LAYER 2 SECURITY INTER-LAYERING IN NETWORKS

A Thesis
Presented to
The Academic Faculty

by

Hayriye C. Altunbasak

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2006

LAYER 2 SECURITY INTER-LAYERING IN NETWORKS

To my wonderful parents and my beloved family.

ACKNOWLEDGEMENTS

Completing this Ph.D. has been an exciting long journey, and it would not have been possible without the help and encouragement of many people. First and foremost, I would like to thank my advisor Professor Henry L. Owen for his continuous support and guidance throughout my studies. Professor Owen has been an awesome mentor -he truly enjoys advising. I admire his passion in teaching and conducting research. I really appreciate that he has given me a lot of freedom in my research. It has been my privilege and honor to be associated with Professor Owen and I will be forever grateful for his constant encouragement and wisdom.

I would also like to thank the members of my committee, Professor John A. Copeland, Dr. Randal T. Abler, Professor Mustaque Ahamad, and Professor Yorai Wardi, for serving on my proposal and dissertation committees.

Furthermore, I have benefited immensely from my interactions with Joachim Sokol, Hans-Peter Huth, and Joachim Grimminger at Siemens AG, in Munich. My thesis topic and several ideas contained in this thesis are the outcomes of our interactions with Siemens on the TATI project. I am very grateful to the Siemens AG team for supporting and funding this research as well.

I will always appreciate the friendship of my past and present officemates and colleagues, John Levine, Sven Krasser, Jerapong Rojanarowan, Julian Grizzard, Jeff Gribschaw, David Glandon, Yu-Xi Lim, Ying Xia, and Kevin Fairbanks, for making time enjoyable here.

My most important acknowledgment is to my family members, my son, Fatih, and my husband, Yucel, who have filled my life with happiness. I can not begin to express the gratitude I have for my parents, Rabiha and Muharrem. I will eternally owe them a great debt of gratitude for their endless love and support. Finally, special thanks go to my husband for his encouragement and understanding in this journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS OR ABBREVIATIONS

ACL	access control list.
AES	advanced encryption standard.
AKA	authentication and key agreement.
ALIEN	anonymous identifiers.
AN	association number.
AP	access point.
API	application programming interface.
ARP	address resolution protocol.
AS	authentication server.
BPDU	bridge protocol data unit.
CA	connectivity association.
CAM	content-addressable memory.
CBC-MAC	cipher block chaining message authentication code.
CBDA	certificate-based dynamic host configuration protocol authentication.
CCM	counter mode with CBC-MAC.
CCMP	counter-mode/cipher block chaining message authentication code protocol.
CGA	cryptographically generated address.
CSI	Computer Security Institute.
DA	destination address.
DHCP	dynamic host configuration protocol.
DHT	distributed hash table.
DoS	denial of service.
DTP	dynamic trunking protocol.
EAP	extensible authentication protocol.

EAPOL	extensible authentication protocol over LAN.
EID	endpoint identifier.
ESP	encapsulating security payload.
FBI	Federal Bureau of Investigation.
FQDN	fully qualified domain name.
G2ID	global data link layer identifier.
GCM-AES	galois/counter mode-advanced encryption standard.
GMAC	galois/counter mode as message authentication code.
GMK	group master key.
GSM	global system for mobile communications.
GTK	group transient key.
HI	host identifier, host identity.
HIP	host identity protocol.
HIT	host identity tag.
i3	Internet indirection infrastructure.
ICMP	Internet control message protocol.
ICV	integrity check value.
ID	identifier.
IDS	intrusion detection system.
IE	information element.
IETF	Internet Engineering Task Force.
IMSI	International Mobile Subscriber Identity.
IP	Internet protocol.
IPSec	IP security.
IRTF	Internet Research Task Force.
ISL	inter-switch link.
IV	initial vector.

KaY	MAC security key agreement entity.
KCK	EAPOL-key confirmation key.
KEK	EAPOL-key encryption key.
KHI	keyed hash identifier.
L2ID	local data link layer identifier.
LAN	local area network.
LC	link count.
LSI	local scope identifier.
MAC	medium/media access control, message authentication code.
MACsec	MAC security.
MAN	metropolitan area network.
MDC	modification detection code.
MIC	message integrity code.
MITM	man-in-the-middle.
MPDU	MAC protocol data unit.
MSDU	MAC service data unit.
NAT	network address translator.
NIC	network interface card.
OSI	open system interconnection.
PHY	physical layer.
PKI	public key infrastructure.
PMK	pair-wise master key.
PN	packet number.
PPP	point-to-point protocol.
PRF	pseudo-random function.
PRN	pseudo-random number.
PSK	pre-shared key.

PTK	pair-wise transient key.
PVLAN	private virtual local area network.
RARP	reverse ARP.
RSA	Rivest Shamir Adelman.
RSN	robust security network.
RSP	resolution service provider.
SA	source address, security association.
SAI	secure association identifier.
SC	secure channel.
SCI	SC identifier.
SecTAG	security TAG.
SecY	MAC security entity.
SEND	secure neighbor discovery.
SFR	semantic-free referencing.
SHA	secure hash algorithm.
SID	secret ID, service identifier.
SIM	subscriber identity module.
SPI	security parameters index.
STA	station.
STP	spanning tree protocol.
STS	station-to-station.
TCI	tag control information.
TK	temporal key, transient key.
TKIP	temporal key integrity protocol.
TLS	transport layer security.
TSN	transient security network.
TTP	trusted third party.

