schemes discussed in Section 2.2 are vulnerable to the threats from node misbehavior discussed in this thesis. However, since these schemes operate based on different fairness guarantees, the scale of the impact varies from scheme to scheme. While the threats may have a global effect in [9] and other schemes which perform scheduling based on global demand information from the network, the effect of these threats is limited to the TAP in IFA [10]. This is because, IFA ensures that all TAPs in the network receive the same share of bandwidth and depends on information exchange in the local neighborhood to perform scheduling. In other words in [9] a misbehaving node (Client/TAP) affects the bandwidth that each and every client receives as a result of the scheduling scheme. In contrast, in [10] a misbehaving node only affects the bandwidth that is received by clients that are (i) associated with the TAP in the case when the misbehaving node is a TAP or (ii) associated with the same TAP as the client that is misbehaving.

2.5 Approaches to Authentication in Wireless Mesh Networks

One of the first steps to secure a network from node misbehavior is admission control. Admission control ensures that only legitimate nodes are allowed access to the network and participate in the protocols present in the network. Admission control can be achieved by means of Authentication and Authorization which ensure identification and accountability of the nodes in the network. However, authentication in wireless networks has a number of challenges due to the open nature of these networks. For example, the security requirements and the level of trust in the network varies based on the deployment environment making it impossible for any one authentication scheme to adequately satisfy all the requirements.

There are several approaches to authentication in wireless mesh networks based on factors like the (i) nature of trust relationships between the nodes, (ii) presence of authentication infrastructure in the network etc. These factors also determine the type of cryptography that can be used to design authentication schemes for wireless mesh networks. Authentication schemes using Symmetric Key Cryptography can be designed when there exists a predefined trust relationship between the nodes in the network. While, Public Key Cryptography based authentication can be used in networks where there is no predefined relationship between the nodes in the network. The authentication can be performed either by dedicated authentication infrastructure like an Authentication, Authorization and Accounting (AAA) server or between the nodes themselves when the presence of such an infrastructure is not feasible or not required. The extensible authentication protocol (EAP) for carrying out client authentication in 802.11i based WLANs is a popular choice with infras-

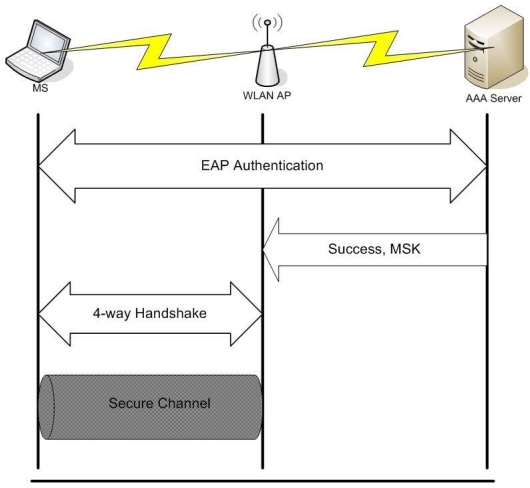


Figure 2.3: EAP based WLAN Access [4]

tructure wireless mesh networks where there exists a clear distinction of trust among nodes(TAPs) which are internal to the network and nodes(Clients) which are external to the network. In such an environment an AAA server which acts as the authentication agent can be used to authenticate the client nodes which join the network. A typical authentication exchange in an EAP based WLAN access is shown in Figure 2.3.

2.5.1 Protocol for Carrying Authentication for Network Access (PANA)

As outlined above, the nature of wireless mesh networks makes it hard for any one specific authentication scheme to be adequate for all the security requirements of the network. Also, since compatibility and integration with other existing wireless and wireline networks is one of the most attractive features of a wireless mesh network, authentication schemes developed for wireless mesh networks have to be compatible and operational with already existing authentication infrastructure. As a result, there is a need to develop a common framework for authentication for network access. The IETF has already formed a working group and has started work on a draft for a Protocol for Carrying Authentication for Network Access (PANA). PANA allows clients to get authenticated to the network using the existing authentication infrastructure in the network without necessarily being aware of the protocol used by this infrastructure. PANA uses EAP as the back-end protocol for carrying out authentication between the client and the authentication infrastructure, since EAP provides flexibility in the choice of authentication and key establishment methods that can be used.

As a result, PANA provides flexibility in terms of the authentication methods used and the nature of trust established in the network. As part of the PANA draft, the working group defines the following entities, (i) PANA Client (PAC), (ii) Enforcement Point (EP) and (iii) PANA Authentication Agent (PAA), and outlines the different deployment and authentication scenarios where PANA is applicable. An authentication session between the PAC and the PAA using PANA is shown in Figure 2.4.

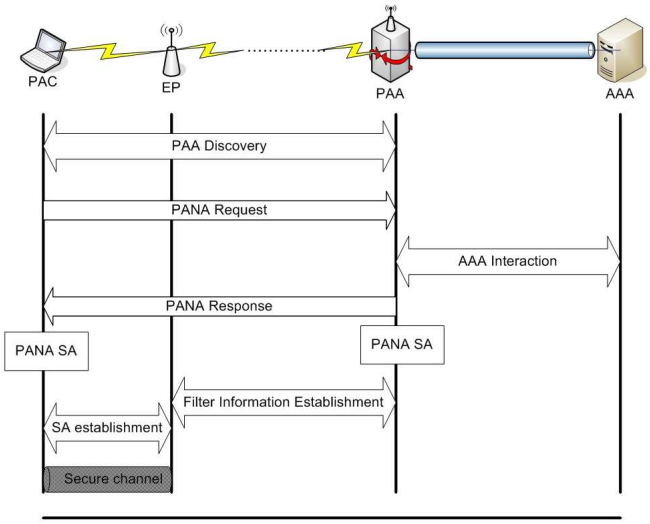


Figure 2.4: PANA Exchange in a WMN [4]

PANA can be used in infrastructure wireless mesh networks to enable client authentication to the network, where each node in the wireless mesh network takes on the role of one of the entities defined in the PANA framework. In a typical authentication session in an infrastructure wireless mesh network, the client node assumes the role of the PAC, the Access point is the EP and the Gateway is the PAA. The PAA may interface with a back-end AAA server for authenticating clients depending on the requirement and presence of such infrastructure. While the EP and PAA are two distinct entities in the authentication example discussed above, it might also be possible that the EP and PAA are collocated on the same device. Such a scenario is possible when public key cryptography is used to achieve authentication and both the TAPs and the client possess public and private key pairs from a recognized Certification Authority. However, unlike in a Centralized Authentication Environment where the TAPs in the network can trust the authentication performed by a AAA server, in a Distributed Authentication Environment, the TAPs may or may not trust the

authentication performed by other TAPs in the network. This difference in the nature of trust, gives rise to two scenarios which have to be considered when designing solutions to detect misbehavior in the network.

2.6 Cryptographic Operations Benchmark

The solutions discussed in this thesis make use of cryptographic operations to detect misbehavior during fair scheduling. In order to evaluate the performance of these solutions, it is essential to benchmark the time taken to perform cryptographic operations on a standard mesh router. We make use of the benchmarking suite provided by OpenSSL [16], an open-source cryptographic library popularly used in the implementation of cryptosystems. The specification of the system used for this benchmarking process is outlined in Table 2.1, it is assumed that these specifications are reasonable for a state of the art mesh router deployed in a wireless backhaul mesh network. The benchmark for the cryptographic operations is listed in Table 2.2 and symbols representing the size and time complexity of these operations are also listed. These symbols are referred while analyzing the performance of the solutions discussed in Chapters 5 and 6.

Table 2.1: System Specification

Processor type	Intel Core2 Duo T2600
Processor speed	2.16 GHz
System Memory	2.0 GB
Disk space	80 GB

Table 2.2: Cryptographic Benchmark

Operation	Space (Symbol)	Size	Time (Symbol)	Time/Operation
md5	S_{MD5}	16	t_{MD5}	$1.1\mu s$
sha1	S_{SHA1}	16	t_{SHA1}	$1.2\mu s$
hmac	S_{HMAC}	16	t_{HMAC}	$0.8\mu s$
des-cbc	S_{DES}	8	t_{DES}	$0.17\mu s$
RSA sign	$S_{RSA-SIGN}$	1024	$t_{RSA-SIGN}$	$2.9ms$
RSA verify	$S_{RSA-VERIFY}$	1024	$t_{RSA-VERIFY}$	$0.1ms$
ECDSA sign	$S_{ECDSA-SIGN}$	160	$t_{ECDSA-SIGN}$	$0.4ms$
ECDSA verify	$S_{ECDSA-VERIFY}$	160	$t_{ECDSA-VERIFY}$	$1.9ms$

Chapter 3

Assumptions and Threat Model

This chapter provides the assumptions about the network, the threat model, and discusses the different threats to fair scheduling due to node misbehavior. Finally, an illustration of the impact of a certain type of misbehavior on the scheme discussed in [3] is presented through the use of an ns-2 simulation where we demonstrate the severity of these threats and the relative ease with which they can be carried out.

3.1 Assumptions

We make the following assumptions regarding the entities in the mesh network :

1. All the benign nodes in the network are time synchronized.
2. Scheduling is performed periodically in time intervals of duration δT called Scheduling Rounds.
3. TAP-TAP communication and TAP-GW communication are authenticated.
4. The GW is the only trusted entity in the network.
5. There exists an underlying authentication framework that enables client authentication.
6. The TAPs do not trust each other individually.
7. The compromised nodes are in a minority in the network.
8. The extent to which a node misbehaves is significant.

3.2 Threat Model

In our research of the problem we assume the following threat model, extending the Dolev-Yao model [17]

1. The adversary can be rational or malicious.
2. The adversary can eavesdrop, capture, drop, resend, delay, or alter packets.
3. The adversary has access to large but not infinite computational resources.
4. The adversary can introduce, capture, tamper or replicate client nodes and TAPs, but not the wireless Gateway or the back-end authentication infrastructure if any exist.
5. The attacker may be a legitimate (authenticated) user of the network.

3.3 Threat Discussion

A framework of threats to fair scheduling in infrastructure wireless mesh networks is built based on the threat model and assumptions specified above. The threats discussed below look at ways in which compromised nodes can tamper the demand information and cause the network to arrive at an inconsistent schedule. The threats can be classified into three broad categories based on the node that is assumed to misbehave during the scheduling.

3.3.1 TAP misbehavior

1. **TAP reporting non-existent clients.**
   A compromised TAP may create an inconsistent view of demand in the network by advertising a large local demand that does not exist in reality. Since TAPs in the network take into account this information to arrive at a schedule, the compromised TAP may be allocated a share that it does not deserve.

2. **TAP failing to report new clients or clients that have already left the network.**
   A compromised TAP may start by reporting correct local demand information and in time fail to report clients that have left the network thus retaining an unfair share of the bandwidth. In a different case, the TAP may stop reporting new clients that have joined the network. While the former is unfair to the other TAPs in the network, the latter is unfair to the clients connecting to the compromised TAP.

3. **TAPs colluding to distribute inconsistent distribution of demand.**
   Two TAPs could collude in a way such that, the first TAP transfers all its demand information to the second TAP, which may then claim this demand as its own. The first TAP might also forward the client credentials it receives to the second TAP, making the demand information generated by the second TAP appear legitimate.

3.3.2 Client Misbehavior

1. **Sybil Attack (A client masquerading as multiple clients).**
   A malicious client could have the ability to masquerade as multiple clients and connect to the mesh network. As a result a TAP could be fooled into reporting a large demand, while the share that it is allocated will all be consumed by a single client.

2. **A compromised client node getting replicated and connecting from different locations in the network.**
   A compromised client node could get replicated and connect to the mesh network through different TAPs. The distributed nature of the network would make it harder for such replication to be detected. The replicated client would then affect the way in which the transmission schedule is calculated.

3. **Client connecting to one or more TAPs within range.**
   A malicious client with an extended communication range may try and connect to the network through one or more TAPs that are within its communication range. As a result this client would be able to get bandwidth from all the TAPs with which it associates. This would create unfairness in the network under a situation where all clients are expected to get an equal share of the bandwidth.

3.3.3 Client and TAP Collusion

1. **A malicious client connecting to a compromised TAP.**
   An attacker may be able to compromise a TAP in the network and circumvent the local authentication schemes embedded in the TAP. As a result the attacker may be able to associate a large number of unauthorized clients to the mesh network. While this in itself is an undesirable property, these unauthorized clients also get factored in as demand at the compromised

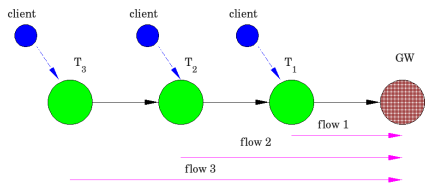
TAP. Hence the mesh network arrives at an inconsistent schedule where, the other legitimate TAPs and clients in the network are penalized.

3.4 Impact of TAP Misbehavior

In this section, we outline the effect of TAP misbehavior on a centralized scheduling scheme CIRMA-MH proposed in [3] through the use of an ns-2 simulation. CIRMA-MH is a centralized scheduling scheme for wireless mesh networks where there exists a central master node that collects bandwidth requests from other nodes in the network. The master node then computes a transmission schedule based on a scheduling algorithm that assigns time slots to the different traffic flows. For the purpose of this illustration, CIRMA-MH has been adapted to model the different levels of hierarchy in a typical wireless backhaul mesh network. In the modified version of CIRMA-MH, there are two types of nodes in the network apart from the master node, the TAPs and the clients. The clients make bandwidth requests to the associated TAP, the TAP aggregates these requests and sends them to the master node. While CIRMA-MH ensures fairness to all the nodes in the network, it operates under the assumption that the per flow bandwidth requests from the nodes are accurate. As a result the scheduler tries to accommodate all the traffic flows that are listed in the aggregated requests from the TAPs without verifying their consistency. It can be observed that in the event of misbehavior, the throughput obtained by all the traffic flows falls drastically since the misbehaving node is allocated a significant number of time slots even though it doesn't have as much demand.

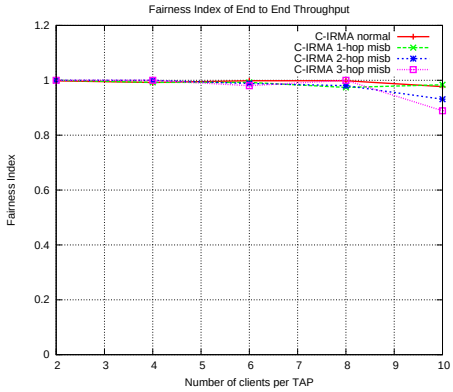
Topology size	1000x1000 m^2
Control Tx Range	250m
Data Tx Range	50 m
Carrier Sense Range	550m
Data Channel Rate	1Mbps
Control Channel Rate	100Kbps
SINR threshold	10dB
Propagation Model	TwoRayGround
Path Loss Index (γ)	4
MAC slot duration	8.4 msec
Slots per Frame	40

(a) Simulation Parameters

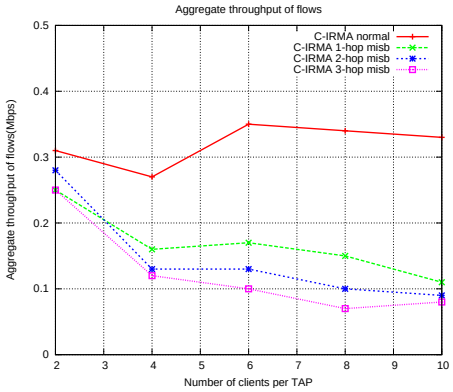


(b) Simulation Topology

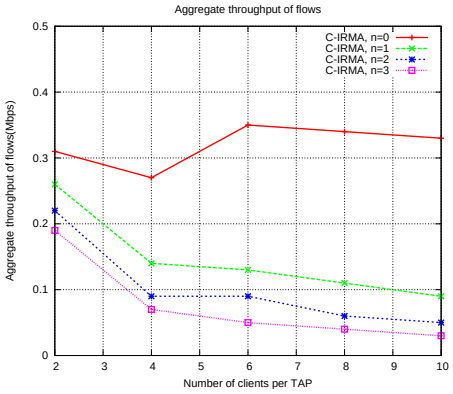
Figure 3.1: Simulation parameters and topology



(a) Fairness Index of End to End Throughput of flows in C-IRMA



(b) Aggregate Throughput of flows in the presence of a single misbehaving node



(c) Aggregate Throughput of flows in the presence of multiple misbehaving nodes

Figure 3.2: Effect of misbehavior on CIRMA-MH

The parameters used in the simulation of CIRMA-MH in [3] are listed in Table 3.1(a) and will be used for this illustration. The simulation is performed for a simple wireless mesh network topology as shown in Figure 3.1(b). The mesh network consists of a Gateway node, three TAPs T_1, T_2 and T_3 and client nodes associated with each of these TAPs. All TAPs are assumed to have the same number of clients in order to simplify the simulation scenario and the simulation is run multiple times with an increasing number of clients associated with each TAP. Three client nodes which are associated with T_1, T_2 and T_3 are selected and generate CBR traffic at the rate of $1Mbps$ towards the gateway node. The TAP misbehavior is implemented in the form of a compromised TAP creating a bandwidth request for every associated client irrespective of the existence of actual demand at these clients.