UIP	unmanaged Internet protocol.
UMTS	Universal Mobile Telecommunications System.
VACL	virtual local area network access control list.
VLAN	virtual local area network.
WEP	wired equivalent privacy.
Wi-Fi	wireless fidelity.
WLAN	wireless local area network.
WPA2	Wi-Fi protected access 2.

SUMMARY

In this research, we propose an architectural framework to secure the data link layer (Layer 2) in Internet protocol (IP) over Ethernet networks. This new network architecture addresses the weak link between Layer 2 and upper layers and accommodates future network architectures. A security inter-layering concept, incorporating crypto-based Layer 2 identities, is proposed. This new architecture is evaluated by theoretical analysis.

CHAPTER 1

INTRODUCTION

1.1 Motivation and Objective

Network security has become more of a concern with the rapid growth and expansion of the Internet. While there are several ways to provide security in the application, transport, or network layers of a network, the data link layer (Layer 2) security has not yet been adequately addressed. In local networks, security weaknesses in the data link layer enable internal attacks. Although switches and routers have some security features built in, they are not enough to fully ensure the security of local networks. Moreover, these features require network administrators' involvement and are prone to misconfiguration. In addition, data link layer protocols used in local area networks (LANs) are not designed with built-in security features. In this dissertation, we focus on increasing the security of the data link layer of Internet Protocol (IP) over Ethernet networks.

In LANs, we observe that several security flaws are caused by insecure addressing in the data link layer and the weak link between the network and data link layers [19]. First, the media access control (MAC) address namespace of the data link layer is not adequate to provide secure services in local networks. MAC addresses are utilized to uniquely identify hosts/devices in the data link layer. While the MAC address of each network interface card is supposed to be globally unique, it can easily be changed. Second, IP and MAC addresses are not bound securely. IP addresses identify hosts in the network layer. Mappings between IP and MAC addresses in Ethernet-based LANs are accomplished by ARP [81]. However, ARP is not a secure protocol. Third, a compromise in the data link layer may not be detected by upper layers where most security implementations exist. Internetworking reference models are composed of layers. In a layered model, each layer offers security services independent of other layers. Unfortunately, layers lack the ability to inform other layers whether any security measures are utilized or security weaknesses exist.

A common misconception is that wired networks offer security since wired networks are not easily accessible. While the IEEE 802.11i standard [5] greatly improves the security in wireless networks, wired networks have been left far behind in the security area with a false sense of security. In wireless local area networks (WLANs), the main source of security risks is the wireless technology's underlying communications medium, specifically airwaves. Nonetheless, WLANs inherit the vulnerabilities that exist in wired LANs, as well [72]. For instance, the loss of data confidentiality, integrity, and origin authenticity, and the threat of denial of service (DoS) attacks exist in both wired and wireless networks. Moreover, as wireless networks become increasingly secure, attackers start exploring and exploiting weaker points, such as wired LANs, in networks. For instance, removing encryption and authentication from WLAN frames in the wired part of the communications, e.g. after passing an access point, enables attacks on the frames as well as the wireless network. Security in wired LANs needs to be addressed to improve overall security in both networks.

One may still argue that security is not needed in the data link layer, in LANs. After all, to perform an attack in the data link layer of a LAN, one should have access to the LAN, and LANs are accessible by a limited number of users. However, the 2003 and 2004 Computer Security Institute and US Federal Bureau of Investigation (CSI/FBI) Crime and Security Surveys, which are conducted by the CSI/FBI each year to raise the level of security awareness as well as determine the scope of computer crime in the United States, show that computer crime threats to large corporations and government agencies come from both inside and outside their electronic perimeters. According to the reports, the theft of proprietary information was the biggest cause of the financial loss in 2003 [82]. Even though the percentage of internal-system attacks is lower than the external attacks in these reports, the damage caused by internal attacks is potentially greater than the damage by external attacks. Security weaknesses in the data link layer enables internal attacks. Moreover, securing the data link layer not only mitigates attacks but also prevents unauthorized users or devices from using LANs. For instance, an insecure device connected to a LAN may be exploited by an attacker and become a victim to host an attack on the network.

Recently, security issues in the data link layer of local area networks (LANs) have started to receive long overdue attention in standards groups and in the literature [10, 11, 19, 20, 70]. For instance, the IEEE 802.1AE MAC Security Task Group has been formed to secure to local and metropolitan area networks [10]. The IEEE 802.1AE Standard for Local and Metropolitan Area Networks (LAN/MANs): MAC Security specifies how all or a part of a network can be secured transparently to peer protocol entities that use the MAC Service provided by IEEE 802 LANs to communicate [11]. The standard defines MAC security (MACsec) entities in end stations that provide connectionless user data confidentiality, frame data integrity, and data origin authenticity utilizing the IEEE Standard 802.1X. However, MACsec does not specify how the relationships between MACsec protocol peers are discovered and authenticated, as supported by key management or key distribution protocols. The IEEE P802.1af, Authenticated Key Agreement for Media Access Control (MAC) Security, which is a proposed amendment to the IEEE Standard 802.1X-2004, will be used to provide authentication and cryptographic key distribution. Although our proposed data link layer security architecture with a key establishment protocol may be incorporated into MACsec, our approach differs from MACsec since our proposed architecture is primarily based on two design concepts: separation of identities and locations, and a new security inter-layering concept.

First, our approach to securing the data link layer is based on separation of identities (end points) and locations (addresses) in networks. Although the argument of whether network locations and identities should be separated in the Internet architecture is controversial, this concept is becoming a reality with several new protocols. For instance, the Host Identity Protocol (HIP) decouples a host identity from its network address by introducing a new name-space [77]. We believe that decoupling an end point identity from its network location is necessary to improve network and security services with mobility in the Internet. We believe that eventually the concept of the separation of identities and network locations will be widely adopted. This dissertation research is one step toward this goal. We choose to utilize cryptographic identities in the data link layer instead of MAC addresses. There are inherit benefits of securely identifying devices/hosts in the data link layer. It enables authentication of hosts/devices in LANs since devices can prove their identities to other devices in the LANs. For example, MAC cloning attacks will not be possible in this architecture, insecure devices will not be allowed to get access to a LAN, and identities may be used to assign various access privileges in networks. As another example, a user may securely use his/her laptop computer both at home and work by simply identifying him/herself in the data link layer of networks. Moreover, while users desire network security, they are typically negligent, careless, or reluctant to get involved in securing the devices they utilize. Consumers and network administrators wish to have network security with little or no burden. To achieve these objectives, we choose to employ secure identities in the data link layer.

Second, we propose a new security inter-layering concept where layers make the security information in each layer available to other layers. Our main motivation behind the security inter-layering concept is to create top-to-bottom secure and flexible network architectures. Security inter-layering allows the usage of the same namespaces at various layers in networks as well. A lower layer may use a different secure namespace which is obtained from the upper layers each time depending on the applications, user parameters, or network settings. This approach allows devices to provide network access based various criteria, such as user's credentials, application types, host's trustworthiness, etc. For instance, users may gain access to LANs by presenting their passwords to network devices where passwords are utilized as data link layer identities. We choose to utilize a secure namespace from the upper layers in the data link layer security architecture instead of introducing a new namespace. A new namespace would require additional directory services or name resolution processes to locate hosts or bind names. Moreover, a new namespace may create unforeseen additional security vulnerabilities. Security inter-layering may also be employed to create secure bindings among namespaces and to protect against mis-bindings. Furthermore, security inter-layering facilitates our objective of binding the network and data link layers.

In this thesis, we examine the data link layer security in IP over Ethernet networks. We propose to utilize secure namespaces instead of MAC addresses to identify network devices in the data link layer. In addition, we introduce a new security inter-layering concept to provide security services in the data link layer. As a part of the concept of security inter-layering, we bind the identity in the data link layer to one ultimate identity. This ultimate identity information is forwarded by upper layers to the data link layer. We should emphasize that utilizing identities with the security inter-layering concept in the proposed architecture creates a unique and flexible data link layer security architecture.

1.2 Dissertation Outline

The remainder of this dissertation is organized as follows. In the next chapter, we survey related work in the field of data link layer security and provide an overview of new network architectures. In the Chapter 3, we further discuss the security inter-layering concept. We also introduce a flow diagram of the proposed framework architecture. Chapter 4 defines how public and private key-based identities and identifiers are employed in the data link layer security architecture. Chapter 5 describes our key establishment protocol. In addition, we present our thought process of ensuring basic security objectives and argue our design principles of the key establishment protocol. In Chapter 6, we describe the overall network structure, key management, and key hierarchy of our data link layer security architecture. Finally, in Chapter 7, we provide a security analysis of the architecture, summarize a number of open issues in our approach, and present plans for future work and concluding remarks.

CHAPTER 2

LITERATURE REVIEW AND BACKGROUND

The continued expansion and evolution of the Internet has introduced new challenges and revealed some of its shortcomings. For instance, recently, there have been discussions on Internet Protocol (IP) addressing and its functions [36], [45], [53]. IP addresses are overloaded since they are used to represent both network locations and node identities. In addition, the current IP-based naming architecture imposes security problems. Moreover, the basic assumption about trust in the Internet is not valid anymore. Now a wide variety of users share the Internet, whereas early Internet usage was mostly for academic purposes. Furthermore, the end-to-end arguments of the Internet are violated since new mechanisms, such as firewalls, Network Address Translators (NATs), etc., are put into networks. Lastly, the simple delivery model of the Internet, which is known as best effort delivery, is not able to provide specific throughput requirements for more demanding applications. As a result, new architectures and protocols are proposed to improve the Internet in the literature.

The methodology used to accomplish addressing in the data link layer and the mapping between network and data link layers is inadequate to provide secure services in networks as well. IP addresses identify hosts in the network layer, whereas Media Access Control (MAC) addresses uniquely identify hosts/machines in the data link control layer in local networks. The mapping between IP and MAC addresses in Ethernet-based local area networks (LANs) is accomplished by the Address Resolution Protocol (ARP). However, the mapping does not bind the IP and MAC addresses securely. Moreover, even though the MAC address of each network interface card in a network is supposed to be globally unique, it can easily be changed.

This chapter, first, focuses on problems in the data link layer (Layer 2) in IP over Ethernet networks. However, most of the discussion applies to other type of networks as well. Second, we investigate the weak link between the network and data link layers (Layer 3 and Layer 2). Third, we provide an overview of the related work in the literature in the area of the data link layer security. Finally, since some of the ideas in this dissertation are inspired by new and existing network concepts, we outline briefly some of these architectural concepts, such as the Host Identity Protocol, a layered naming architecture for the Internet, the Cryptographically Generated Addresses, IEEE 802.11i, etc., as well.

2.1 Security in the Data Link Layer (Layer 2)

The data link layer (Layer 2) in IP over Ethernet networks is prone to several attacks since the Layer 2 security has not been adequately addressed yet. Three most commonly known Layer 2 sniffing attacks are ARP poisoning, MAC flooding, and port stealing.