We first measure Jain's Fairness Index [18] for CIRMA-MH under normal behavior, as well as in the presence of misbehavior as shown in Figure 3.2(a). The Fairness index is defined as $(\sum_{i=1}^{n} x_i)^2/(n\sum_{i=1}^{n} x_i^2)$, where n denotes the number of traffic flows in the network, and x_i the throughput of the ith traffic flow. While it may seem that CIRMA-MH guarantees fairness to all the flows in the network irrespective of the number of hops or the presence of misbehavior, it does so at the cost of reduced throughput. This is due to the allocation of a fixed number of slots to the different flows in the network. The throughput is further reduced drastically in the presence of misbehavior as the central scheduler assigns slots to each of the clients reported by the compromised TAP, and these go waste as the clients have nothing to transmit. This is illustrated in Figure 3.2(b) where, the aggregate throughput of the three flows is measured. It can be observed that in the presence of a single misbehaving node, the aggregate throughput obtained in the network falls by as much as $40-50\%$ and this reduction increases as the number of hops between the misbehaving node and the gateway increases. We also show in Figure 3.2(c) that the throughput received by the clients falls drastically as the number of misbehaving TAPs increases. However, it is interesting to note that a single misbehaving node has the same effect on the aggregate throughput as multiple misbehaving nodes. This shows that an attacker need only compromise a single TAP to have a significant effect on the throughput obtained by the clients of the network.

Chapter 4

Verification Framework

It is clear from the above discussion that fair scheduling in wireless mesh networks is vulnerable to node misbehavior. The scheduling schemes have to be augmented or new schemes must be developed to provide robustness in the presence of misbehaving nodes. A general goal of such a scheme should be to secure the exchange of scheduling related information from misbehaving nodes. This can be achieved by providing a method for verification and detection of inconsistencies in the demand being reported by the TAPs. In this chapter, a generic demand-claim based verification framework is proposed based on which two concrete schemes for verification are designed. The solutions incorporate elements of the demand-claim based verification framework but differ in the architecture of the network where they are applied and the cryptographic primitives which are used to implement the framework.

One of the proposed solutions is for networks where authentication is distributed, in other words, an authentication session involves only the client and the TAP with which it associates. This involves the use of public key cryptography and hence assumes that a public key infrastructure is already in place. Each entity of the network (TAP/client) possesses a public/private key pair obtained from a common Certification Authority (CA) which it uses for generating digital signatures.

The other solution is proposed for networks where the authentication is centralized, in other words, there exists a centralized authentication infrastructure such as an AAA server, which handles the authentication of the clients to the network. Hence it is assumed that authentication and key establishment methods are already in place to establish a secure channel between the client and the AAA server, and also between the client and the TAP. We also assume that secret keys shared between a TAP and the back-end AAA server are pre-distributed during network initialization and can be replenished from time to time.

Table 4.1: Cryptographic Notations

Notation	Description
$[M]_X$	Digital Signature on Message M by entity X
$M_1 \| M_2$	Message M_1 concatenated with Message M_2.
$H(M)$	Hash image of Message M using the one-way hash function H.
$MAC_{k_{XY}}(M)$	Message Authentication code for Message M using secret key k_{XY}.

4.1 Notations and Terminology

We first begin by defining the notations and terms that will be used while describing the schemes. In our discussion of the solutions, we use the notations listed in Table 4.1 to denote the cryptographic operations used in the specification of our solutions, where X,Y denote the communicating parties, H is a one-way hash function, M denotes the Message or payload and k_{XY} denotes the secret key shared between entities X and Y.

The formal definitions of terms that will be used while describing the verification framework are provided below.

- **Client commitment:** A commitment from the client to the TAP that binds the client to the TAP for a particular scheduling round.
- **TAP commitment:** The commitment from the TAP to all its clients certifying the association between the TAP and its associated set of clients for a particular scheduling round.
- **Demand Claim:** A demand claim as the name suggests is information broadcast by a TAP providing the current demand at the TAP. In other words, the demand claim is proof of the demand requirements at a TAP.
- **Reactivation tokens:** The tokens used to reactivate an existing association between a client and a TAP for a particular scheduling round.

4.2 Framework Overview

In an infrastructure wireless mesh network, all the traffic is generated by the clients and the TAPs simply act as intermediate nodes forwarding traffic to the Gateway. In such a network where a fair scheduling scheme is implemented, the TAPs exchanging demand information can only

advertise as much demand as the amount of associated client traffic. Using these characteristics, the intuition behind the verification framework is that, in order for the demand information to be considered authentic, it must include inputs from the TAPs as well as from the clients whose traffic is being forwarded. In other words, the clients, which are the entities that generate traffic and have the actual bandwidth requirement also need to participate in the scheduling process.

The Verification framework consists of three main stages (i) Commitment Generation, (ii) Demand Claim Generation and (iii) Demand Claim Verification. Each of these stages can be further divided into multiple stages based on the scheme that implements this framework. The three stages of the verification framework are outlined below.

4.2.1 Commitment Generation

During the information exchange phase of a fair scheduling scheme, a client in the network generates a verifiable commitment certifying its association with a TAP in the network for a particular scheduling round. A TAP aggregates all such commitments from its set of associated clients and generates its own verifiable commitment agreeing to the association with this associated set of clients. This aggregation alleviates the burden on the TAP from having to commit to each individual association.

4.2.2 Demand Claim Generation

The TAP after having collected the client commitments and generating its own commitment, proceeds to generate a demand claim. The demand claim is a verifiable representation of the current demand at the TAP and is constructed using the client and TAP commitments. The demand claim is then exchanged with other TAPs as part of the scheduling scheme. The demand claim can be used both to estimate and authenticate the demand at a TAP.

4.2.3 Demand Claim Verification

As part of a fair scheduling scheme, a TAP on receiving demand information from other TAPs in the network, proceeds to use this demand to compute a schedule for transmission. However, before this information can be used to perform scheduling, the verification framework requires that the demand claim be verified. Since the nature of the demand claim makes it verifiable by any other client or TAP in the network, the TAPs using this demand information proceed to verify the claim and generate an alert if any inconsistencies are found during verification. The nodes in the network

may then choose to mitigate this misbehavior in a number of different ways which are beyond the scope of this work.

The Framework is inspired from the approach adopted in [19] where Parno, Perrig and Gligor propose a distributed detection mechanism to detect replicated sensor nodes in a wireless sensor network. The scheme proposed in [19] makes use of the concept of emergent properties proposed by Gligor in [20]. Emergent properties are properties that arise out of the collective action of multiple nodes. In the context of the verification framework proposed in this work, the properties emerge out of the actions of both the clients and the TAPs which generate commitments that constitute a demand claim. The use of emergent properties makes it difficult for any one node participating in the scheduling to misbehave and escape detection. It also ensures that any demand information exchanged as part of the scheduling process is generated using inputs from all the entities concerned as a result, making it harder for a node to misbehave in the first place.

4.3 Security Properties

This section outlines the security properties that must be satisfied by a scheme implementing the verification framework. These properties represent the conditions violated by a misbehaving node for each of the threats discussed in Chapter 3.

4.3.1 Client Authentication

Client authentication is the first step towards ensuring that the exchanged demand information is based on estimates from legitimate nodes. It precludes external adversaries from taking part in the fair scheduling scheme without being detected. An infrastructure wireless mesh network serving clients, needs to have authentication mechanisms in place to ensure that only legitimate clients can gain access to the network. Since the nature of trust in the network may vary depending on the type of deployment, the definition of a legitimate client also varies depending on the type of deployment. For example, in a network environment where the users and the network operator have a predefined contract for usage of services, a legitimate client is a client which has successfully negotiated such a contract and has the credentials to prove its validity. However, in an environment where there exists no predefined relationship between a client and the network operator, a legitimate client is a client which can prove its identity.

In order for this property to be satisfied, the verification framework works under the as-

sumption that there is an underlying authentication infrastructure in place, and all clients entering the network have to authenticate themselves in order to use the network resources. Work is already in progress on the PANA (Protocol for carrying Authentication for Network Access) framework which is a protocol aimed at enabling flexible authentication in multihop networks. The PANA framework specifies three different entities, each of which directly correspond to the entities that participate in the verification framework namely the (i) client (PANA Client), (ii) TAP (Enforcement Point/PANA Authentication Agent) and (iii) Gateway (PANA Authentication Agent).

4.3.2 Commitment Unforgeability

The verification framework is proposed on the basis that all nodes in the network participate in the scheduling process to ensure robustness of the scheme against misbehaving nodes. This is achieved by the use of commitments which authenticate the association between a client and a TAP. Since both the client and TAP commitments are needed to authenticate an association, commitment unforgeability is a critical requirement of any scheme implementing the verification framework. This property ensures that any subset of compromised nodes cannot successfully forge a legitimate demand claim for a particular TAP and escape detection. One way to achieve commitment unforgeability is by constructing commitments using information that is private to the corresponding nodes. In the case of public key cryptography, commitments can be constructed as digital signatures using the node's private key, while in the case of symmetric key cryptography, commitments can be constructed as Message Authentication Codes (MAC) using a shared secret generated between the node and a Trusted Third Party (TTP).

4.3.3 Claim Verifiability

In a Fair scheduling scheme, TAPs in the network generate bandwidth requests to obtain timeslots for transmission. The bandwidth requests are sent to a central scheduling node in the case of centralized scheduling or exchanged among nodes in the local neighborhood in the case of distributed scheduling. It can be observed that, the accuracy of the demand information thus exchanged is critical to the correctness and efficiency of any fair scheduling scheme (centralized or distributed). The verification framework requires that the bandwidth requests generated by TAPs be verifiable in order to detect inconsistencies resulting from misbehavior. This is achieved by constructing bandwidth requests as demand claims where a claim is a verifiable representation of demand at a particular TAP. The demand claim is constructed using both the client and TAP com-

mitments, and is updated as the local demand at a TAP changes due to clients joining or leaving the network.

The extent of verification of a demand claim may vary depending on the fair scheduling scheme. For example, while in [9] the demand request is propagated to all the nodes in the network, in [10], the demand request is propagated upstream to all the TAPs in the path to the Gateway. Since the verification framework is designed to be integrable with any underlying fair scheduling scheme that employs exchange of information, the demand claim generated by the framework must be compatible with any extent of verification. As a result, the verification framework requires that the demand claim be publicly verifiable as this precludes the dependence on any one type of fair scheduling scheme.

4.4 Performance Metrics

The performance metrics used to analyze the verification framework can be divided based on two criteria (i) Computation Overhead and (ii) Communication Overhead. We use delay (time) and percentage control overhead as the measuring parameters in computing the overhead incurred by the verification framework.

4.4.1 Computation Overhead

- Commitment Construction Delay ($\tau_{cmtConst}$) :
  The commitment construction delay is the time taken by a client to construct a commitment. As the verification framework defines the generic structure of a commitment,$\tau_{cmtConst}$ varies for the different implementations of the verification framework proposed in this thesis and depends on the cryptographic primitives used to generate the client commitment.

- Claim Construction Delay ($\tau_{claimConst}$) :
  The claim construction delay is the time taken by a TAP to construct a demand claim. As with $\tau_{cmtConst}$, $\tau_{claimConst}$ also depends on the cryptographic primitives used to construct the demand claim while also depending on the structure of the demand claim.

- Commitment Verification Delay (τ_{cmtVer}) :
  The commitment verification delay is the time taken by a TAP to verify a commitment and is dependent on the cryptographic primitives used to construct the client commitment.

- Claim Verification Delay ($\tau_{claimVer}$) :
  The claim verification delay is the time taken by a TAP to verify the demand claim from another TAP. $\tau_{claimVer}$ depends on the cryptographic primitives used to generate the claim and also on the structure of the demand claim. Apart from this, the claim verification delay is also directly dependent upon the number of active clients associated with the TAP making the claim.

4.4.2 Communication Overhead

- Commitment Transmission Delay ($\tau_{cmtTrans}$) :
  The commitment transmission delay is the time taken for a client commitment to reach the corresponding TAP. The commitment transmission delay is affected by factors such as the structure of the commitment, which is determined by the cryptographic primitives being used. It is also affected by the amount of interference in the mesh network, which is determined by the number of clients with active flows.

- Claim Transmission Delay ($\tau_{claimTrans}$) :
  The claim transmission delay is the time to transmit a demand claim. This may vary depending on the underlying scheduling protocol that is augmented by the verification framework. For example, in a centralized scheduling algorithm like CIRMA-MH, the claim transmission delay is the time taken for the claim to reach the centralized scheduling node. It is also subject to different factors like the structure of the claim, the amount of interference in the mesh network and the distribution of the traffic flows in the mesh network.

- Signaling Overhead (δo):
  The Signaling overhead is the amount of extra control data that has to be exchanged in order to detect misbehavior in the network. The signaling overhead can be roughly estimated in terms of the number of additional messages sent as part of the verification framework. However, as the verification framework works with the scheduling scheme, the signaling overhead varies with the the extent of integration between the scheduling scheme and the verification framework. While in a tightly coupled setting where the commitments and the demand claim are piggybacked on the control messages of the scheduling protocol, the communication complexity of the verification framework can be very low, in a loosely coupled setting where explicit control messages have to be sent to transmit verification data, the communication complexity can be significant, again, this is subject to the underlying scheduling scheme.

Since the verification framework is designed to extend any existing fair scheduling scheme that relies on exchange of demand information, it is reasonable to analyze the performance of the verification framework with respect to the fair scheduling scheme that it extends. For this purpose, we extend the ns-2 implementation of CIRMA-MH discussed in [3] to analyze the communication overhead incurred by the verification framework and use the OpenSSL benchmark listed in Table 2.2 to analyze the computation overhead. Also, while analyzing the computation overhead, we only take into account the overhead of the cryptographic operations involved as it is reasonable to assume that the overhead from these operations is very significant compared to the other operations involved.

4.5 Overview of Solutions

We now proceed to discuss the proposed solutions in detail, providing the basis for detecting misbehavior during fair scheduling in a wireless backhaul mesh network. The solutions are classified into different chapters based on the nature of the authentication in the network and the type of cryptography used to generate and verify the commitments and the claim. Chapter 5 discusses a public key based verification scheme that is designed for an environment where the clients are authenticated by the TAPs with which they associate. This type of authentication is referred to as distributed authentication. In such an environment, other TAPs in the network cannot implicitly trust the authentication process since the TAPs are themselves vulnerable to compromise. As a result, public key cryptography is used to provide universal verifiability of the demand claim and the commitments. Chapter 6 outlines a scheme based on symmetric key cryptography and Secure Hash based sequential Aggregate and Forward secure Signatures (S-HAFS) [11]. This scheme is designed for a network environment where there exists a dedicated authentication infrastructure like an AAA server that handles the authentication of clients to the network and this type of authentication is referred to as Centralized Authentication.

The scheme in Chapter 5 is presented in two parts. First, we consider a scenario where the TAPs in the network are trusted and the clients are expected to misbehave and propose a basic variant of the signature based verification scheme. We then discuss the purpose of such a scheme and the advantages it offers in terms of security and performance. Second, we consider a scenario in which both the TAPs and the clients in the network are expected to misbehave. We highlight the inadequacies of the basic variant and propose a more complex variant of the verification scheme. Finally, we observe that the complex variant incurs a significant amount of overhead and propose a

solution to improve the performance while preserving the security guarantees of the scheme.

The scheme in Chapter 6 is presented with a single assumption that both the clients and TAPs in the network are expected to misbehave. We briefly discuss the S-HAFS scheme proposed in [11] and highlight the modifications to the scheme in order to suite the requirements of the verification framework. We then present each stage of the HAFS based verification scheme in detail, and analyze its security and performance with respect to the properties and metrics defined in the verification framework.

Chapter 5

Misbehavior Resistant Fair Scheduling in a Distributed Authentication Environment

In this chapter we propose a realization of the verification framework for a network environment where the authentication of clients is performed by the TAPs with which they associate. This architecture is a realization of the PANA framework when the PANA Authentication Agent (PAA) and the Enforcement Point (EP) are collocated on the same node. Such an environment is possible when there exists no predefined relationship between the clients and the network, and the presence of a centralized authentication infrastructure is not feasible or required. The distributed nature of authentication implies that TAPs in the network cannot trust the authentication process or any symmetric key based credentials that might be generated during such a process. This necessitates the use of public key cryptography for credential generation in order to achieve public verifiability.

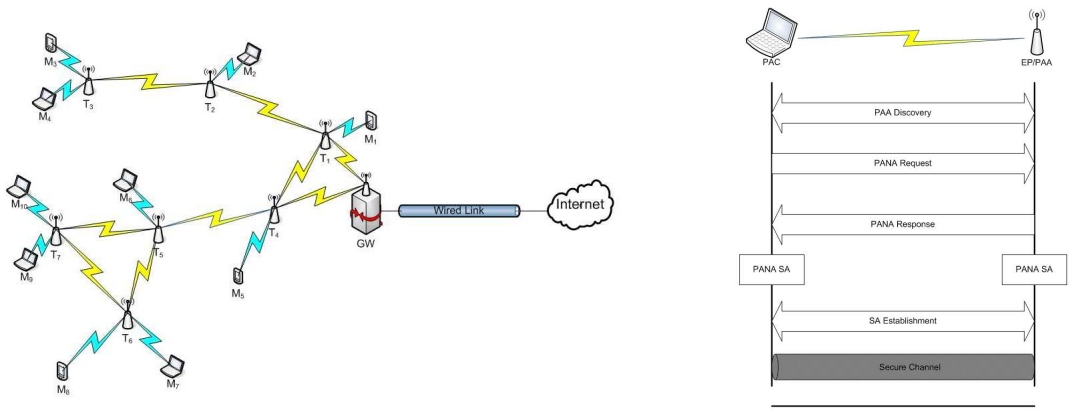
5.1 Overview

First, we provide a brief outline of the network architecture for which such schemes are proposed. We then proceed to describe a basic variant of the verification framework when it is assumed that all TAPs in the network are trusted to behave properly. We suggest several enhancements to the basic variant to mitigate specific threats that are outlined in Chapter 3 and analyze the

security and performance of this scheme. We then relax the assumption made for the basic variant and propose a more complex variant of the verification framework which ensures detection of both client and TAP misbehavior. We analyze the security and performance of the complex variant and propose a probabilistic selective verification scheme that alleviates the computational burden on the TAPs performing claim verification. The selective verification scheme reduces the redundancy in claim verification while ensuring a high probability of detecting misbehavior. As a result, while a node may get away with a small amount of misbehavior, the scheme ensures that the probability of detection increases rapidly as the extent of misbehavior increases.