ARP is a network layer protocol used to map an IP address to a physical machine address recognizable in the local network, such as an Ethernet address. When a host machine wishes to find a physical address for an IP address, it broadcasts an ARP request, which includes the IP address, on to the network. The host that owns the IP address sends an ARP reply message with its physical address. Each host machine maintains a table, called ARP cache, used to convert IP addresses to MAC addresses. Since ARP is a stateless protocol, every time a host gets an ARP reply from another host, even though it has not sent an ARP request for that reply, it accepts that ARP entry and updates its ARP cache [32]. The process of updating a target host's ARP cache with a forged entry is referred to as poisoning. The attacker sends a forged ARP reply with host B's IP address and the attacker's MAC address to host A. In addition, the attacker sends a forged ARP reply with host A's IP address and the attacker's MAC address to host B. The traffic between host A and B goes through the attacker allowing sniffing. This attack can be performed between a host and a router as well.

Content-Addressable Memory (CAM) tables store MAC addresses, switch port numbers, and Virtual Local Area Network (VLAN) information at switches. They are fixed sizes. In the MAC flooding attack, the attacker floods the switch with MAC addresses using forged ARP packets until the CAM table is full. Then, the switch goes into hub-like mode and starts broadcasting the traffic without a CAM entry.

Another attack possibility involving switches is the port stealing attack. The port stealing attack uses the ability of the switches to learn to bind MAC addresses to ports. When a switch receives traffic from a port with a source MAC address, it binds the port number and the MAC address. In this attack, first, the attacker floods the switch with forged ARP frames with the target host's MAC address as the source address and the attacker's MAC address as the destination address. Since the target host sends frames as well, there is a race condition. The switch sees frames with the same source MAC address on two different ports and constantly changes the binding of the MAC address to the port. If the attacker is fast enough, frames intended for the target host are sent to the attacker's switch port and not to the target host. The attacker steals the port to the target host so the traffic goes through it first, then to the target host. When a frame arrives to the attacker, the attacker performs an ARP request. The attacker asks for the target hosts' IP address in the ARP request. While waiting for the ARP reply, the attacker stops the flooding. The receipt of the ARP reply means that the target hosts' switch port has been restored to its original binding. Finally, after receiving the ARP reply, the attacker forwards the frame to the target host. The attacker repeats this whole process for new frames [85], [80].

In addition to these attacks, there are Layer 2-based broadcasting, Denial of Service (DoS), MAC cloning, and hijacking attacks. In the broadcasting attacks, the attacker sends spoofed ARP replies to the network. These ARP replies set the MAC address of the network router to the broadcast address. This causes all the outbound traffic to be broadcasted enabling sniffing. This type of attack also affects the network capacity. In the Layer 2-based DoS attacks, the attacker updates the ARP caches in the network with non-existent MAC addresses. The MAC address of each network interface card in a network is supposed to be globally unique. However, it can easily be changed enabling MAC cloning. The attacker uses a DoS attack to disable the network connection of the victim and then uses the IP and MAC addresses of the victim. In the Layer 2-based hijacking attack, an attacker takes control of a connection between two computers in the network. For instance, the attacker takes control of telnet session after the victim logs in to a remote computer.

There are several ways to mitigate these types of attacks. One of these actions is to enable port security on the switch. Port security ties a physical port on the switch to a MAC address. It allows the administrator to manually set a list of fixed MAC addresses to a port, or it can be auto-configured by the switch during the first frame transmission on the port. A change in the specified MAC address for a port or flooding of a port can be controlled in many different ways through switch administration. The port can be configured to shut down or block the MAC addresses that exceed a specified limit. Because of the performance impact involved in keeping track of the extra MAC addresses, the recommended best practice is to shut down the port that exceeds the limit [71]. Port security prevents MAC flooding and cloning attacks. However, port security does not prevent ARP spoofing. Port security validates the source MAC in the frame header, but ARP frames contain an additional source MAC field in the data payload, and clients use this field to populate their caches [81].

Another recommended action is to employ static ARP entries. Static ARP entries are permanent entries in an ARP cache. It prevents most of the attacks. However, this method is impractical. Administrators must create new entries on every machine on the network every time a new host is connected, or when a Network Interface Card (NIC) is replaced. Furthermore, it does not allow the use of some Dynamic Host Configuration Protocol (DHCP) configurations.

The third method of the defense is to utilize Intrusion Detection Systems (IDSs). These can be configured to listen for high amounts of ARP traffic. However, IDSs are prone to false positive reports. There are also tools specifically designed to listen for ARP traffic on the networks such as Arpwatch [8]. Arpwatch monitors Ethernet activity and maintains a database of Ethernet (MAC)/IP address pairings seen on the network. It alerts the system administrator via e-mail if any change happens. It is also possible to utilize Reverse ARP (RARP) to detect MAC cloning. In addition, there are methods to detect machines in promiscuous mode on the network.

Lastly, VLANs are employed as a security measure to limit number of clients susceptible to attacks. VLANs create network boundaries, which ARP traffic cannot cross. Then again, VLANs are not always an option and have their own set of vulnerabilities. VLAN Hopping, Spanning Tree, and Private VLAN attacks are some of the possible attacks in VLANs.