5.2 Network and Authentication Architecture

The network architecture for a mesh network with a distributed authentication environment is shown in Figure 5.1(a). The mesh network consists of TAPs $T_{1...7}$ and a Gateway GW which form the stationary backbone. The client nodes $M_{1...10}$ connect to the mesh network by associating with the closest TAP. Clients authenticate themselves to the TAP through an underlying authentication framework like PANA which is assumed to exist in the network. A typical PANA authentication exchange between the client and the TAP is shown in Figure 5.1(b).



(a) Mesh network architecture

(b) PANA Authentication exchange

Figure 5.1: Distributed Authentication Environment

This type of mesh network architecture is possible for a public area wireless mesh network (PAWN) like the ones deployed in airports and shopping centers. These mesh networks offers network services for free or at subsidized rates to all customers without the requirement of having

a predefined contract. As a result, authentication is essentially needed for identification, hence the client is granted access to the network as long as it is able to prove its identity.

5.3 Basic Variant : Dealing with Client Misbehavior

Client misbehavior manifests essentially in two major forms (i) client replication and (ii) client impersonation. Given the assumption that the TAPs in the network are trusted, the above misbehavior can be mitigated by a basic variant of the signature based verification scheme. The three stages of the verification framework for the basic variant of the scheme are outlined below.

5.3.1 Commitment Generation

Clients entering the mesh network first identify the TAP with which they wish to associate. This is usually done by means of beaconing where the TAPs periodically broadcast their presence for the benefit of clients that are new to the network, the clients then identify the TAP with the strongest signal and initiate the mutual authentication and association session. At the beginning of next scheduling round, the client generates a commitment which binds its own ID with the ID of the TAP with which it associates and sends it to the TAP. The TAP aggregates these client commitments and then generates a single commitment common for all clients associated with it and then sends it back to the clients. At the end of this stage, the TAP has a set of client IDs and its own commitment to the association with this set of clients, it then proceeds to construct the demand claim. The specification of the commitment generation phase is outlined below.

Let M_i denote the identity of client i such that $\forall i \neq j, M_i \neq M_j$, T_j denotes the identity of TAP j with which it associates and $Comm_{M_i}$ denote the client commitment to the TAP. At the beginning of the scheduling round, the client generates a commitment $Comm_{M_i}$ and sends it to the TAP, where $t_{current}$ is the start time of the current scheduling round.

$$M_i \rightarrow T_j : Comm_{M_i} \quad \text{where } Comm_{M_i} = [M_i||T_j||t_{current}]_{M_i}$$

TAP T_j on receiving the commitment verifies the ID M_i of the client and the association specified in the commitment. Once the TAP has received and verified commitments from all the clients that wish to associate with it, it generates a common commitment $Comm_{T_j}$ and sends this commitment to its clients, where $Comm_{T_j} = [H(M_i||.....||M_j)||t_{current}]_{T_j}$.

$$T_j \rightarrow M_{i...j} : M_i, ..., M_j, Comm_{T_j} \quad \forall M_i, .., M_j \text{ associated with TAP } T_j$$

This common commitment $Comm_{T_j}$ amortizes the cost of signature generation on the TAP since it no longer has to generate a signature for each of the associated clients. The TAP then proceeds to generate a demand claim representing the local demand information.

5.3.2 Demand Claim Generation

A TAP receiving demand information from another TAP must have a means to verify the authenticity and consistency of the information reported in the claim. This requires that the demand information generated by a TAP as part of scheduling has to be verifiable. After aggregating the commitments received from the clients and generating its commitment, a TAP proceeds to generate the demand claim for the particular scheduling round as shown below.

$$T_j \rightarrow T_{1..j-1,j+1,...K} : M_i, ..., M_j, Comm_{T_j}$$

The claim is made up of the client IDs, the TAP commitment and other relevant information. In other words, the claim is a verifiable account of all clients that are associated with a particular TAP. This information can be used to detect clients that have been compromised and replicated in the network or clients that appear to be misbehaving in other ways as described in Chapter 3. The TAP then broadcasts this demand claim to other TAPs in the network for verification.

5.3.3 Demand Claim Verification

A TAP upon receiving a claim first aggregates the client IDs and verifies the TAP commitment, this ensures that the TAP has committed to this set of clients. The TAP then checks its local list of client IDs for any collision with the client IDs present in the claim. In other words, it verifies that the clients included in the claim are not replicated within the set of clients associated with itself. In the event that a collision is detected, the TAP may trigger node revocation and network reconfiguration mechanisms.

5.3.4 Cross Verification of Claims

While the above scheme is robust against replicated clients associated with uncompromised TAPs, the scheme is vulnerable to a particular form of client replication arising out of client-TAP collusion where, a replicated malicious client associates itself with several compromised TAPs.

As a result, the compromised TAPs which are fully aware of this replication may choose to ignore it, while an uncorrupted TAP may fail to detect this replication as it only verifies for collisions between the clients included in the received claim and its associated set of clients. Hence a cross verification mechanism is proposed where the TAP that receives and verifies claims, keeps track of all the clients that have been seen so far in the claims in a form that can be easily verified for collisions. A simple way of achieving this is for the TAP to store all the client IDs it has received and verified so far in a hash table and to verify this hash table for collisions whenever it receives a demand claim from any TAP in the network. This effectively precludes the threat of a replicated client escaping detection by only associating with compromised TAPs.

Also, while the cross verification of the demand claims ensures that a replicated client that associates only with compromised TAPs is detected, it still assumes that all TAPs in the network share a set of common neighbors that can detect this replication. As a result, for large scale wireless mesh networks that employ distributed scheduling where demand claims are exchanged only in the local neighborhood, a replicated client connecting to two TAPs that have no common neighbors can still escape detection even in the presence of cross verification. We propose two different assumptions about the propagation of the demand claim in the network and either of them can be used to detect the threat outlined above.

The first assumption is that, the TAPs send the verifiable demand claim to the Gateway (GW) which is centrally located in the network while using simple bandwidth requests in the local neighborhood to perform scheduling. The GW is expected to verify every demand claim that is sent during a scheduling round and detect and respond to misbehavior. Since the GW receives the demand claim from every TAP in the network it has information about all the clients that are present in the network for a particular scheduling round and can detect the threat of client replication outlined above. However, this assumption imposes an enormous amount of computation overhead on the GW as it has to verify a significant number of demand claims when the size of the network is very large and this overhead is further amplified in the complex variant that is discussed later as the GW cannot perform selective verification (also discussed later) since it is the only entity that verifies the claim for inconsistencies.

The second assumption is that, the demand claim from a TAP is not just broadcast in the local neighborhood as part of scheduling, but is flooded in the entire network such that it reaches every other TAP in the network. The design of the secure flooding scheme that is used to flood the demand claims is out of the scope of this research and is assumed to exist. As a result of this assumption, for a particular scheduling round, a TAP is assumed to receive a demand claim

from every other TAP in the network and can cross verify these claims to detect the type of client replication that is outlined above. While the advantage of this assumption is that it breaks down the computation involved in claim verification over multiple TAPs and enables the use of selective verification for the complex variant discussed later to further reduce the computation burden, it does incur a significant amount of communication overhead since the network is flooded by a large number of claims when the network size is large and this happens for every scheduling round.

5.3.5 Reactivating the Client Associations

A major drawback of the basic variant is that, a TAP may not always have a consistent view of the demand even in the presence of well behaved clients. In a scenario where clients constantly join and leave the network, the TAP does not have a mechanism by which it can detect the current local demand at a given instant of time except by examining the client IDs associated with it. As a result, the TAP does not know if a client has left the network unless it has specifically been notified by the client, however, this might not be the case in most WMNs. A client may simply choose to power down or terminate its connection and leave, in which case the TAP is left with a view of non-existent demand.

One way to remedy this problem would be for the TAP to expect a fresh commitment from the client at the beginning of every scheduling round. Hence, at the beginning of every scheduling round, the TAP receives a new set of client commitments based on which it constructs its own commitment and proceeds to construct the demand claim. This scheme also forces the clients to re-authenticate themselves periodically which proves a counter against threats like session hijacking.

However, it can be observed that every time a client renews its commitment, it has to generate a new signature on the same commitment but with a different timestamp. This is quite cumbersome on a client with low computational abilities, and can prevent a client from ever re-associating with a TAP if it is not computationally fast enough to renew its commitment within the interval of a scheduling round. In order to mitigate this problem, a hash chain mechanism is used to achieve re-authentication. In the new scheme, the client M_i includes a token $K^0_{M_i}$ in the initial client commitment generated when it first associates with a TAP T_j. $K^0_{M_i}$ serves as the commitment to the values in the hash chain that will be used for re-authentication. The values $K^j_{M_i}$ for all $j = 1...n$ are called reactivation tokens. The commitment sent by a client M_i that has been associated with the TAP T_j for l scheduling rounds is shown below, where $t^1_{M_i}$ is the start time of the first scheduling round after M_i has associated with T_j and $t^l_{M_i}$ is the start time of the lth scheduling round after

M_i has associated with T_j and is the same as the start time of the current scheduling round in the network $t_{current}$ ($t^l_{M_i}$ = $t_{current}$).

$$M_i \rightarrow T_j : Comm^l_{M_i} \quad \text{where } Comm^l_{M_i} =< M_i, K^l_{M_i}, l, t^l_{M_i}, [M_i||T_j||K^0_{M_i}||t^1_{M_i}]_{M_i} >$$

and

$$K^0_{M_i} \xleftarrow{H} K^1_{M_i} \xleftarrow{H} K^2_{M_i} \xleftarrow{H} K^3_{M_i} \xleftarrow{H} ..... \xleftarrow{H} K^{L-1}_{M_i}$$

5.3.6 Caching Reactivation tokens

While the use of reactivation tokens alleviates the computation burden on the client, the client still has to transmit the signature as part of the commitment and the TAP still has to perform a signature verification in order to verify the client commitment. One possible way to reduce both the communication overhead of the commitment transmission as well as the computation overhead of commitment verification is by having the TAP cache the reactivation tokens received at the beginning of the scheduling round from the clients. As a result, the TAP only needs to perform one signature verification operation for the first time it receives a commitment from the client and can verify the authenticity of all the subsequent commitments from the client by performing hash operations. However, this does introduce some overhead in terms of storage space required at the TAP to cache the client commitments. Also, in order to prevent the TAP from having to cache the client commitments indefinitely, the clients need to explicitly inform the TAP about the lifetime L of their association with the TAP, in other words, the number of scheduling rounds that this association will persist. As a result, the client M_i sends the following commitment for the first scheduling round after it has associated with TAP T_j as shown below.

$$M_i \rightarrow T_j : Comm^0_{M_i} \quad \text{where } Comm^0_{M_i} =< M_i, K^0_{M_i}, L, t^1_{M_i}, [M_i||T_j||K^0_{M_i}||t^1_{M_i}]_{M_i} >$$

And for any l^{th} ($l < L$) scheduling round after that, the client sends the following commitment shown below.

$$M_i \rightarrow T_j : Comm^l_{M_i} \quad \text{where } Comm^l_{M_i} =< M_i, K^l_{M_i} >$$

The TAP T_j can then verify this commitment by performing a hash operation on $K^l_{M_i}$ and verify if its equal to $K^{l-1}_{M_i}$ that was cached from the previous round. This performance enhancement has to be slightly modified for the the complex variant of the Signature based scheme described

later, as the TAP also has to cache the signature from the original client commitment and use it to construct its demand claim for every scheduling round. The complex variant of the signature based scheme will be discussed without this enhancement.

5.3.7 Security Analysis

In this section, we analyze the security of the basic variant with respect to the security properties that are outlined in the verification framework.

- **Client Authentication**
  Client authentication is assumed to be done via the underlying PANA framework. The clients (PAC) entering the network initiate a PANA authentication session with the TAP (EP/PAA) and exchange credentials to prove their identity. In a distributed authentication environment, this is achieved by the use of public key cryptography by using digital signatures. As a matter of fact, the initial commitment generation phase of the signature based verification scheme can also be considered as part of the mutual authentication session between the TAP and the client. Once the client has successfully authenticated itself to the TAP, the TAP includes the client ID in its list of associated clients. Client authentication ensures that a TAP will not include a client in its demand request unless it has been authenticated.

- **Commitment Unforgeability**
  As observed from the basic variant, the clients send a commitment to the TAP at the beginning of each scheduling round. The commitment involves a signed request for association with the TAP, the reactivation token for the current scheduling round and the number of scheduling rounds that the client has been associated with the TAP. The TAP on receiving these client commitments generates its own commitment to the association with the set of clients. Both the TAP and client commitments meet the requirement of unforgeability since they are generated using digital signatures. The use of reactivation tokens reduces the burden of commitment verification while still preserving the property of commitment unforgeability due to the one way property of the hash function that is used to generate the reactivation tokens.

- **Claim Verifiability**
  Since the TAPs in the network are trusted, the demand claim needs to be verified only for client replication. The structure of the demand claim makes this verification possible as it is generated using the client IDs and the signed commitment from the TAP. A TAP receiving a

demand claim verifies the commitment from the TAP to verify the integrity of the claim. It then proceeds to check for any collisions between the clients listed in the claim and the local set of associated clients. As a result, the basic variant satisfies the property of claim verifiability and precludes the threat of client replication. The cross verification of demand claims from different TAPs further boosts the security of the verification scheme as it precludes the threat of malicious clients associating with compromised TAPs.

5.3.8 Performance Analysis

In this section, we analyze the performance of the basic variant with respect to the performance metrics outlined in the verification framework. We first proceed to analyze the performance metrics pertaining to the computation overhead of the verification framework.

- **Commitment Construction Delay**
  The construction of the commitment involves generating a signature on the client-TAP association and the reactivation token. This is the only significant cryptographic operation involved as we do not consider the overhead incurred from generating the reactivation tokens since this operation is performed only once during initialization. As a result, the commitment construction delay, $\tau_{cmtConst} = t_{SIGN}$, where t_{SIGN} is the time taken to generate a digital signature.

- **Claim Construction Delay**
  The most significant operation in claim construction is generating the TAP commitment from the client commitments. This involves performing a hash operation to aggregate the client IDs and generating a signature on this aggregation. The claim construction delay, $\tau_{claimConst} = t_{MD5} + t_{SIGN}$, where t_{MD5} is the time taken to perform an md5 hash operation.

- **Commitment Verification Delay**
  In order to verify a commitment, a TAP has to first hash the reactivation token l times for the number of sessions the client has been associated with the TAP. The TAP then has to verify the signature that is included in the commitment to verify the authenticity of the commitment. The commitment verification delay, $\tau_{cmtVer} = l * t_{MD5} + t_{VERIFY}$, where l is the number of scheduling rounds over which the client has been associated with a TAP and t_{VERIFY} is the time taken to verify a digital signature.

- **Claim Verification Delay**
  The claim verification delay involves, performing a hash operation to aggregate the client

IDs included in the claim and verify the signature on the TAP commitment included in the claim in order to verify its authenticity. The TAP then searches for a collision between the clients listed in the claim and its associated clients, this operation can be performed in $O(1)$ time based on the way the client IDs are stored in the TAP. The claim verification delay, $\tau_{claimVer} = t_{MD5} + t_{VERIFY} + O(1)$.

We now proceed to discuss the performance metrics pertaining to the communication overhead of the verification framework.

- **Commitment Transmission Delay**
  Commitment transmission in the basic variant involves transmission of the signed association between the client and the TAP along with the reactivation token for the current scheduling round. As a result, a typical $\tau_{cmtTrans}$ is close to the order of delay in transmitting a signature.

- **Claim Transmission Delay**
  Since the basic variant involves transmission of the client IDs and the signature, the claim is almost identical in size to the commitment. As a result the $\tau_{claimTrans}$ is of the same order as the $\tau_{cmtTrans}$.

- **Signaling Overhead**
  The signaling overhead of the basic variant occurs due to the commitment transmission and the claim transmission. The signaling overhead per scheduling round in a typical mesh network with k TAPs and n clients with an active traffic session per TAP can be roughly estimated as $(n+1) * k * S_{SIGN} + n * k * S_{MD5}$ where S_{SIGN} is the Size of a digital signature and the S_{MD5} is the size of an md5 hash.

A more concrete analysis of the communication overhead is discussed in Chapter 7 based on experimental evaluation of CIRMA-MH augmented with the basic variant. The results from this experimental evaluation of the basic variant are discussed along with the results from the evaluation of the other solutions proposed in this thesis.

5.4 Complex Variant : Dealing with TAP Misbehavior

In the basic variant proposed above, we assume that the TAPs in the network are trusted and the clients are the only misbehaving nodes. While this assumption is valid, it is not practical, as

the deployment environments of a WMN may not always ensure that the TAPs in the network are beyond compromise. As the network infrastructure used is cheap and the deployment environment is in most cases a public environment, TAPs can also be active targets of an attacker looking to gain entry and compromise the network. In such a scenario, the above variant of the verification scheme proves inadequate, and requires certain enhancements to make it more robust to TAP misbehavior.

The basic variant of the verification framework ensures that the demand information being reported is consistent i.e. it succeeds in detecting any attempts by a malicious client to manipulate the demand information. However, if the TAPs in the network are no longer impervious to compromise, it becomes necessary to verify the authenticity of the claim apart from its consistency. As a result, the claim should be structured in a way such that it is resistant to manipulation by a compromised TAP. A more complex variant of the verification framework that incorporates this requirement is outlined below.

5.4.1 Commitment Generation

The client-TAP mutual commitment generation phase remains the same as in the basic variant, save for some minor changes. Following the same notations as above, a client M_i that has been associated with TAP T_j for l sessions generates a client commitment $Comm^l_{M_i}$ during the beginning of the scheduling round as shown below, where $t^1_{M_i}$ is the start time of the first scheduling round after M_i has associated with T_j and $t^l_{M_i}$ is the start time of the lth scheduling round after M_i has associated with T_j and is the same as the start time of the current scheduling round in the network $t_{current}$ ($t^l_{M_i}$ = $t_{current}$).

$$M_i \rightarrow T_j : Comm^l_{M_i} \quad \text{where } Comm^l_{M_i} =< M_i, K^l_{M_i}, l, t^l_{M_i}, [M_i||T_j||K^0_{M_i}||t^1_{M_i}]_{M_i} >$$

The TAP on receiving this commitment verifies the signature in the commitment and the association that is requested. The TAP at this point generates its commitment $Comm_{T_j}$ using the client IDs as in the basic variant. The TAP then proceeds to generate the demand claim.

$$Comm_{T_j} = [H(M_i||....||M_j)||t_{current}]_{T_j}$$

5.4.2 Demand Claim Generation

In the basic variant of the scheme, the structure of the claim included the IDs of the clients associated with the TAP, this structure proves sufficient when the TAPs in the network are

trusted. However, under the assumption that the TAPs are capable of misbehavior, the claim has to be made robust against manipulations by a compromised TAP. As discussed in Chapter 3, TAP misbehavior manifests in the form of a TAP misreporting or manipulating local demand information in the claim. In order to make the claim resistant to manipulation, it is restructured to include the (i) client commitments and (ii) the TAP commitment. This claim is then broadcast to all other TAPs in the network.

$$T_j \rightarrow T_{1..j-1,j+1,...K} : Comm_{M_i}^{l_i}, ..., Comm_{M_j}^{l_j}, Comm_{T_j}$$

5.4.3 Demand Claim Verification

A TAP upon receiving this claim performs the following operations, (i) verifies the signature on the TAP commitment, (ii) checks the claim for client replication and (iii) verifies the signatures on each of the client commitments included in the claim. In other words, the process can be described as follows, the TAP's commitment to the clients listed in the demand claim is first verified. Once the TAP's commitment has been confirmed, the client commitments are scanned to detect collisions as this would indicate a possible replication. Then, the existence of the clients is verified by verifying the client commitments.

5.4.4 Security Analysis

In this section, we analyze the security of the complex variant with respect to the security properties outlined in the verification framework.

- **Client Authentication**
  The complex variant ensures client authentication, since it operates on the same assumption of an underlying authentication framework as the basic variant.

- **Commitment Unforgeability**
  Commitment unforgeability becomes much more critical in the context of the complex variant as the TAPs are no longer trusted to behave properly. As a result, it is important to ensure that a compromised TAP in the network cannot forge client commitments and evade detection. The structure of the client commitments uniquely authenticates the client's agreement to the association with the TAP, while the use of digital signatures makes them unforgeable as in the basic variant. The same can also be said about the TAP commitment since it is signed by the TAP to authenticate its agreement to the association with the given set of clients.

- **Claim Verifiability**
  Claim verifiability also assumes increased significance in the context of the complex variant, since the TAPs verifying a claim have to verify both the authenticity and consistency of the information reported in the claim. As a result, the claim would have to incorporate more information than in the basic variant and this is where the major difference between both solutions lies. The claim now includes the client commitment in place of the client ID as the unforgeability of the commitments makes them ideal to ensure the authenticity of the demand being advertised in the claim. The use of digital signatures provides public verifiability of the commitments while at the same time ensuring the unforgeability property.

5.4.5 Performance Analysis

In this section, we analyze the performance of the complex variant with respect to the performance metrics outlined in the verification framework. We first proceed to analyze the performance metrics pertaining to the computation overhead of the verification framework.

- **Commitment Construction Delay**
  The commitment construction in the complex variant is the same as in the basic variant and as a result, $\tau_{cmtConst} = t_{SIGN}$.

- **Claim Construction Delay**
  The claim construction delay is also the same as in the basic variant, however, the claim now consists of the client commitments instead of the client IDs. The claim construction delay, $\tau_{claimConst} = t_{MD5} + t_{SIGN}$.

- **Commitment Verification Delay**
  As the structure of the commitment has not changed between the basic and the complex variant, the commitment verification delay is the same as in the basic variant and is given as $\tau_{cmtVer} = l * t_{MD5} + t_{VERIFY}$.

- **Claim Verification Delay**
  The structure of the claim has been modified to include the client commitments instead of the client IDs. As a result, the process of claim verification is more computation intensive when compared to the basic variant. Claim verification now involves performing a hash operation to aggregate the client IDs and verify the signature on the TAP commitment in order to verify its

authenticity. The TAP then has to verify the n client commitments included in the claim and also search for collisions between the clients listed in the claim and its associated clients. The claim verification delay is now defined as, $\tau_{claimVer} = t_{MD5} + t_{VERIFY} + n * \tau_{cmtVer} + O(1)$.

We now proceed to discuss the performance metrics pertaining to the communication overhead of the verification framework.

- **Commitment Transmission Delay**
  Commitment transmission in the complex variant is the same as in the basic variant since the structure of the client commitment has not changed. As a result, $\tau_{cmtTrans}$ remains unchanged.

- **Claim Transmission Delay**
  Unlike the basic variant which involves transmission of the client IDs and the signature as part of the claim, the complex variant is required to transmit the client commitments in place of the client IDs to ensure authenticity and verifiability of the demand claim. As a result, the demand claim is much larger in size compared to the client commitment and hence $\tau_{claimTrans}$ in the complex variant is much larger compared to $\tau_{cmtTrans}$.

- **Signaling Overhead**
  The signaling overhead of the complex variant also has to include the overhead of transmitting the client commitments in the claim apart from the overhead from the basic variant. As a result, the signaling overhead per scheduling round in a typical mesh network with k TAPs and n clients with an active session per TAP, can be roughly estimated as $k * (2 * n + 1) * S_{SIGN} + 2 * n * k * S_{MD5}$.

The concrete analysis of the communication overhead based on experimental evaluation of CIRMA-MH augmented with the complex variant is discussed in Chapter 7. The results from this experimental evaluation of the complex variant are discussed along with the results from the evaluation of the other solutions proposed in this thesis.

5.5 Performance Overhead and Solution

It can be observed immediately that the complex variant poses a large communication overhead on the TAP generating the claim. Also, the scheme requires the TAP verifying the claim

to perform signature verification operations roughly proportional to the number of clients included in a claim which can prove to be extremely cumbersome on the TAP. The TAP verifying the claim also has to maintain a list of client IDs it has verified for cross verification purposes. This imposes a memory requirement on the TAP and also requires that the TAP be capable of performing a search operation on this list of clients in a relatively short interval of time.

An enhancement to the complex variant which alleviates some of these performance bottlenecks is presented below. The trade-offs between performance and security as a result of this enhancement is also discussed. Also, while this enhancement improves certain performance aspects of the scheme, it does not significantly affect the security guarantees of the scheme.

5.5.1 Selective Verification of Demand Claims

In the scheme presented above, a TAP verifying a claim would have to perform a number of signature operations proportional to the number of client commitments included in the claim. This imposes a lot of burden on the TAP's computational abilities and hence is not scalable. A possible solution to alleviate the verification burden is proposed in the form of a selective verification scheme where in, a TAP receiving a demand claim from another TAP chooses to verify the claim with a probability $p1$ and verifies a subset of client commitments included in the claim instead of every client commitment. The subset chosen can be random or otherwise based on certain conditions that the TAP is supposed to verify before it can be satisfied with the demand claim.

In the selective verification scheme, a TAP on receiving a demand claim performs the following operations, (i) decides to verify the claim with a probability $p1$ and if it chooses to verify the claim, (ii) verifies the signature on $Comm_{T_j}$, (iii) checks for collisions between the $Comm_{M_i}^{l_i}$s included in the claim and the list of $Comm_{M_j}^{l_j}$s that it maintains for cross verification, (iv) if there are collisions, chooses to verify the colliding client commitment to check for consistency else, (v) if there are no collisions, chooses a random subset of $Comm_{M_i}^{l_i}$s from the claim and verifies their authenticity.

The intuition behind selective verification of client commitments is that, in the complex variant outlined above, multiple TAPs in the network verifying each and every client commitment in a claim would perform the same redundant operations that can be performed by a single uncorrupted node to yield the same outcome. However, by selective verification, the problem is randomly broken and distributed among multiple TAPs which perform the verification. The collective outcome of this process exhibits almost the same probability of detection as the naive approach. Hence, this

Table 5.1: Theoretical Analysis Parameters

Parameter	Description
N	The total number of clients reported in the demand claim.
n	The number of non-existent reported in the claim.
k	The total number of TAPs in the local neighborhood.
m	Percentage of client commitments verified per claim.
$p1$	The probability that a TAP chooses to verify a claim .
p	The probability of detecting a misbehaving TAP.

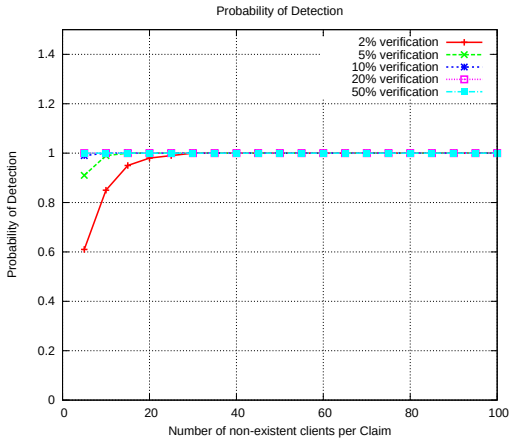
approach minimizes the redundancy in the verification process thus improving the performance by minimizing the verification overhead. While selective verification may not yield a 100% probability of detecting misbehavior, the probability of detection rapidly increases as the extent of misbehavior increases. This is a reasonable solution under the assumption that the WMN is capable of tolerating misbehavior to a small extent and if a node is always assumed to misbehave in a significant manner to gain any advantage.

5.5.2 Security Analysis

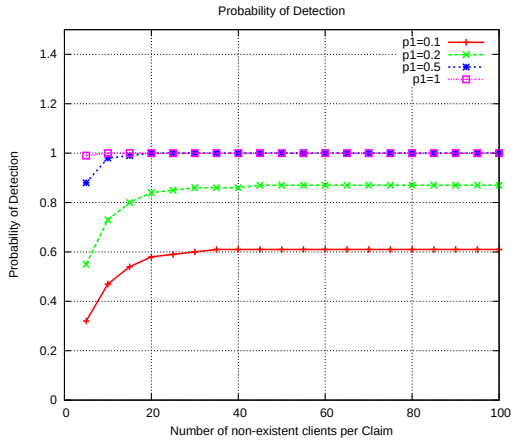
As the selective verification scheme involves probabilistic verification of client commitments and demand claims, the security analysis of this scheme has to analyze the probability of detecting TAP misbehavior apart from the security properties of the verification framework. The security guarantee of the selective verification scheme lies in the fact that, although the scheme does not guarantee 100% detection of misbehavior, it exhibits a high probability of detecting misbehavior which rapidly increases as the extent of misbehavior increases. As a result, the probability of detection is a metric against which the security of our scheme can be validated. The probability of detecting a malicious TAP misreporting demand, given the parameters in Table 5.1 is given by the equation below.

$$p = 1 - (p1 * ((\prod_{i=0}^{m}(N - n - i)/(N - i)) - 1) + 1)^{k-1}$$

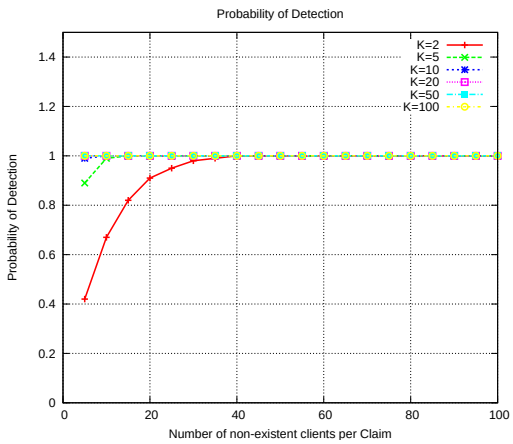
Based on this equation it can be observed that as the number of fake clients reported per claim n increases, the probability of detection p also increases. Similarly, as the percentage of client commitments that are verified by each TAP per claim m increases, the probability of detection shows a rapid increase. Figure 5.2(a) illustrates these two points and demonstrates why the selective verification scheme offers a high probability of detecting TAP misbehavior. Figure 5.2(b) illustrates the effect of the parameter $p1$ on the probability of detection, it can be observed that the choice of



(a) Probability of detection for increasing values of m, where $N = 100, p1 = 1, k = 10$.



(b) Probability of detection for increasing values of $p1$, where $N = 100, m = 10, k = 10$.



(c) Probability of detection for increasing values of k, where $N = 100, m = 10, p1 = 1$.

Figure 5.2: Probability of detecting TAP misbehavior

parameter $p1$ has a significant impact on the probability of detection. Also, it can be observed that as the number of TAPs in the local neighborhood of the misbehaving TAP increases the probability of detection also increases as shown in Figure 5.2(c) since the number of TAPs that will be verifying the claim increases. The plot in Figure 5.2(c) can also be used to analyze the impact of multiple colluding TAPs on the probability of detecting TAP misbehavior. As the number of TAPs in the local neighborhood that collude with the compromised TAPs increases, the number of TAPs actually verifying the claim decreases. However, from Figure 5.2(c) we observe that, even when the number of TAPs in local neighborhood that verify the claim is low, they still guarantee a high probability of detection as the extent of misbehavior increases.

5.5.3 Performance Analysis

The selective verification scheme drastically reduces the burden of a TAP verifying the claim since the TAP no longer has to verify each and every client commitment. Also, the parameter $p1$ further reduces the number of operations a TAP has to perform to verify a claim since the TAP may or may not choose to verify the claim with the given probability. A quantitative analysis of the performance of the selective verification scheme is given below.

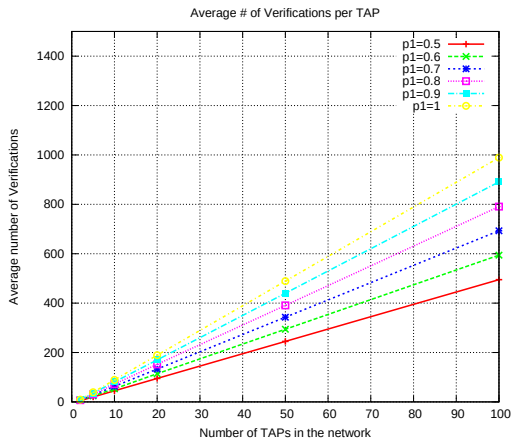
Based on the parameters listed in Table 5.1, the average number of verifications per scheduling round that a TAP has to perform in order to verify the claims it receives from all the other TAPs is given by the equation below.

$$\text{Avg. \# of verifications} = m * p1 * (k - 1)$$

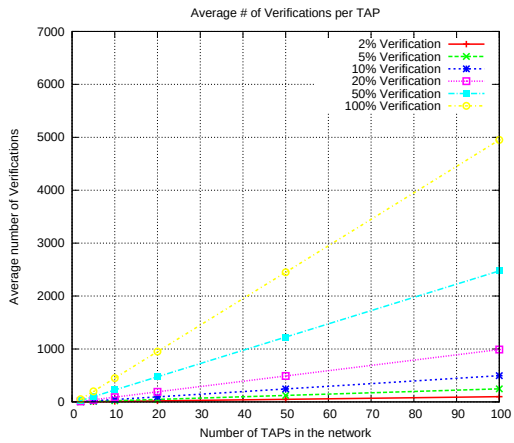
We measure the average number of verifications per TAP with respect to the total number of TAPs in the local neighborhood of a TAP (k). It can be observed that for a given value of m, as the probability of verification $p1$ increases, the average number of verifications that need to be performed increases as shown in Figure 5.3(a). Also, for a fixed probability of verification, as the percentage of commitments verified per claim m increases, the average number of verifications increases linearly as shown in Figure 5.3(b).

5.5.4 Trade-off Analysis

In the previous sections, both the security and the performance of the verification scheme were measured with respect to the following metrics (i) probability of detection and (ii) average number of verifications performed at each TAP. In light of the above observations, it is beneficial



(a) Average number of verifications for increasing $p1$



(b) Average number of verifications for increasing m

Figure 5.3: Average number of verifications per TAP

to perform a trade-off analysis in order to determine the optimal values of the parameters that are critical to the operation of the scheme. It can be immediately observed that the values $p1$ and m play a critical role in determining both the performance and security of the verification scheme. Hence a trade-off analysis based on these two parameters is carried out below.

Both $p1$ and m have a linear effect on the average number of verifications per TAP as illustrated in Figure 5.3. For higher values of m the average number of verifications per TAP drastically increases, while the increase is much slower for increasing values of $p1$. As a result, a smaller m is more beneficial to the performance of the scheme. However, the effect of m on the probability of detection of misbehavior needs to be measured in order to ensure that the selection of a lower m does not drastically reduce the security of the scheme.

It can be immediately observed from Figure 5.2 that, the effect of parameter m on the probability of detection is not as prominent as its effect on the average number of verifications per TAP, this is illustrated by the negligible change in the probability of detection for increasing values of m as illustrated in Figure 5.2(a). However, the effect of increasing values of $p1$ is more prominent on the probability of detection as can be observed in the sharp decrease in the probability of detection for lower values of $p1$ as illustrated in Figure 5.2(b). Based on this analysis, we can infer that a high value of $p1$ and a low value of m would ensure optimum performance while still maintaining a high probability of detection as the extent of misbehavior increases.