VLAN hopping attacks allow the attacker to bypass a Layer 3 device when communicating from one VLAN to another. The attack works by taking advantage of an incorrectly configured trunk port [71]. Trunk ports are generally used between switches to route traffic for multiple VLANs across the same physical link. As a consequence, trunk ports have access to all the VLANs. Inter-Switch Link (ISL, by Cisco) or 802.1Q encapsulation is used with the trunk ports. In addition, the Dynamic Trunking Protocol (DTP) is used to automate ISL/802.1Q trunk configurations. The DTP operates between switches and synchronizes the trunking mode on link ends. The DTP state on an ISL/802.1Q trunking port can be set to Auto, On, Off, Desirable, or Non-Negotiate. In a basic VLAN Hopping attack, a station disguises itself as a switch using a rogue DTP frame if the setting on the port is trunking favorable. As a result, the station becomes a member of all the VLANs. Since the basic VLAN hopping attack is prevented in the new versions of switches, attackers have developed Double Encapsulated VLAN Hopping attacks. This attack is also based on the DTP [84]. This attack uses the fact that switches perform only one level of decapsulation. The attacker sends double encapsulated 802.1Q frames. The first switch removes the first encapsulation and sends it back out. The second switch removes the second encapsulation sending the frame to another VLAN ID. This attack only works if the attacker has the same native VLAN as the trunk. Again, the setting on the port should be trunking favorable for this type of attack. In this attack, the attacker can only send frames (unidirectional). To mitigate this type of attack, administrators should disable Auto-trunking, use a dedicated VLAN ID for all trunk ports, disable unused ports and put them in an unused VLAN, and avoid using VLAN 1 (only the defaults are allowed in VLAN 1).

Spanning Tree Protocol (STP) is a link management protocol that provides loop-free topologies in a redundant Layer 2 infrastructure. In STP, messages are sent using Bridge Protocol Data Units (BPDUs). The STP elects a root bridge to prevent loops in a network. A root bridge continuously transmits network topology information to other bridges, using the STP. The root bridge always forwards frames out to all its ports. The Standard 802.1D STP takes about 30-50 seconds to deal with a failure or root bridge change. The attacker sends BPDUs to force these changes creating a DoS condition on the network [46]. The attacker can become a root bridge sending BPDU messages. Then, the attacker performs DoS, man in the middle, etc. types of attacks. The attacker is required to be dual homed to perform this attack. There are two features on switches used to mitigate this type of attack: BPDU Guard and Root Guard. BPDU Guard disables ports upon detection of a BPDU message on the interface. Root Guard disables interfaces that become the root bridge due to their

BPDU advertisement.

Private VLANs (PVLANs) are used to create distinct networks within a VLAN. PVLANs work by limiting which ports within a VLAN can communicate with the other ports in the same VLAN. The attacker may send a frame with a rogue MAC address (the one of the router) but with the IP address of the victim. Switches do not forward the frame to the victim, but the router forwards the packet to the victim. This way the intended PVLAN security is bypassed. In this attack, the attacker can only sends frames (unidirectional). In order to mitigate this attack, an ingress Access Control List (ACL) can be setup on the router interface, or VLAN ACL (VACL) can be used.

In LANs, Dynamic Host Configuration Protocol (DHCP) is used to dynamically allocate IP addresses to computers for a time period. It is possible to attack DHCP servers causing DoS in the network or impersonate a DHCP server. For instance, an attacker acting as a DHCP client may cause a DoS attack by generating a large number of DHCPDISCOVER messages to request IP addresses, spoofing a different MAC address for each message. The attacker (the rogue client) responds to the resulting DHCPOFFERs to quickly exhaust available IP addresses at the DHCP servers. Even though it is possible to use ARP Request and PING messages to query the addresses used in the network, the rogue client may listen to these messages and answer them. Some DHCP servers use a list of specific MAC addresses to restrict clients. However, since DHCP clients broadcast their MAC addresses to request service, these MAC addresses can easily be spoofed by an attacker for a later use. An attacker may use a DoS attack to prevent a target/victim machine accessing to the network. The attacker, then, renews the IP lease of the victim's machine so that the attacker can use the victim's IP address. In addition, it is possible to gain service on a network by listening a valid MAC address and then spoofing it [84]. There are other methods that exploit DHCP service. Authentication of the DHCP messages is required to prevent these types of attacks.

The layer 2 attacks presented in this section are not comprehensive. Other attacks worth mentioning are Multicast Brute-Force Failover Analysis, Random Frame Stress attack, and attacks based on proprietary protocols. Furthermore, most of the network management protocols are insecure causing additional vulnerabilities. Attacks summarized in this section reveal that a secure data link layer architecture is essential to prevent these vulnerabilities.

2.2 The Weak Link Between the Data Link Layer (Layer 2) and the Network Layer (Layer 3)

ARP is a simple stateless protocol providing mapping between IP addresses and physical machine addresses recognizable in local networks. It is an important part of the Layer 2 and Layer 3 link. This simple protocol does not include any type of authentication, leading to major insecurity.

There are three core problems related to ARP that cause the weak link between Layer 2 and Layer 3 and create an overhead. First, there is no secure binding of IP and MAC addresses. Ideally, there should be a trust mechanism to identify hosts and tie their identities to end-point identifiers used by upper layer protocols.

Second, ARP does not allow for more than one resolution to be done in the same packet. ARP and the existing naming structure do not directly allow multi-homing features where a host has several network interfaces.

Third, ARP introduces an overhead by constantly mapping IP and MAC addresses. In IP over Ethernet networks, after a device identifies itself with a MAC address and establishes its IP address, a more efficient protocol may use only the IP address to address the device.

DHCP is a critical part of the Layer 2 and Layer 3 link, as well. DHCP servers assign temporary IP addresses to hosts in networks. DHCP provides further configuration information such as a subnetwork mask, default gateway, DNS servers, etc.