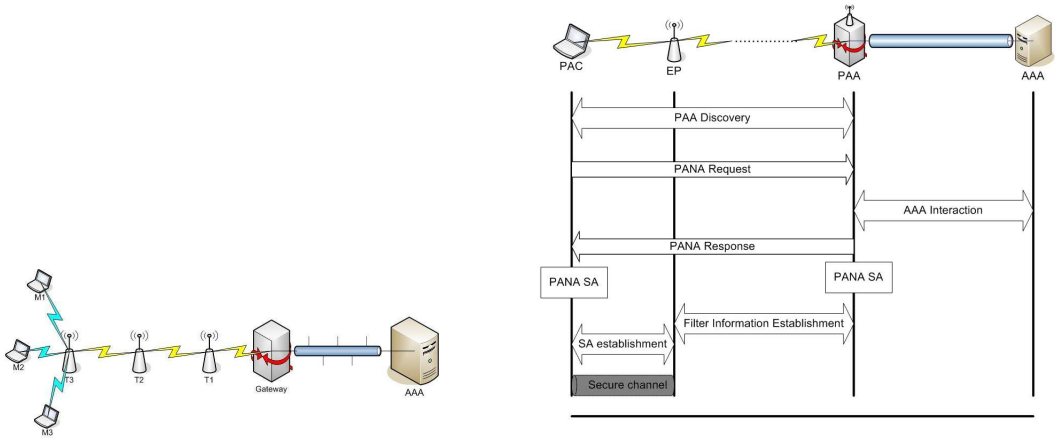
Chapter 6

Misbehavior Resistant Fair Scheduling in a Centralized Authentication Environment

In this chapter, we propose a demand verification scheme for an environment where there exists a central trusted entity in the mesh network which performs operations like network monitoring and admission control. This architecture is a realization of the PANA framework when the PANA Authentication Agent (PAA) and the Enforcement Point (EP) are not collocated but are two distinct components of the network. In such an architecture, there exists an Authentication, Authorization and Accounting server (PAA) and the TAPs (EP) act as pass-throughs during client authentication. All nodes in the network trust the AAA server and use the credentials issued by the server to generate and verify commitments. This centralized trust environment enables the use of the symmetric key cryptography in implementing the demand verification scheme.

6.1 Overview

First, we provide a brief outline of the network architecture for which this scheme is proposed. We then proceed to describe the HAFS [11] based verification scheme in detail. The HAFS based verification scheme has been designed as an alternative to the signature based scheme proposed in chapter 5. We then analyze the security and performance of the HAFS scheme with respect to the properties of the verification framework. The HAFS based verification scheme ensures the de-



(a) Mesh network architecture

(b) PANA Authentication exchange

Figure 6.1: Centralized Authentication Environment

tection of both TAP and client misbehavior while adding significantly reduced overhead compared to the signature based scheme.

6.2 Network and Authentication Architecture

We illustrate the architecture of a simple mesh network in a Centralized Trust Environment in Figure 6.1(a). The mesh network consists of mesh routers $T1, T2, T3$, a Gateway (GW) and an Authentication, Authorization and Accounting (AAA) server. There are three clients $M1, M2, M3$ associated with mesh router $T3$. The Centralized Trust Environment requires that each of the clients has to authenticate itself to the AAA server in order to access the network's resources. This authentication is assumed to be done via an underlying authentication framework like PANA that is already present in the mesh network. A typical client authentication session is shown in Figure 6.1(b).

This type of mesh network architecture is possible for community and public area wireless mesh networks (PAWNs) like the ones deployed by ISPs in small urban and rural communities. These mesh networks offer network services like broadband Internet access to clients with a predefined contract with the ISP. In this environment, authentication is required for admission control and accountability, hence the client is granted access to the network if it can prove that it has successfully negotiated a contract with the service provider.

6.3 A HAFS based Verification Approach

In this section, we consider an adaptation of the Secure Hash based sequential Aggregate and Forward secure Signature (S-HAFS) scheme [11] and propose a verification approach to detect node misbehavior during fair scheduling. The verification scheme builds upon the existing S-HAFS approach using it as the underlying framework for generating verifiable commitments and demand claims.

The verification scheme consists of six phases, (i) Time associated key and TAP token generation, (ii) Client Token generation, (iii) Commitment generation, (iv) Demand claim generation, (v) Time associated key release and (vi) Demand claim verification. While most of these phases are derived directly from the underlying S-HAFS scheme, the context of use of the S-HAFS scheme is different from that required for demand verification. As a result, modifications are made to the S-HAFS scheme to meet the requirements of demand verification. In S-HAFS, an aggregate signature is constructed from the chain root that is generated by the sender. The encrypted chain root obtained from the AAA server is then provided to the verifier, which has to perform a decryption operation before proceeding to verify the signature. In S-HAFS, since there is a single entity that constructs the aggregate signature, one encrypted chain root is sufficient to perform verification. In contrast, in the verification scheme, a claim is generated using commitments from several entities. As a result, if the S-HAFS scheme were to be used to perform verification, it would require the verifier to perform one decryption for each of the entities involved in the claim generation. Hence, the S-HAFS scheme is modified such that the clients generate the chain roots by decrypting the token obtained from the AAA server, while the TAP generates its chain roots as per the S-HAFS scheme. The verification process then requires the verifier to obtain the client chain roots by generating Message Authentication Codes (MAC) and obtain the chain root of the TAP generating the claim after performing a decryption operation. This modification alleviates the burden on the verifier by moving the decryption operation to the clients generating the commitments.

The scheme has three entities each of which is involved in computation during the different phases. A description of the scheme outlining the different phases is given below.

6.3.1 Time Associated Key and TAP token Generation

This is the first phase in the S-HAFS scheme and involves the generation of the time associated encrypted chain roots for all the TAPs in the network. Assuming that scheduling starts from time t_0, at time t_{-1} , the AAA chooses L time periods $t_0 < t_1 < t_2 .... < t_{L-1}$ and generates a hash

chain of time associated keys as

$$tk_0 \xleftarrow{H_1} tk_1 \xleftarrow{H_1} tk_2 \xleftarrow{H_1} tk_3 \xleftarrow{H_1} ..... \xleftarrow{H_1} tk_{L-1}$$

For every TAP T_i, the AAA performs the following actions

1. Generates the time trapdoor keys for the TAP T_i for each of the L time periods as $tk_j^{T_i} = MAC_{tk_j}(T_i)\ \forall j = 0, ..., L-1$
2. Generates a random number $y_0^{T_i}$ which serves as the initial chain root generator and computes the secret chain root $K_0^{T_i,0} = H_2(y_0^{T_i})$.
3. Derives a hash chain of chain root generators as

$$y_0^{T_i} \xrightarrow{H_1} y_1^{T_i} \xrightarrow{H_1} y_2^{T_i} \xrightarrow{H_1} y_3^{T_i} \xrightarrow{H_1} ..... \xrightarrow{H_1} y_{L-1}^{T_i}$$

4. Generates L encrypted secret chain roots for each of the L time periods as $C_j^{T_i} = E_{tk_j^{T_i}}(K_0^{T_i,j})$ $\forall j = 0, ..., L-1$.
5. Sends the following message to T_i.

$$AAA \rightarrow T_i : y_0^{T_i}, C_0^{T_i}, C_1^{T_i} ....., C_{L-1}^{T_i}$$

The AAA server releases $tk_{-1} = H_1(tk_0)$ to the Gateway, all the TAPs and clients currently associated with the mesh network through an authenticated broadcast. The key tk_{-1} will be used to authenticate the release of all the subsequent time associated keys.

6.3.2 Client Token Generation

The client token generation phase is similar to the time associated key and TAP token generation phase. However, unlike the pre-distribution of tokens in the S-HAFS scheme, the tokens are generated for an authentication session between a client entering the network and the AAA server. The structure of the secret chain root for the clients is different from that generated for the TAPs. This is done to optimize the performance of the verification phase while still preserving the security properties of S-HAFS.

A client M_i entering the network at time t_w $(w < L)$ first identifies the TAP it wishes to associate with by choosing the TAP with the strongest beacon signal, in this case $T3$. The client then proceeds to authenticate itself to the network by initiating an authentication session with the AAA server associated with the network. At the end of a successful authentication session, a secure channel is established between M_i and AAA. M_i then generates a hash chain of the form shown below.

$$Z_w^{M_i} \xleftarrow{H_1} Z_{w+1}^{M_i} \xleftarrow{H_1} Z_{w+2}^{M_i} \xleftarrow{H_1} .... \xleftarrow{H_1} Z_{L-1}^{M_i}$$

The values in the hash chain serve as tokens to reactivate a client's association with the network. After generating the hash chain, M_i sends the commitment $H_1(Z_w^{M_i})$ to the AAA server and $T3$. The AAA server on receiving this commitment performs the following actions.

1. Generates the time trapdoor keys for the client M_i for each of the $L - w$ time periods as $tk_j^{M_i} = MAC_{tk_j}(M_i)\ \forall j = w-1, ..., L-1$.
2. Generates $L - w$ encrypted chain root generators for each of the $L - w$ time periods as $C_j^{M_i} = E_{tk_j^{M_i}}(H_2(tk_{j+1}^{M_i}))\ \forall j = w-1, ..., L-1$.
3. Sends the following message to M_i.

$$AAA \rightarrow M_i : tk_{w-1}^{M_i}, C_{w-1}^{M_i}, C_w^{M_i} ....., C_{L-1}^{M_i}$$

6.3.3 Commitment Generation

Commitment generation is similar to the aggregate signature generation in S-HAFS. However, while aggregate signature generation involves generating a publicly verifiable signature on an aggregation of one or more messages, commitment generation involves generating a publicly verifiable commitment to the association between the client and the TAP. While commitment generation by the TAP is identical to the aggregate signature generation by a sender in S-HAFS, commitment generation by the client involves generating an encrypted chain root using inputs from both the AAA server (TTP) and the client (sender), this is done to alleviate the burden on the TAP that verifies the client commitments in a claim.

At time t_w, each of the clients $M1, M2, M3$ associated with TAP $T3$ have been successfully authenticated by the AAA server and possess the encrypted tokens that will be used to generate

their respective commitments. In the beginning of time t_w, each of the clients M_i $(i = 1, 2, 3)$ perform the following operations to generate the commitment.

1. Decrypts $C_{w-1}^{M_i}$ to obtain the chain root generator $H_2(tk_w^{M_i}) = D_{tk_{w-1}^{M_i}}(C_{w-1}^{M_i})$ for the current scheduling round starting at t_w.
2. Generates the secret chain root $K_0^{M_i,w} = H_1(Z_w^{M_i}||H_2(tk_w^{M_i}))$
3. Computes signature on the association $M_i - T3$ as $\sigma_w^{M_i} = MAC_{K_0^{M_i,w}}(M_i||T3)$
4. The client then sends the commitment $Comm_{M_i} =< Z_w^{M_i}, \sigma_w^{M_i} >$ to $T3$.

$$M_i \rightarrow T3 : Comm_{M_i}$$

$T3$ on receiving the commitments from all its clients proceeds to perform the following actions.

1. Computes the secret chain root as $K_0^{T3,w} = H_2(y_w^{T3})$.
2. Aggregates the client commitments as follows $\sigma_w^{M_{1,3}} = MAC_{K_0^{T3,w}}(\sigma_w^{M1}||\sigma_w^{M2}||\sigma_w^{M3})$.
3. Computes the TAP commitment as $Comm_{T3} =< \sigma_w^{M_{1,3}}, C_w^{T3} >$.
4. Computes the chain root generator for the next scheduling round y_{w+1}^{T3} as $y_{w+1}^{T3} = H_1(y_w^{T3})$ and deletes y_w^{T3}.

The TAP then proceeds to generate the demand claim using the client IDs and its own commitment.

6.3.4 Demand Claim Generation

$T3$ generates the demand claim $< M1, M2, M3, Z_w^{M1}, Z_w^{M2}, Z_w^{M3}, Comm_{T3} >$ and forwards it to other TAPs in the network.

$$T3 \rightarrow T1, T2, GW : M1, M2, M3, Z_w^{M1}, Z_w^{M2}, Z_w^{M3}, Comm_{T3}$$

6.3.5 Time Associated Key Release

This phase is identical to the periodic trapdoor release of the S-HAFS scheme. After time interval δt known as the key release interval, the AAA server releases its time associated key tk_w to all the TAPs and clients associated with the mesh network. The time associated key tk_w is used by TAPs to verify the demand claims sent by other TAPs in the network and at the same time used by both the TAPs and clients to obtain the the secret chain roots for the next scheduling round. The time interval δt is chosen such that at time $t_w + \delta t$ all the TAPs have generated and broadcast the demand claim and the claim has been received by the corresponding TAPs that choose to verify this claim. In other words, any commitment or demand claim received after $t_w + \delta t$ is discarded.

6.3.6 Demand Claim Verification

Demand claim verification is similar to the aggregate signature verification in S-HAFS. In S-HAFS, the verifier has to perform a decryption for verifying the signature from every sender, however, the demand claim is a collection of commitments from the TAP and all its associated clients each of which can be considered as an independent sender. As a result, performing a decryption for each client listed in the claim imposes a huge computational burden on the verifying TAP. Therefore, the S-HAFS scheme has been modified such that, a TAP verifying a demand claim has to perform only one decryption to verify the TAP commitment, while computing a MAC to verify a client commitment.

Once the time associated key tk_w has been released by the AAA server, a TAP verifying the demand claim from $T3$ performs the following operations.

1. Checks the list of client IDs specified in the claim for collisions with the local set of client IDs associated with the verifying the TAP.

2. Computes the time trapdoor key for each client M_i specified in the demand claim as $tk_w^{'M_i} = MAC_{tk_w}(M_i)$.

3. Computes the secret chain root for each client M_i specified in the demand claim as $K_0^{'M_i,w} = H_1(Z_w^{M_i}||H_2(tk_w^{'M_i}))$.

4. Computes $\sigma_w^{'M_i} = MAC_{K_0^{'M_i,w}}(M_i||T3)$.

5. Computes the time trapdoor key for TAP $T3$ as $tk_w^{'T3} = MAC_{tk_w}(T3)$.

6. Obtains its secret chain root as $K_0^{'T3,w} = D_{tk_w^{'T3}}(C_w^{T3})$.

7. Computes $\sigma_w^{'M_{1,3}} = MAC_{K_0^{'T3,w}}(\sigma_w^{'M1}||\sigma_w^{'M2}||\sigma_w^{'M3})$.

8. If $\sigma_w^{'M_{1,3}} == \sigma_w^{M_{1,3}}$ included in the TAP commitment, then accepts the demand claim and schedules accordingly, else generates an alert of the form $(T_{verifier}, T_{claimer}, t_w)$ and broadcasts it on all links.

6.3.7 Security Analysis

In this section, we analyze the security of the basic variant with respect to the security properties that are outlined in the verification framework.

1. **Client Authentication**
   The verification scheme works under the assumption that there is an underlying authentication infrastructure in place and all clients entering the network authenticate themselves to the central AAA server in order to use the network resources.

2. **Commitment Unforgeability**
   Commitment unforgeability is a critical requirement that needs to be satisfied to ensure that the commitments can only be generated by a TAP making a demand claim and the clients that are associated with this TAP. Since the commitment generated by a node (Client/TAP) is in essence a form of an aggregate signature generated using the S-HAFS scheme, the commitment satisfies all the security properties that are satisfied by an aggregate signature. The S-HAFS scheme ensures that the aggregate signatures are unforgeable since they can only be generated by nodes which can derive the secret chain roots within the time associated key release interval. The secret chain roots are derived from the tokens generated by the AAA server and client authentication ensures that only authenticated clients have access to these tokens. Also, since the secret chain root is generated using the ID of the client, every secret chain root is unique to the client and hence unforgeable. Predistribution of secret chain root generators to the TAPs during the initialization phase ensures that the TAP commitments are unforgeable as well.

   It can also be observed that the verification scheme requires both the client and the TAP to commit to their association with each other. The client commits to the association by explicitly specifying the TAP and the scheduling round in its commitment, the TAP commits to the

association by generating its commitment from the associated client commitments. As a result the nature of the commitments makes them valid only when generated by the corresponding nodes in the association being specified in the commitments. The commitments ensure the authenticity of the association which in turn can be used to guarantee the authenticity of demand claim advertised by each TAP.

3. **Verifiability of the Demand Claim**
   A TAP has to generate a demand claim in order to request time slots for transmission. The demand claim is used by the other TAPs in the mesh network to determine the number of active flows at the requesting TAP and calculate a transmission schedule accordingly. Since the demand claim is a key component of the scheduling process, ensuring the authenticity and verifiability of the demand claim is a critical requirement. An aggregate signature generated using the S-HAFS scheme is publicly verifiable since any node which possesses the time associated key from the Trusted Third Party (TTP) and the Identity of the sender generating the signature can verify the aggregate signature. As a result, the commitments generated by clients and TAP can be verified by any node in the network after the key release interval δt, and since the demand claim is made up of commitments from the clients and the TAP, the claim is also publicly verifiable.

6.3.8 Performance Analysis

In this section, we analyze the performance of the HAFS based verification scheme with respect to the performance metrics outlined in the verification framework. We first proceed to analyze the performance metrics pertaining to the computation overhead of the verification framework, however, as part of this analysis we do not analyze the overhead during the initialization phase of the HAFS scheme which involves time associated key generation and token generation and distribution.

- **Commitment Construction Delay**
  The construction of the commitment involves generating a Message Authentication Code (MAC) to derive the time trapdoor key for the previous scheduling round from the time associated key released by the AAA server, decrypting the token for the current scheduling round to obtain the chain root generator. The client then performs a hash operation to generate the secret chain root used for the scheduling round, it uses this secret chain root to generate a MAC on the association between the client and the TAP which is used as the client commitment. As a result, the commitment construction delay, $\tau_{cmtConst} = t_{DES} + t_{SHA1} + 2 * t_{MAC}$,

where t_{DES} is the time taken to perform encryption/decryption, t_{SHA1} is the time taken to perform a sha1 hash operation and t_{MAC} is the time taken to compute an HMAC.

- **Claim Construction Delay**
  The most significant operation in claim construction is generating the TAP commitment from the client commitments, this involves performing a hash operation to generate the secret chain root for the scheduling round and perform a MAC operation on the aggregate of the client commitments and compute the chain root generator for the next scheduling round by performing a hash operation. The claim construction delay, $\tau_{claimConst} = t_{MD5} + t_{MAC} + t_{SHA1}$.

- **Commitment Verification Delay**
  A TAP verifying a commitment has to perform the following operations, compute the trapdoor key for the client by generating a MAC on the time associated key released by the AAA server and the client ID. The TAP then performs a hash operation to compute the chain root generator of the client, performs another hash operation to compute the client's secret chain root for the scheduling round and finally computes the client commitment by performing a MAC operation. It then verifies if the generated and cached commitments are same to verify its authenticity. As a result, the commitment verification delay, $\tau_{cmtVer} = 2*t_{MAC}+t_{SHA1}+t_{MD5}$.

- **Claim Verification Delay**
  The claim verification delay involves verifying the commitment for each of the n clients listed in the demand claim. The TAP then proceeds to verify the TAP commitment by computing the time trapdoor key for the TAP making the claim by performing a MAC operation and performing a decryption operation to obtain the TAP's secret chain root for the current scheduling round. The verifier TAP then aggregates the generated client commitments and computes the TAP commitment by performing another MAC operation and finally compares this with the TAP commitment included in the claim. Hence, the claim verification delay, $\tau_{claimVer} = n * \tau_{cmtVer} + 2 * t_{MAC} + t_{DES}$.

We now proceed to discuss the performance metrics pertaining to the communication overhead of the verification framework.

- **Commitment Transmission Delay**
  Commitment transmission in the HAFS based scheme involves transmission of the message

authentication code (MAC) on the association between the client and the TAP and a token from the client which is used for reauthentication. Since the HAFS based verification approach employs the use of symmetric key cryptography, $\tau_{cmtTrans}$ of the HAFS scheme is less compared to $\tau_{cmtTrans}$ in the basic and complex variant of the solution discussed in Chapter 4. $\tau_{cmtTrans}$ in the HAFS based scheme is of the order of transmission time of a MAC.

- **Claim Transmission Delay**
  Unlike the solutions discussed in Chapter 5 which involves transmission of the digital signature as part of the claim, the HAFS based scheme instead transmits message authentication codes and reactivation tokens as part of the claim. While $\tau_{claimTrans}$ is still larger than $\tau_{cmtTrans}$, it is much less compared to the $\tau_{claimTrans}$ discussed in Chapter 4 and again is of the order of transmission of a MAC.

  Signaling Overhead
  The signaling overhead of the HAFS based verification scheme involves overhead from commitment transmission, claim transmission and key broadcast from the central AAA server. As a result the signaling overhead per scheduling round in a typical mesh network with k TAPs and n clients with an active session per TAP can be roughly estimated as $n * k * (S_{MAC} + S_{MD5}) + k * ((n * S_{MD5}) + S_{MAC}) + (r + 1) * S_{MD5}$ where S_{MAC} is the Size of a message authentication code and the S_{MD5} is the size of an md5 hash and r is the number of rebroadcasts of the time associated key released by the AAA server, and $r = k$ in our current implementation of the scheme.

The concrete analysis of the communication overhead based on experimental evaluation of the extended ns-2 implementation of CIRMA-MH is discussed in Chapter 7. The results from this experimental evaluation of the complex variant are discussed along with the results from the evaluation of the other solutions proposed in this thesis.

Chapter 7

Experimental Evaluation and Discussion

In this chapter, we evaluate the solutions proposed in chapters 5 and 6 by performing simulations in ns-2 [21]. Since the verification framework is designed such that it can be integrated with an existing scheduling scheme, the evaluation of the solutions is performed by extending the ns-2 implementation of CIRMA-MH proposed in [3] to include an instance of the verification framework. As part of this evaluation, we evaluate the performance and security of the verification framework with respect to the instance that has been integrated with CIRMA-MH. We first discuss the simulation setup used to perform the evaluation where we discuss the network environment and traffic profiles used to characterize a wireless backhaul mesh network. We then discuss the experiments performed to evaluate the performance metrics outlined in the verification framework. We finally discuss the merits and limitations of the different approaches.

7.1 Experimental Setup

As part of the evaluation we simulate a wireless backhaul mesh network that consists of a Gateway, Transient Access Points(TAPs) and client nodes. The CIRMA-MH scheme proposed in [3] is used as the centralized scheduling scheme and is modified such that, it schedules links based on bandwidth requests from the TAPs, and the TAPs in turn collect and aggregate bandwidth requests from the client nodes. For the sake of simplifying the modification, it is assumed that both the client-TAP and TAP-TAP communication occur on the same control frequency, unlike in a wireless backhaul mesh network where the client-TAP and TAP-TAP communication is assumed to occur on orthogonal frequencies. The simulation setup uses the same parameters listed in Table 3.1(a). The simulation is performed on a chain topology shown in Figure 7.1 with five TAPs connected to

a Gateway, each TAP has a set of associated clients which generate traffic. In order to simplify the experiment it is assumed that the clients are distributed equally among all the TAPs. The clients generate traffic at a constant rate of $1Mbps$ for a duration of 120 milliseconds towards the Gateway and are chosen at random on a per-TAP basis. There is an initial period of 100 milliseconds in order to enable node discovery and route stabilization in the network before the traffic starts. An instance of the verification agent is attached to each node along with the control agent of CIRMA-MH and is responsible for generating client commitments and demand claims and other relevant information with respect to the verification scheme that is implemented. We have also assumed for the sake of simplicity that all the clients are already associated with a TAP in the network during network initialization and are assumed to be static throughout the duration of the simulation. We do not simulate the scenario where clients join, leave the network or migrate from one TAP to the other.

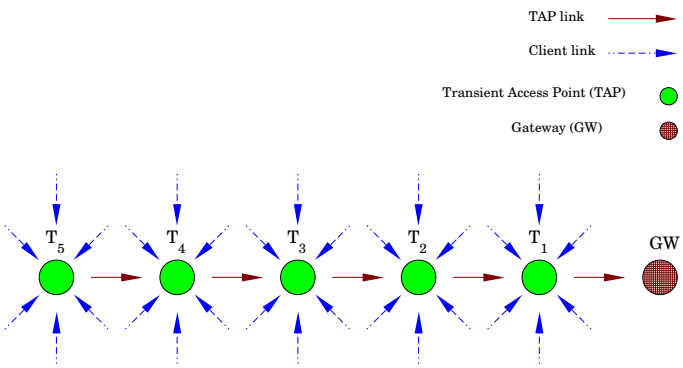


Figure 7.1: Simulation topology

We evaluate the following extensions of CIRMA-MH with respect to the performance metrics outlined in the verification framework. First, we extend CIRMA-MH with the HAFS based scheme proposed in chapter 6. As part of this extension we add the client commitments and demand claim information as headers onto the traffic update messages that are sent as part of control signaling in CIRMA-MH and perform commitment and claim verification during simulation. However, there are two different variations of this extension in order to observe the effect of complete integration and partial integration of the verification framework with the scheduling scheme. In the

HAFS-PGBK extension, the key release message of the HAFS based verification scheme is piggy-backed onto the TDMA schedule message that is unicast to each node after the master node has computed the schedule for the corresponding scheduling round. While in the HAFS-BCAST extension, the key release message is explicitly broadcast by the master node at the end of the key release interval. Second, we extend CIRMA-MH with the basic variant of the signature based verification scheme proposed in chapter 5 and this extension is referred to as SIG-BASIC. However, unlike the HAFS extension of CIRMA-MH, this is not a complete extension as it involves only the specification of the respective header formats for the client commitments and the demand claim and does not involve the actual signature generation and verification that is performed at the nodes. This is a reasonable extension of the scheduling scheme in order to evaluate the communication and signaling overhead as the computation performed at each node does not affect the simulation. Finally, we extend CIRMA-MH with the complex variant of the signature based verification scheme and this extension is referred to as SIG-COMPLEX and is similar to the extension of CIRMA-MH with SIG-BASIC. The extensions and the terminology used to refer to them are listed in Table 7.1. We use ECDSA as the digital signature algorithm for both SIG-BASIC and SIG-COMPLEX.

Table 7.1: Verification Schemes

Scheme	Description
NORMAL	The original CIRMA-MH scheme.
HAFS-PGBK	The HAFS based verification scheme with the key piggy backed.
HAFS-BCAST	The HAFS based verification scheme with the key broadcast.
SIG-BASIC	The basic variant of the signature based scheme.
SIG-COMPLEX	The complex variant of the signature based scheme.

7.2 Results and Analysis

The following parameters listed in Table 7.2 will be used to conduct the evaluation of the verification schemes. We first begin by analyzing the verification schemes with respect to the performance metrics of the verification framework.

7.2.1 Performance Metrics

We first proceed to measure the Claim Transmission Delay ($\tau_{claimTrans}$) of the different solutions in order to analyze the effect of the extension on the performance of the underlying scheduling scheme. The first experiment is to measure $\tau_{claimTrans}$ by varying the number of clients

Table 7.2: Measurement Parameters

Symbol	Description
t_{flows}	Total number of clients in the network having active flows towards the gateway.
n_{flows}	Average number of clients per TAP having active flows.
$N_{clients}$	Number of clients associated per TAP.
l	The average number of active sessions over which a client is associated with the TAP.
N_{TAPs}	Number of TAPs in the network.
$\tau_{cmtTrans}$	Time to transmit a client commitment.
$\tau_{claimTrans}$	Time to transmit a demand claim.
δo	Percentage of control overhead added due to the verification scheme.
$\tau_{claimDetect}$	Time to detect inconsistency in a demand claim.
$\tau_{cmtDetect}$	Time to detect inconsistency in a client commitment.
$\tau_{keyBcast}$	Time taken to broadcast the key to the entire network.

associated with each TAP while keeping the number of active flows in the network constant, this is done in order to keep the overhead added from the client commitments and the demand claim constant and to measure the auxiliary overhead of the verification scheme. In other words, this experiment is carried out to measure the overhead from the other parts of the verification scheme excluding the client commitments and the demand claim. The results are shown in Figure 7.2.

The Claim Transmission Delay is fairly constant for varying number of clients as observed in Figure 7.2 since it depends on the number of active flows in the network, t_{flows} which is constant. It can be observed that, $\tau_{claimTrans}$ is significantly higher for SIG-COMPLEX when compared to the other three extensions HAFS-PGBK,HAFS-BCAST and SIG-BASIC. This is due to the fact that the structure of the demand claim in SIG-COMPLEX includes the client commitments as well and as a result is very large when compared to the demand claims in the other three verification schemes. It can also be observed that $\tau_{claimTrans}$ of SIG-BASIC is comparable to that of HAFS-BCAST and HAFS-PGBK, this is due to the fact that, while structure of the claim in SIG-BASIC is different from HAFS-PGBK and HAFS-BCAST, it is comparable to them in size.

The next experiment is to measure $\tau_{claimTrans}$ by varying the number of active flows in the network while keeping the number of clients associated per TAP constant. This is done to analyze the overhead due to the client commitment and the demand claim that are added on top of the control messages of CIRMA-MH. The results are shown in Figure 7.3.

The Claim Transmission Delay increases as the number of active flows in the network increases. This is due to the fact that, the size of the demand claim depends on the number of clients from which the TAP receives commitments for a particular scheduling round. As the number of

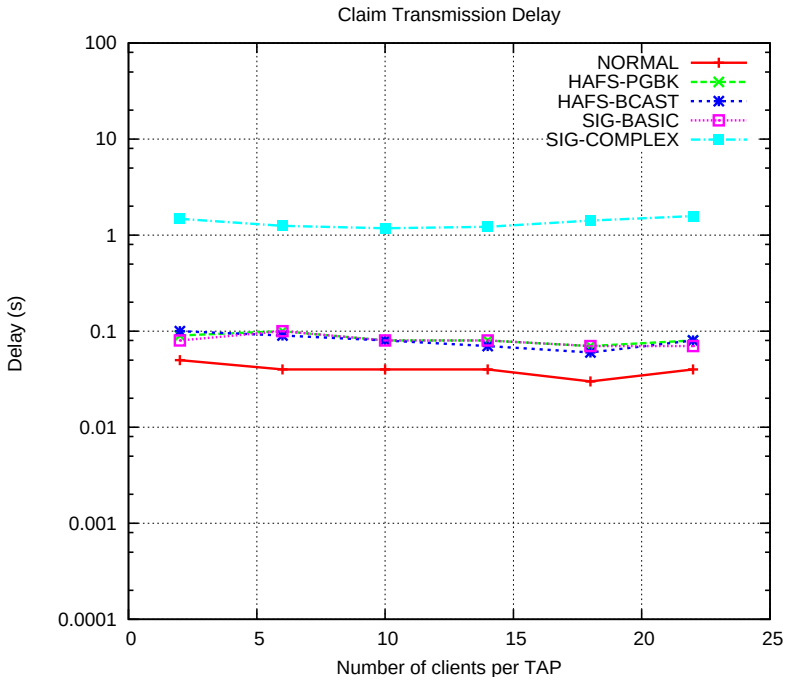


Figure 7.2: Claim Transmission Delay, $\tau_{claimTrans}$ for increasing $N_{clients}$, when $t_{flows} = 5$, $N_{TAPs} = 5$

active flows in the network increases, the size of the demand claim increases and hence $\tau_{claimTrans}$ increases. It can be observed that $\tau_{claimTrans}$ are almost identical for SIG-BASIC, HAFS-PGBK and HAFS-BCAST. Also, $\tau_{claimTrans}$ is subject to the distribution of traffic flows in the network as it determines the number of claim messages generated per scheduling round as well as the size of the claim message. As a result of this random selection of traffic flows, there are slight fluctuations in the measurement of $\tau_{claimTrans}$.

We now proceed to measure the Commitment Transmission Delay for the different verification schemes. In the first experiment we measure $\tau_{cmtTrans}$ for varying number of clients associated per TAP. As stated above this is performed to analyze the auxiliary overhead of the verification scheme on $\tau_{cmtTrans}$. The results are reported in Figure 7.4.

The Commitment Transmission Delay is constant as the number of clients associated per TAP increases as expected since, the number of clients with active flows remains constant and the delay from the underlying scheduling scheme is minimal. It is interesting to observe that $\tau_{cmtTrans}$ is the same for both SIG-BASIC and SIG-COMPLEX unlike $\tau_{claimTrans}$ since the structure of the commitment remains the same both in SIG-BASIC and SIG-COMPLEX. Also, $\tau_{cmtTrans}$ is

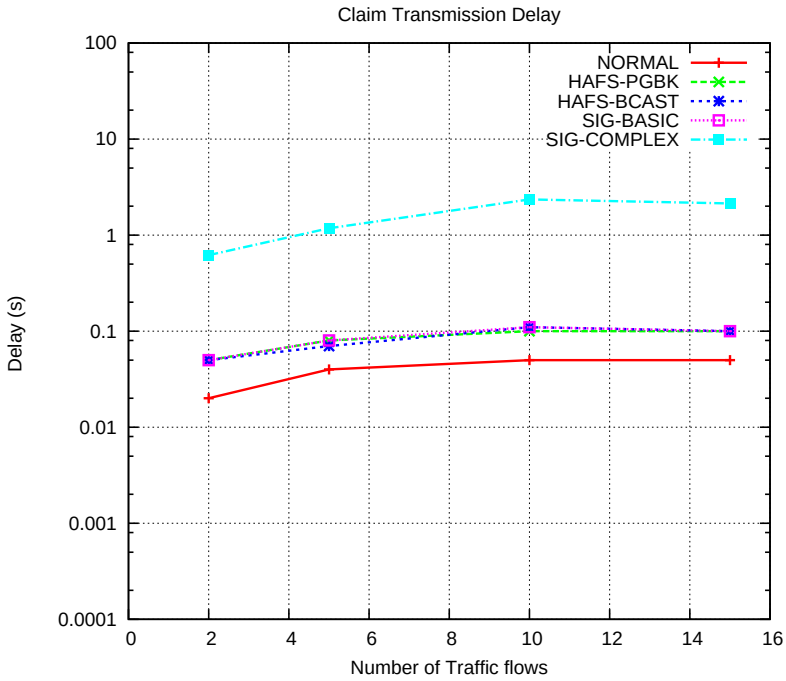


Figure 7.3: Claim Transmission Delay, $\tau_{claimTrans}$ for increasing t_{flows}, when $N_{clients} = 14$, $N_{TAPs} = 5$

identical for both HAFS-BCAST and HAFS-PGBK as the structure of the commitment is identical in both extensions and $\tau_{cmtTrans}$ is not affected by the key broadcast from the central agent.

In the next experiment, we measure $\tau_{cmtTrans}$ for varying number of active flows in the network in order to analyze the overhead due to the client commitment and demand claim added on top of CIRMA-MH. The results for this experiment are shown in Figure 7.5.

The Commitment Transmission Delay increases as the number of active flows in the network increases and is consistent with the normal expectation. This is due to the fact that, as the number of active flows in the network, t_{flows} increases, the number of active flows per TAP also increases. As a result, contention during the transmission of the commitment increases among the clients associated with a TAP. Again similar to the results observed in Figure 7.4, we find that $\tau_{cmtTrans}$ of both HAFS-BCAST and HAFS-PGBK are almost identical, and the same can be observed for SIG-BASIC and SIG-COMPLEX.