However, DHCP is an inherently insecure protocol.

There are two core problems regarding DHCP that cause a weak link between Layer 2 and Layer 3. First, when DHCP servers lease IP addresses to clients, they do not enforce a secure binding between IP and MAC addresses, which may be used to verify the authenticity of any frame/packet later on the network. Second, when clients/machines are identified by MAC addresses, it does not ensure if the clients are who they claim to be. It is one of the goals of this thesis to remove this weak link between Layer 2 and Layer 3.

2.3 Related Work

In this section, we provide an overview of token-based and delayed authentication methods to authenticate DHCP messages, and the 802.1AE Media Access Control (MAC) Security (MACsec) to secure bridged Local or Metropolitan Area Networks (L/MANs) [10]. In addition, we provide a summary of the IEEE 802.11i, which is an amendment of the IEEE 802.11 wireless LAN standard [5] and is proposed to improve the security in wireless networks. We also summarize the Extensible Authentication Protocol over LANs (EAPOL), which employs an authentication server to authenticate each user on a network. We utilize some concepts defined in these architectures, such as access control and key hierarchy, in our new data link layer security architecture.

In this section, we also discuss some of the new network concepts and architectures in the literature. Researchers focus on various aspects of the Internet to address the problems of today's Internet. For instance, the Host Identity Protocol (HIP) [77] introduces a new namespace, the Host Identity. HIP maps an end-point identity to a host identity. Alternatively, Balakrishnan *et.al.* [31] propose a new layered naming architecture for the Internet to address architectural problems in the current Internet. While, in the new layered naming architecture for the Internet, new namespaces try to decouple host identities, services, and content from locations or hosts, Cryptographically Generated Addresses (CGAs), on the other hand, try to bind identities or security parameters to IP addresses to verify the authenticity of messages received from addresses or hosts. Moreover, there has been growing interest in anonymous identifiers and related concepts, such as anonymity, pseudonymity, unlinkability, and privacy, in the Internet community. Although we do not directly utilize these new concepts or architectures, they are beneficial to understand the design logic behind our new data link layer architecture.

2.3.1 Secure DHCP

In order to secure DHCP, the Internet Engineering Task Force's (IETF) DHCP working group has released RFC 3118 describing token-based and delayed authentication methods to authenticate DHCP messages [50]. In the token-based authentication method, servers and clients exchange passwords or tokens in plain text over the wire. This method does not provide strong security for DHCP. The delayed authentication method utilizes a shared symmetric key to mutually authorize DHCP clients and DHCP servers. In the delayed authentication method, the key is not sent over the wire. In addition, it uses a nonce (an arbitrary number which occurs only once) or the time in the DHCP packets to prevent replay attacks. The client and the server agree on a secret ID (SID) that references the shared secret key without sending it over the wire. This secret key is used to hash DHCP packets.

In addition to these methods, Glazer *et. al.* [58] have proposed a Certificate-Based DHCP Authentication (CBDA) method. In both delayed authentication and CBDA, authentication information is sent in each DHCP packet as an option. CBDA implements the core components of delayed authentication. However, CBDA utilizes X.509 certificates or certificate chains with a common signer in the option field of the DHCPDISCOVER and DHCPOFFER packets. In CBDA, in the DHCPREQUEST and DHCPACK packets, only the signed hashes of the packets are sent in the options.

Another authentication method worth mentioning is the DHCP Authentication via Kerberos V [67]. This authentication method authenticates only the client. Furthermore, it involves communication with a Kerberos server in addition to the DHCP server communications.

These DHCP authentication methods still have some issues. For instance, the delayed authentication requires that shared cryptographic keys are previously distributed to the clients and servers. Another issue of delayed authentication is key flexibility. When a client changes networks, it requires a different key to access the new DHCP server. Since different networks require different keys, this method introduces the issue of managing multiple shared secret keys. The authentication method via Kerberos identifies only the client and does not prevent rogue servers. CBDA has some issues as well. In practice, DHCP packets are kept small. However, certificates are large and may cause problems if long certificate chains are used. There are also trust issues present in any certificate chaining protocol [58]. Moreover, the certificate revocation policy may be very complicated to set up. Finally, as long as there is no authentication method used in Layer 2 frames, one can use a DoS attack to prevent a victim from accessing the network temporarily and spoof its IP and MAC addresses. There is still a need to incorporate an authentication method with DHCP that binds IP and MAC addresses in the network at any point.

2.3.2 MACsec [10], [11]

Recently, the 802.1AE Media Access Control (MAC) Security Task Group has been formed to secure bridged Local or Metropolitan Area Networks [10]. The IEEE 802.1AE Standard for Local and Metropolitan Area Networks (LAN/MANs): Media Access Control (MAC) Security specifies how all or a part of a network can be secured transparently to peer protocol entities that use the MAC service provided by

IEEE 802 LANs to communicate [11]. The standard defines MAC security (MACsec) entities in end stations and bridges that provide connectionless user data confidentiality, frame data integrity, and data origin authenticity. The goal of the standard is to facilitate secure communication over publicly accessible LAN/MAN media for which security has not already been defined, and allow the use of the IEEE Standard 802.1X in additional applications. However, the standard's scope does not include key management and the establishment of secure associations [10], [11].

MACsec supports connectionless data integrity, data origin authenticity, confidentiality, and replay protection security services. It may also limit denial of service attacks. However, MACsec does not support non-repudiation or protect against traffic analysis.

MACsec provides security services on a frame by frame basis without introducing any additional frames. MACsec introduces an additional transit delay due to the increase in the MAC Service Data Unit (MSDU) size required to convey security information and essential buffering requirements.

MACsec defines how the MAC Security Entity (SecY) operates with the MAC Security Key Agreement Entity (KaY). Each KaY discovers the KaYs present in other stations attached to the same LAN, mutually authenticates and authorizes those stations, and creates and maintains the secure relationships between the stations that are used by the SecYs to transmit and receive frames [11]. However, MACsec does not specify how the KaY works.

A secure Connectivity Association (CA) is created to provide the MAC Service and MACsec for connectivity among stations. Each SecY only participates in a single CA at any one time, and administrative controls can limit the number of peer SecYs that can participate in that CA. Each CA is supported by Secure Channels (SCs). There is one SC for the secure transmission of frames from one of the systems to all

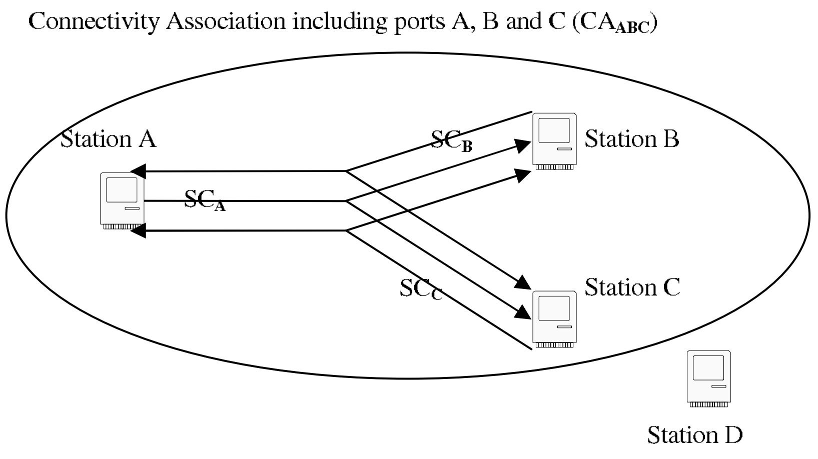
Connectivity Association including ports A, B and C (CA_ABC)
Station A
Station B
Station C
Station D
SC_A
SC_B
SC_C

Figure 1: A Secure Connectivity Association (CA) and Secure Channels $(SC_A, SC_B, and\ SC_C)$ among three stations.

the others in the CA. Figure 1 shows the CA created by MACsec Key Agreement including the stations A, B and C. This CA excludes station D. In the figure, the three SCs that support the CA provide secure communication among the stations. All the SCs use the same cipher suite at any one time. Each SC is identified by a unique 48-bit Universally Administered MAC Address, identifying the system of which the transmitting SecY is part, concatenated with a 16-bit Port number, identifying the SecY within that system. Each SC comprises a succession of SAs, each SA representing a single value of the transient session key(s) used for a period by the cipher suite to support the SC. Each SA is identified by the SC Identifier (SCI) concatenated with an Association Number (AN). The Secure Association Identifier (SAI) thus created allows the receiving SecY to identify the SA, and thus the session key(s) to be used to decrypt and authenticate the received frame. The AN, and hence the SAI, is only unique for the SAs that can be used or recorded by participating SecYs at any instant. When the service guarantees provided include replay protection, the MACsec protocol requires a separate replay protection sequence number counter for each SA, as well.

The SecY provides both secure and insecure services to the users of its Controlled Port and Uncontrolled Port respectively, where Controlled and Uncontrolled Ports are part of the 802.1X. The SecY operates without integrity, origin, or confidentiality protection if the Null Cipher Suite is selected by management. The services provided by the SecY when a cipher suite is selected include the MAC Service Data Unit (MSDU) encryption, Integrity Check Value (ICV) calculation to protect the MAC Protocol Data Unit (MPDU), and inclusion of a SC field. The SC presents the address where the encryption is applied. In a multipoint or Provider Bridge network, the MAC Source Address (SA) and Destination Address (DA) are not the addresses of the intermediate devices that are encrypting and decrypting, they are the original addresses, the end to end addresses. If the SecY is a part of the Bridge stack, its address will not be seen at the endpoints. In that case, in order to make the knowledge of where the encryption is applied available, the SC is used to provide the address of the Bridge port.

The SecY can include a Security TAG (SecTAG) in the initial octets of the provider MSDU, prior to the user data and ICV. The MACsec protocol specifies the mandatory cipher and integrity suites as Null, Galois/Counter Mode-Advanced Encryption Standard (GCM-AES), and GCM as Message Authentication Code (GMAC). The cipher suites, except the Null Cipher Suite, provide confidentiality or integrity or both confidentiality and integrity. The addition of a SecTAG is required for the cipher suites, except the Null Cipher Suite. KaY associated with the SecY provides and periodically updates the keys for the cipher suite. If confidentiality is required, the user data parameter (MSDU) is encrypted. The destination address, source address, and the SecTAG fields are not encrypted. If data integrity is required, an Integrity Check Value (ICV) is calculated over the destination address, source address, SecTAG, and user data (after encryption, if applicable). A simple frame format for Ethernet using MACsec is presented in Figure 2.