We now proceed to measure the signaling overhead δo due to the extension of CIRMA-MH with an instance of the verification framework. The signaling overhead δo is measured as percentage change in the total size of signaling data that is transmitted by CIRMA-MH in the control

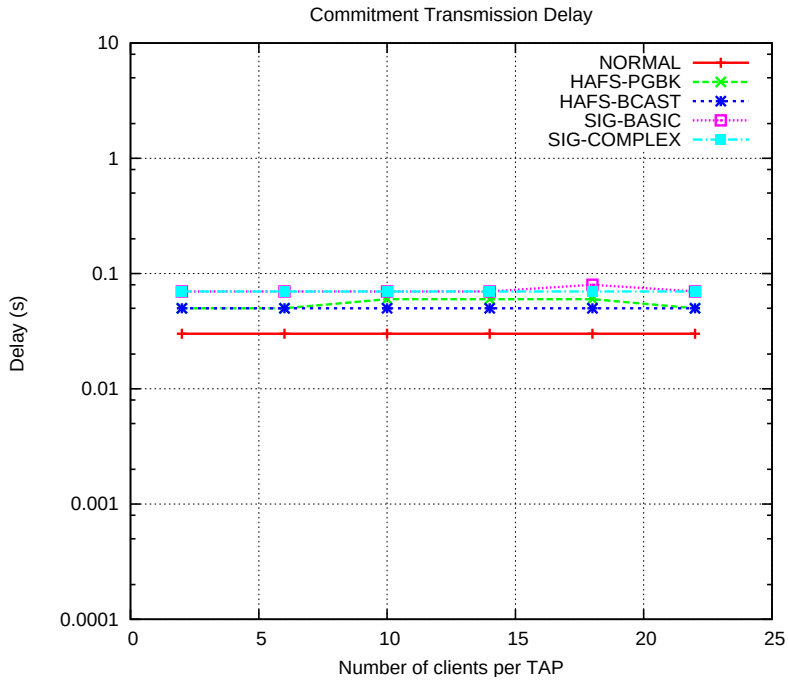


Figure 7.4: Commitment Transmission Delay, $\tau_{cmtTrans}$ for increasing $N_{clients}$, when $t_{flows} = 5$, $N_{TAPs} = 5$

channel during the simulation. This is performed to analyze the overall effect of the verification scheme on the underlying scheduling scheme. In the first experiment, we measure δo with respect to the number of clients associated per TAP. In this experiment, the number of active flows in the network is kept constant in order to keep the overhead from the client commitment and demand claim constant and analyze the signaling overhead added by the rest of the verification scheme. The results of this experiment are shown in Figure 7.6.

It can be observed from Figure 7.6 that the signaling overhead decreases as the number of clients per TAP, $N_{clients}$ increases. This is because as $N_{clients}$ increases, the amount of control signaling in the underlying CIRMA-MH increases, while the amount of control signaling that is introduced by the verification scheme remains constant. However, there are some interesting observations that can be made from Figure 7.6. First, δo introduced by HAFS-BCAST is slightly lesser than δo introduced by HAFS-PGBK, this is due to the fact that HAFS-PGBK involves the key being piggybacked on the TDMA schedule message that is unicast to each node and hence contributes proportionally to the overhead as there are number of nodes in the network. While in HAFS-BCAST, the key is broadcast by the master node and is rebroadcast by the respective TAPs

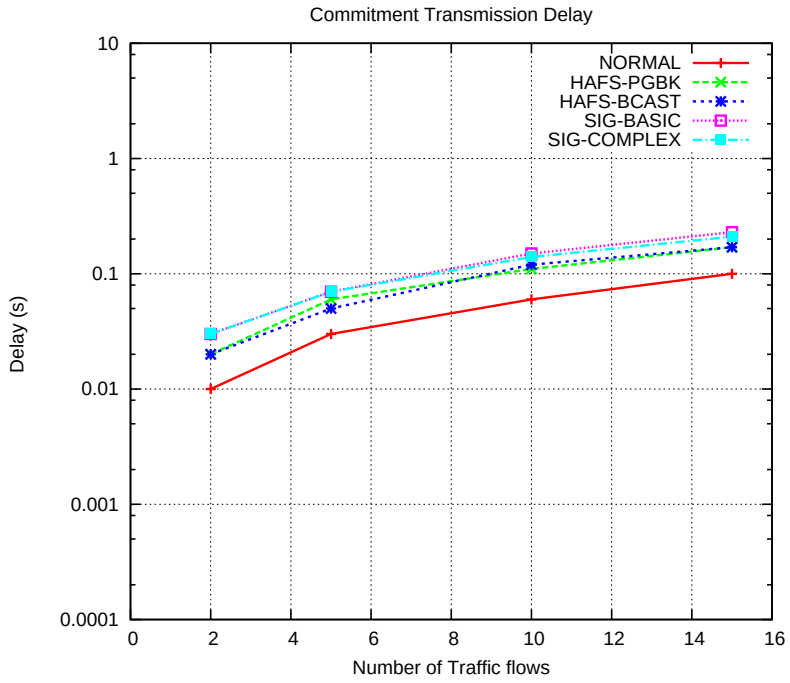


Figure 7.5: Commitment Transmission Delay, $\tau_{cmtTrans}$ for increasing t_{flows}, when $N_{clients} = 14$, $N_{TAPS} = 5$

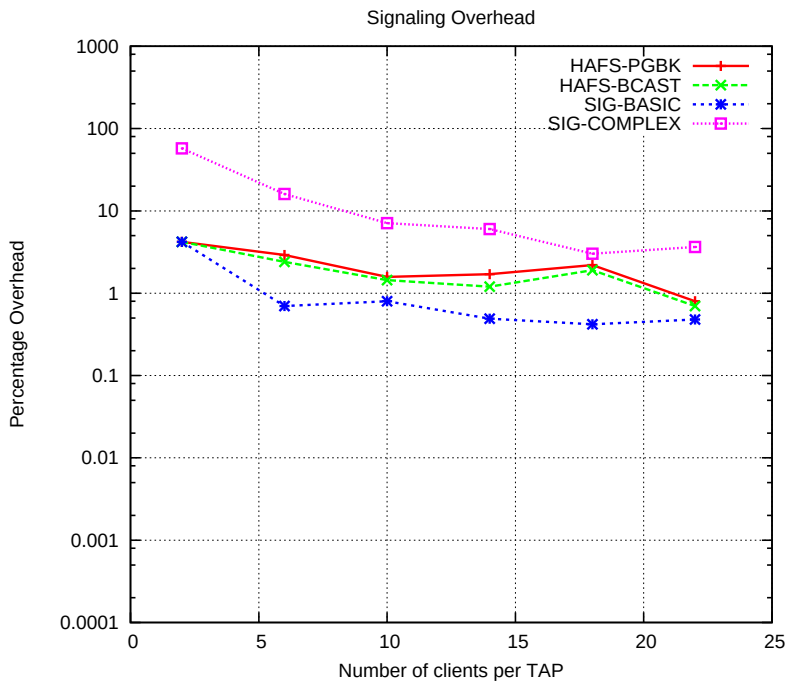


Figure 7.6: Signaling Overhead, δo for increasing $N_{clients}$, when $t_{flows} = 5$, $N_{TAPs} = 5$

in their client neighborhood as a result adding a constant overhead irrespective of the number of clients per TAP. Second, δo falls rapidly in SIG-BASIC and SIG-COMPLEX when compared to HAFS-BCAST and HAFS-PGBK since in the HAFS based verification scheme, additional overhead is introduced by the token distribution and periodic key release by the master node. Third, δo for SIG-BASIC is less compared to HAFS-PGBK and HAFS-BCAST since SIG-BASIC only adds overhead in terms of the client commitments and the demand claim and this overhead is almost comparable to the overhead added due to client commitments and demand claims in HAFS-BCAST and HAFS-PGBK.

In the next experiment, we measure δo with respect to the number of traffic flows in the network. In this experiment, the number of active flows in the network is increased while the number of nodes in the network is kept constant. This is performed to analyze the overhead added by the client commitment and the demand claim in the extension of CIRMA-MH. The results of this experiment are shown in Figure 7.7.

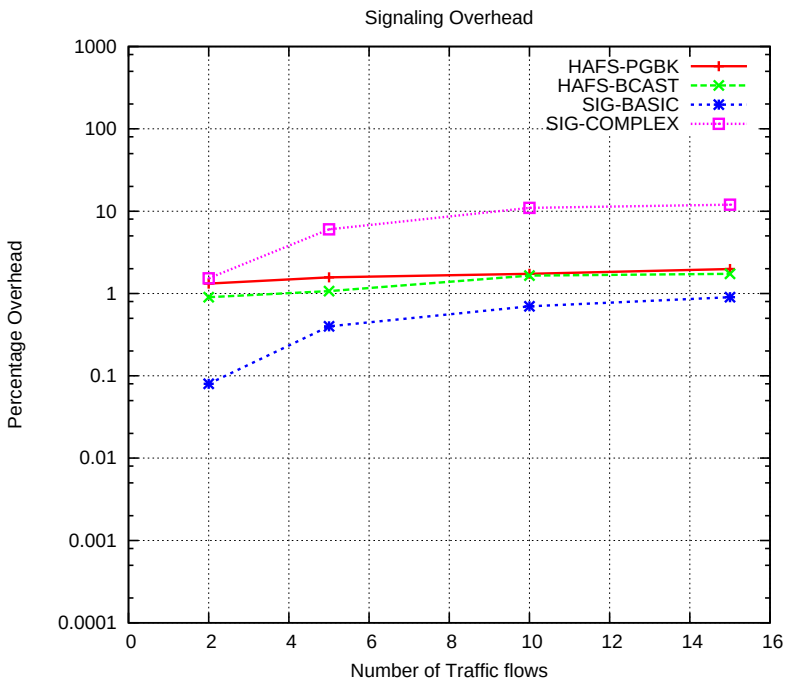


Figure 7.7: Signaling Overhead, δo for increasing t_{flows}, when $N_{clients} = 14$, $N_{TAPs} = 5$

It can be observed from Figure 7.7 that the signaling overhead increases as the number of active flows in the network increases. This is due to the increase in the number of client commitments and demand claims transmitted in each scheduling round. It can also be observed that

the rate of increase of δo for SIG-BASIC and SIG-COMPLEX is higher than HAFS-BCAST and HAFS-PGBK as the amount of overhead added by each client commitment and demand claim is higher for SIG-BASIC and SIG-COMPLEX. As a result, although δo is less for SIG-BASIC for lower number of traffic flows, it is expected to surpass δo of HAFS-PGBK and HAFS-BCAST as the number of traffic flows in the network increases.

7.2.2 Misbehavior Detection Delays

Apart from analyzing the verification scheme with respect to the security properties and the performance metrics outlined in the verification framework, we now proceed to analyze the delay in detecting misbehavior with respect to the scheduling scheme that has been augmented, viz. CIRMA-MH. We use two parameters to measure the efficiency of the schemes in detecting misbehavior during fair scheduling and these are computed from the performance metrics of the verification framework.

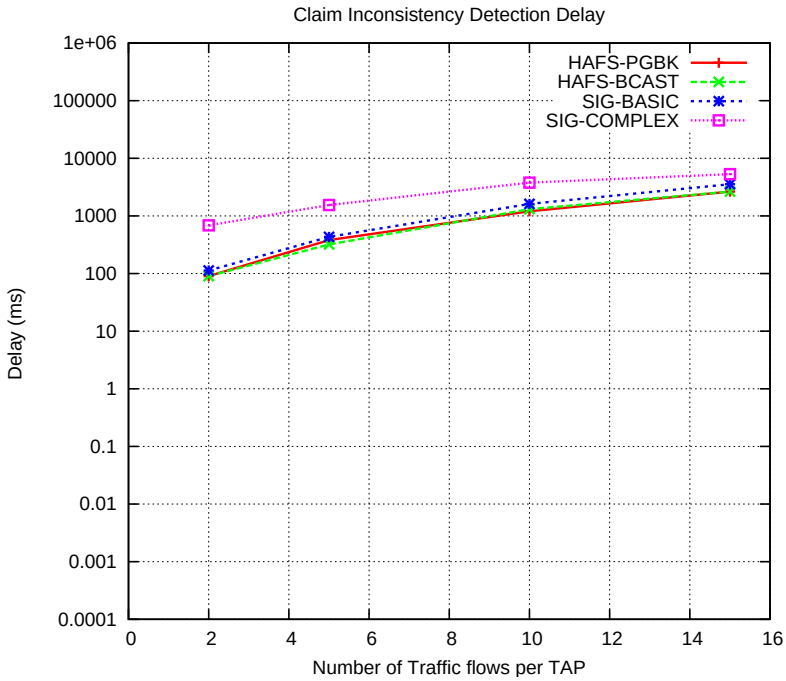
The first parameter, Claim Inconsistency Detection Delay ($\tau_{claimDetect}$), is the delay in detecting inconsistencies in the demand claim that has been reported and can be roughly estimated by the following equation.

$$\tau_{claimDetect} = \tau_{cmtConst} + n_{flows} * \tau_{cmtTrans} + \tau_{claimConst} + \tau_{claimTrans} + \tau_{claimVer}$$

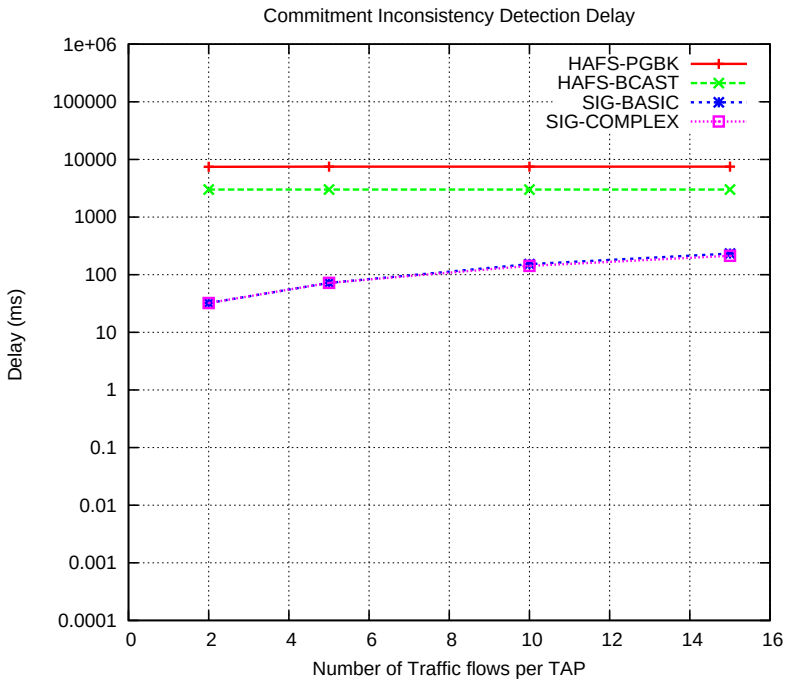
The second parameter, Commitment Inconsistency Detection Delay ($\tau_{cmtDetect}$), is the delay in detecting inconsistencies in the commitments sent by the client and can be roughly estimated by the following equation.

$$\tau_{cmtDetect} = \tau_{cmtConst} + \tau_{cmtTrans} + \tau_{cmtVer}$$

Figures 7.8(a) and 7.8(b) plot the detection delays for increasing number of flows per TAP. It can be observed from Figure 7.8(a) that $\tau_{claimDetect}$ for HAFS-PGBK, HAFS-BCAST and SIG-BASIC are similar while it is significantly higher for SIG-COMPLEX. The added delay in detection for SIG-COMPLEX occurs due to the overhead during claim transmission and claim verification. In Figure 7.8(b), it is interesting to observe that $\tau_{cmtDetect}$ is low for SIG-BASIC and SIG-COMPLEX when compared to the symmetric key schemes HAFS-PGBK and HAFS-BCAST. The reason for this is because, SIG-BASIC and SIG-COMPLEX offer immediate verification of commitments. As a result, a TAP on receiving a client commitment can immediately verify it for inconsistencies before including it in the claim. However, in HAFS-PGBK and HAFS-BCAST, a TAP cannot verify the



(a) Claim Inconsistency Detection Delay, $\tau_{claimDetect}$



(b) Commitment Inconsistency Detection Delay, $\tau_{cmtDetect}$

Figure 7.8: Misbehavior Detection Delay for increasing n_{flows} when $N_{clients} = 16$, $l = 4$, $N_{TAPs} = 5$

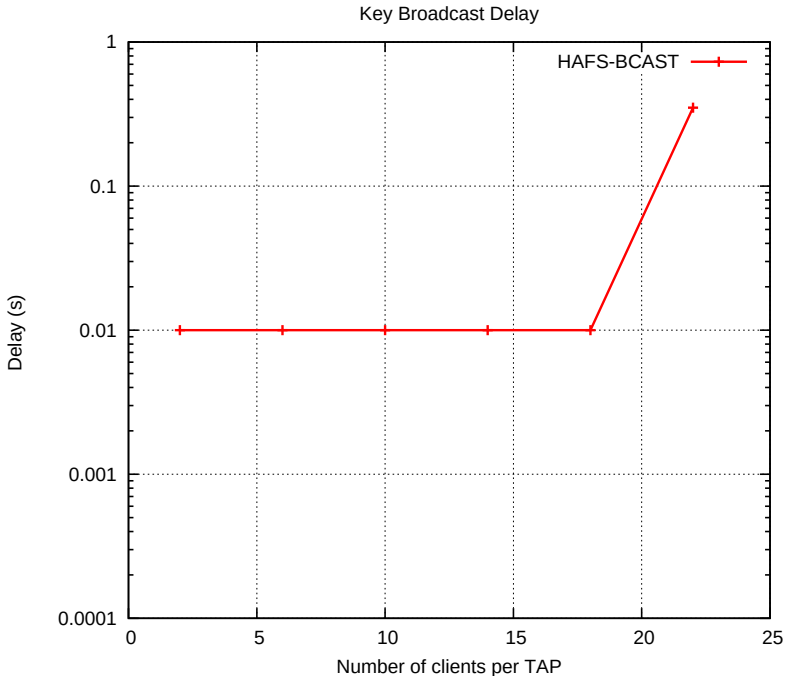
client commitments on reception as it has to wait for the release of the time associated key by the AAA server which happens after the key release interval δt. Also, $\tau_{cmtDetect}$ is higher for HAFS-PGBK since the key is unicast with the TDMA schedule message to every node in the network in contrast to HAFS-BCAST where the key is broadcast after a key release interval of δt = $3s$.

While $\tau_{cmtDetect}$ represents the delay in detecting a misbehaving client in all the verification schemes, the same cannot be said about $\tau_{claimDetect}$. In SIG-COMPLEX, $\tau_{claimDetect}$ represents the delay in detecting a misbehaving TAP since $\tau_{cmtDetect} < \tau_{claimDetect}$ due to the immediate verification of client commitments. As a result, in SIG-COMPLEX a TAP can detect inconsistencies in the commitment before including it in the claim. Hence, any inconsistency in the claim is a clear indication of a misbehaving TAP. While SIG-BASIC also offers immediate verification of commitments, it operates under the assumption that the TAPs in the network are trusted and as a result, $\tau_{claimDetect}$ does not reflect the delay in detecting a misbehaving TAP. However, in both HAFS-PGBK and HAFS-BCAST, $\tau_{cmtDetect} > \tau_{claimDetect}$ since in CIRMA-MH, the claim verification is performed by the master node which is also assumed to be the trusted AAA server. Since the master node possesses the time associated keys for all the scheduling rounds, it can verify a claim immediately upon reception, while a TAP has to wait for the time associated key release by the master node before verifying the client commitment. As a result, any inconsistency in the claim can be caused either by a misbehaving TAP or client and cannot be explicitly attributed to any one node until the TAP has verified the client commitments.

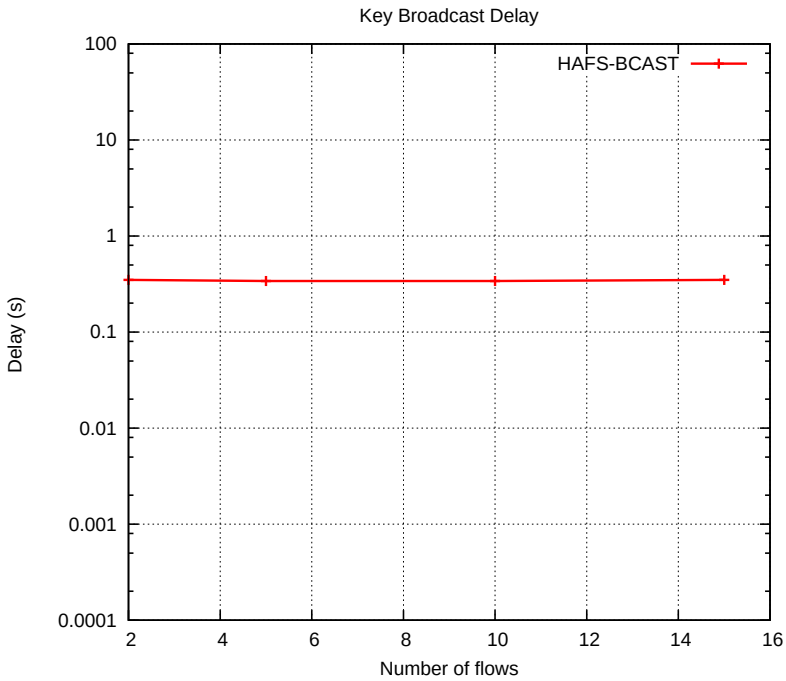
7.2.3 HAFS Scheme Specific Analysis

While the metrics used to analyze the performance of the verification schemes so far are generic to any scheme implementing the verification framework, there could be metrics specific to a verification scheme that are critical in determining its performance. One such metric is the Key Broadcast Delay ($\tau_{keyBcast}$) for the HAFS based verification schemes and is defined as the time taken to broadcast the key to the whole network. The HAFS based verification schemes depend on the key broadcast from the trusted AAA server in order to enable almost all the critical operations of the verification framework namely, (i) Commitment Generation, (ii) Demand Claim Generation and (iii) Demand Claim Verification. In the next set of experiments we measure the key broadcast delay, $\tau_{keyBcast}$ for varying values of t_{flows} and $N_{clients}$.

Figures 7.9(a) and 7.9(b) illustrate the impact of parameters $N_{clients}$ and t_{flows} on the $\tau_{keyBcast}$. It can be observed from Figure 7.9(a) that $\tau_{keyBcast}$ remains relatively constant for in-



(a) Key Broadcast Delay for increasing $N_{clients}$, $t_{flows} = 5$, $N_{TAPs} = 5$



(b) Key Broadcast Delay for increasing t_{flows}, $N_{clients} = 22$, $N_{TAPs} = 5$

Figure 7.9: Key Broadcast Delay, $\tau_{keyBcast}$

creasing values of $N_{clients}$, the sudden increase in $\tau_{keyBcast}$ is because of initial interference from the control messages of CIRMA-MH during the topology discovery phase which extends beyond the 100 second stabilization interval. Similarly, it can be observed from Figure 7.9(b) that $\tau_{keyBcast}$ remains constant for increasing values of t_{flows}.

The reason for the constant value of $\tau_{keyBcast}$ with respect to $N_{clients}$ and t_{flows} is due to the manner in which the key is broadcast as well as the design of the the underlying CIRMA-MH scheme. The key broadcast for the HAFS scheme is designed in a way such that, the key is first broadcast by the AAA server and from there on, the key is rebroadcast by the TAPs in the network. As a result, $\tau_{keyBcast}$ depends on the number of TAPs N_{TAPs} and on the number of hops between the AAA server and the TAPs and this is where the design of the underlying scheduling scheme comes into the picture. CIRMA-MH is designed in such a way that, a node's transmission and interference range in the control plane are five times that of the same values in the data plane. As a result, the topology formed in the control and data plane are completely different and due to this extended communication range in the control plane, most of the TAPs in the network fall within one hop of the AAA server thus keeping $\tau_{keyBcast}$ constant.

The value of $\tau_{keyBcast}$ also has an implication on the security of the verification scheme since the key broadcast might itself be subject to threats from an adversary looking to compromise the verification scheme. However, the low value of $\tau_{keyBcast}$ as shown in Figures 7.9(a) and 7.9(b) and the nature of token distribution in the HAFS scheme preclude an attacker from compromising the key broadcast without being detected.

7.3 Discussion

The experimental analysis of the verification schemes reveal that an instance of the verification framework can be used to extend a fair scheduling scheme that depends on the exchange of information, with minimal extra overhead while ensuring detection of misbehaving nodes. In this section, we discuss the inferences drawn from the above analysis and the trade-offs involved between the different extensions to CIRMA-MH.

While the HAFS based verification schemes, HAFS-PGBK and HAFS-BCAST ensure misbehavior detection with minimal added delays and signaling overheads, these schemes are dependent on certain assumptions about the network. For example, the HAFS based verification scheme requires tight time synchronization among all the nodes in the network in order to ensure the correct running of the scheme. While this may be possible to achieve in a wireless backhaul

mesh network, it may not always be feasible in an environment where the client's have high mobility and a highly transient relationship with the network. In such a case, the likelihood of a client being associated with a TAP for the entire duration of the scheduling round would be very low and thus the traffic schedule would not reflect the true demand at a TAP. The design of the HAFS based verification scheme makes it compatible only with those scheduling schemes where scheduling is performed periodically and hence precluding it from being used with schemes where scheduling is done on-demand. But the biggest shortcoming of the HAFS based verification scheme is the fact that it does not provide immediate authentication. While this limitation is not severe in the case of CIRMA-MH, where the verification is done by the central agent which is also assumed to act as the trusted AAA server, in a distributed scheduling scheme, the TAPs in the network would have to wait for the duration of the key release interval, δt, before being able to verify both the client commitments and the demand claims. This limitation opens the HAFS based verification scheme to a variety of DoS attacks and also affects the efficiency of isolating the misbehaving node after detecting an inconsistency. However, despite these limitations, the HAFS based verification scheme provides guaranteed detection of misbehavior and the best performance among the solutions discussed in this thesis and can be integrated with a broad category of the scheduling schemes provided the scheduling schemes support the conditions that HAFS-BCAST and HAFS-PGBK require or can be modified to do so.

The signature based verification scheme on the other hand can either provide guaranteed detection at the cost of extremely high computational overhead or reduced overhead at the cost of probabilistic detection. However, the signature based verification scheme has certain positive advantages over the HAFS based verification scheme. First, the signature based scheme does not require the presence of a centralized AAA server and as a result can be easily used to extend any underlying scheduling scheme centralized or distributed. Second, the signature based scheme does not require tight synchronization between the nodes except to keep track of the number of sessions over which a client has been associated with a TAP in order to enable fast reauthentication, and thus is capable of supporting on-demand scheduling. Finally, the third and the most important advantage of the signature based scheme is that it provides immediate authentication. As a result, a TAP receiving a client commitment or a demand claim can immediately verify its authenticity thus precluding many types of DoS attacks that can affect the HAFS based verification scheme and can effectively isolate the misbehaving node. Interestingly, while it may be apparent from the above analysis that the basic variant of the signature based scheme SIG-BASIC incurs very minimal overhead in extending CIRMA-MH to enable misbehavior detection, SIG-BASIC can only deal with misbehaving clients

and does not provide a mechanism to detect misbehaving TAPs. As a result, SIG-BASIC is a very ideal candidate for a closed network where the TAPs are assumed to be trusted or there are other mechanisms by which TAP misbehavior can be detected. Also, while the signature based verification scheme using selective verification does provide a mechanism to optimize the computation overhead at each node, the communication overhead remains unaffected since the signature scheme requires the client commitments and the TAP commitment to be transmitted as part of the demand claim in order to enable misbehavior detection. As a result, the signature based verification scheme would be ideal for a network with high capacity links that minimize the delay and clients and TAP with the computational ability to sign and verify digital signatures efficiently.

Chapter 8

Conclusions and Future work

8.1 Conclusions

In this thesis, we address the impact of node misbehavior on fair scheduling in wireless backhaul mesh networks. While there has been research on securing routing protocols against misbehavior, securing fair scheduling schemes against node misbehavior is addressed for the first time in this work. We discuss the different threats due to misbehavior on a generic fair scheduling scheme which relies on the exchange of information between nodes to perform scheduling. We illustrate the impact of a type of misbehavior on a proposed Centralized Integrated Routing and MAC scheduling scheme called CIRMA-MH [3] and demonstrate the fact that with relatively less effort, an adversary can cause a severe reduction in the performance of the network. We then propose a generic verification framework to detect misbehavior in a wireless backhaul mesh network. The verification framework makes use of commitments and demand claims in order to authenticate the traffic information that is exchanged as part of the scheduling algorithm. We then outline the security properties that must be satisfied by a scheme that implements this verification framework and define the performance metrics that are used to evaluate this framework. We propose two implementations of this verification framework.

The first scheme is a digital signature based verification scheme that is ideal in a distributed authentication environment and discuss two variants of this scheme. The first variant called the basic variant is proposed under the assumption that all the TAPs in the network are trusted and the clients are the source of the misbehavior. We then relax this assumption and propose a complex variant of the same scheme in order to deal with TAP misbehavior as well. We analyze both these schemes with respect to the security properties of the verification framework and verify that they

satisfy these properties. We observe that the signature based schemes incur severe performance overhead due to the use of digital signatures and propose several solutions that would reduce the overhead.

We then propose a symmetric key based verification scheme which is ideal in a centralized authentication environment where there exists a trusted AAA server. The symmetric key based verification scheme is derived from the Secure Hash based sequential Aggregate Forward secure Signature (S-HAFS) scheme proposed in [11]. We modify the S-HAFS scheme in order to optimize its performance with respect to the verification framework. We then analyze the security of the HAFS based verification scheme with respect to the security properties of the verification framework and verify that they are satisfied.

We conduct an experimental evaluation of the both the signature based scheme and the HAFS based scheme by performing an ns-2 simulation on the extended versions of CIRMA-MH that has been augmented with each of these schemes. We show through the experimental results that it is both practical and feasible to augment an existing scheduling scheme with an instance of the verification framework while guaranteeing detection of misbehavior.

8.2 Future Work

In this thesis we have addressed the threats due to node misbehavior on fair scheduling in a wireless mesh network. However, we have limited ourselves to solving the problem of misbehavior detection and have not considered the response mechanisms once the misbehavior has been detected. Further research needs to be performed in this direction to formulate an optimal response mechanism that would ensure that the misbehaving node is sufficiently penalized while at the same time minimizing the effect of the misbehavior to a small set of nodes in the network. This response mechanism along with the misbehavior detection mechanism would form a complete solution to mitigate the threat of node misbehavior on fair scheduling in wireless mesh networks.

In this thesis, we have proposed a generic verification framework and have discussed two implementations of this framework using different cryptographic primitives to satisfy the security properties. However, both these implementations have certain limitations that restrict their application to any generic scheduling algorithm. As a result, there is scope for research into designing a more optimized solution that builds upon the verification framework and the two schemes proposed in this work such that, this solution can be integrated with any scheduling scheme irrespective of the assumptions or the network environment in which the underlying scheduling scheme operates. One

possible direction towards this problem would be to formulate a solution that is not just based on the use of cryptographic primitives to enable detection but is also based on the use of certain properties inherent to the network that can be verified by the use of existing mechanisms. For example, the broadcast nature of the wireless medium is one possible example of a property that can be exploited to enable detection of misbehavior, another example would be the use of location dependent properties to detect the presence or absence of traffic in a certain neighborhood.

As part of the experimental evaluation of the verification schemes, we use the Centralized Integrated Routing and MAC scheduling with Min Hop Routing (CIRMA-MH) proposed in [3] as the candidate scheduling protocol that is augmented to enable verification of the traffic demand. While the experimental results from the evaluation of these schemes with CIRMA-MH are promising, they are not representative as CIRMA-MH is just one instance of a class of scheduling algorithms. As a result, the verification schemes have to be evaluated with other scheduling schemes both centralized and distributed in order to analyze the ease of augmentation as well as the effect of the the nature of scheduling on the performance and security properties of the verification framework.

Bibliography

[1] Naouel Ben Salem and Jean-Pierre Hubaux. Securing Wireless Mesh Networks. *IEEE Wireless Communications*, 13(2), 2006.

[2] Jangeun Jun and M.L. Sichitiu. The nominal capacity of wireless mesh networks. *Wireless Communications, IEEE [see also IEEE Personal Communications]*, 10(5):8–14, Oct 2003.

[3] S. Ganu Z. Wu and D. Raychaudhuri. Irma: Integrated routing and mac scheduling in wireless mesh networks. In *Second IEEE Workshop on Wireless Mesh Networks (WiMesh)*, pages 109–118, 2006.

[4] Yan Zhang, Jijun Luo, and Honglin Hu. *Wireless Mesh Networking*. Auerbach Publications, Boston, MA, USA, 2006.

[5] I. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: A survey, 2005.

[6] P. Bahl, A. Balachandran, W. Russell A. Miu, G. Voelker, and Y.M. Wang. Pawns: Satisfying the need for ubiquitous connectivity and location services. *IEEE Wireless Communications Magazine*, 9(1):40–49, 2002.

[7] S. Sarkar and L. Tassiulas. End-to-end bandwidth guarantees through fair local spectrum share in wireless ad-hoc networks. *Automatic Control, IEEE Transactions on*, 50(9):1246–1259, Sept. 2005.

[8] Baochun Li. End-to-end fair bandwidth allocation in multi-hop wireless ad hoc networks. In *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, pages 471–480, Columbus, Ohio, USA, 2005. IEEE Computer Society.

[9] Naouel Ben Salem and Jean-Pierre Hubaux. A Fair Scheduling for Wireless Mesh Networks. In *The First IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005.

[10] Violeta Gambiroza, Bahareh Sadeghi, and Edward W. Knightly. End-to-end performance and fairness in multihop wireless backhaul networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 287–301, New York, NY, USA, 2004. ACM.

[11] Attilla Altay Yavuz and P. Ning. Hash based sequential Aggregate and Forward secure Signature scheme : HAFS. Unpublished Manuscript.

[12] L. Li and P.A.S Ward. Structural unfairness in 802.11-based wireless mesh networks. *CNSR 2007. Fifth Annual Conference on Communication Networks and Services Research.*, pages 213–220, 2007.

[13] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11b. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, 2:836–843 vol.2, 30 March-3 April 2003.

[14] R. Nelson and L. Kleinrock. Spatial tdma: A collision-free multihop channel access protocol. *Communications, IEEE Transactions on [legacy, pre - 1988]*, 33(9):934–944, Sep 1985.

[15] Theodoros Salonidis and Leandros Tassiulas. Distributed dynamic scheduling for end-to-end rate guarantees in wireless ad hoc networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 145–156, New York, NY, USA, 2005. ACM.

[16] OpenSSL : The Open Source toolkit for SSL/TLS. `http://www.openssl.org/`.

[17] Danny Dolev and Andrew C. Yao. On the security of public key protocols. Technical report, Stanford, CA, USA, 1981.

[18] A. Durresi R. Jain and G. Babic. Throughput fairness index : An explanation. *ATM Forum/99-0045*, February, 1999.

[19] Bryan Parno, Adrian Perrig, and Virgil Gligor. Distributed detection of node replication attacks in sensor networks. In *SP '05: Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pages 49–63, Washington, DC, USA, 2005. IEEE Computer Society.

[20] Virgil Gligor. Security of emergent properties in ad-hoc networks. In *Proceedings of International Workshop on Security Protocols, Apr. 2004*, 2004.

[21] The Network Simulator NS-2. http://www.isi.edu/nsnam/ns/.