The SecTAG provides protocol discrimination and version numbering as well. It is also utilized to identify the Secure Channel (SC) used to transmit the frame and

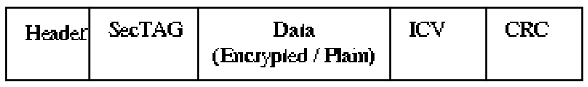

Figure 2: Ethernet frame format with MACsec.

the Secure Association (SA) within the context of the SC. Furthermore, the SecTAG contains a Packet Number (PN) to ensure that every frame transmitted on any given SA is unique. The PN is used to provide replay protection. Finally, the SecTAG allows the length of the MPDU to be unambiguously determined in environments where media access control methods (e.g. 802.3) pad transmitted frames to enforce minimum frame size.

When a Cipher Suite is selected (except the Null Cipher Suite), the SAI decoded from the SecTAG of a valid MPDU, the destination MAC address, source MAC address, the octets of the SecTAG, the octets of the Secure Data and ICV are presented to the Cipher Suite implementation. The Cipher Suite implementation identifies the validation parameters associated with the SA for the received frame using the SAI. Then, (using the validation parameters) it validates the addresses, the SecTAG, and the User Data and decrypts (if encrypted) the User Data. If any of the parameters are invalid, or MSDU extraction fails, the received frame is discarded, and no indication is made to the user of the SecY.

The standard notes that if the security transform for 802.3 is applied only to data, not to control frames, it will not offer protection against disclosure due to ARP spoofing, in which an attacker sends gratuitous ARP messages claiming to have IP addresses of other stations. The attacker can then intercept, read, and alter messages between any two points in a point to point topology. The ICV will be recalculated and the altered data will pass the integrity check. If the message is cryptographically protected above Layer 2, for example with IPsec, the data cannot be read or changed by the attacker. This threat occurs at the intersection between Layer 2 and Layer 3. In addition, there is no protection against legitimate users in ARP spoofing. It is a case of legitimate user with bad intentions. MACsec does offer the ability to identify the bad user. The ICV is recomputed every time the frame presented to the MAC layer, so a man-in-the-middle (legitimate user) can make unauthorized changes and it will pass the integrity check. Thus, MACsec provides point to point integrity, but not global integrity [11].

2.3.3 Host Identity Protocol (HIP) [77]

The Host Identity Protocol introduces a new namespace, the Host Identity. HIP maps an end-point identity to a host identity. In today's Internet, an IP address reflects the point of attachment of the host to the network. However, IP addresses are also used to identify hosts. The current IP-based naming architecture imposes security problems. IP addresses can easily be spoofed. Attackers can assume the identity of a victim by stealing its IP address as well. The current approach to solve this is the use of certificates by trusted authorities in conjunction with application layer encryption. HIP tries to solve these problems by decoupling the host identity from its network address and introducing a new name-space [78].

In HIP, hosts are identified by their Host Identifier (HI). The HI is also the host's public key. Since there are different public key algorithms that can be used with different key lengths, the HI is not good for using as a packet identifier or as an index into the various operational tables needed to support HIP [77]. A 128 bit long hashed version of it, the Host Identity Tag (HIT) is used to represent the HI at a protocol level. The HIT is used in the HIP payloads and to index the corresponding state in the end hosts. A third representation of the HI is the 32 bit long local scope identifier (LSI). It is designed to be able to replace IP addresses in the IP version 4 application programming interface (API). However, a host performing a HIP handshake may discover that the LSI formed from the peer's HIT collides with another LSI in use locally. In that case, the host should be able to handle the LSI collision locally such that application calls can be disambiguated. Transport protocols bind to the HIT or LSI rather than to an IP address. This enables the host to change its point of attachment to the network without terminating any ongoing transport connections. An HI or HIT is returned by a directory service together with one or multiple IP addresses at which the host might be reached. The HI (since it is a public key) is used to assure the authenticity of the host. This assumes that the returned HI has not been tampered with, e.g. by means of a secure directory service.

The base HIP exchange is utilized to establish state between an Initiator and a Responder. It consists of four packets. This four-packet design helps to make HIP DoS resilient. The Initiator first sends a trigger packet, I1, to the Responder. The second packet, R1, which contains a puzzle, starts the actual exchange. The puzzle is a cryptographic challenge that the Initiator must solve before continuing the exchange. The Initiator sends the solution in its reply, I2. The I2 message is discarded by the Responder if it does not contain a correct solution. The last three packets of the exchange, R1, I2, and R2, constitute a standard authenticated Diffie-Hellman key exchange. It should be noted, however, that both the Initiator's and the Responder's HITs are transported as such (in cleartext) in the packets, allowing an eavesdropper with a priori knowledge about the parties to verify their identities. Data packets start after the 4th packet. The 3rd and 4th HIP packets may carry a data payload in the future. HIs, HITs, and LSIs are not carried explicitly in the headers of user data packets. Instead, the IPsec Security Parameters Index (SPI), 32 bits, is used in data packets to index the right host context. The SPIs are selected during the HIP exchange. For user data packets, the combination of IPsec SPIs and IP addresses are used indirectly to identify the host context, thereby avoiding an additional explicit protocol header. Finally, HIP is designed as an end-to-end authentication and key establishment protocol [77].

SPIs are used in Encapsulating Security Payload (ESP) to find the right security association for received packets. The ESP SPIs have added significance when used with HIP; they are a compressed representation of the HITs in every packet. Since the SPI has significance at the receiver, only the $\prec$ DST, SPI $\succ$ tuple, where DST is a destination IP address, uniquely identifies the receiver HIT at every given point of time. The same SPI value may be used by several hosts at any point of time. A single $\prec$ DST, SPI $\succ$ value may denote different hosts at different points of time, depending on which host is currently reachable at DST. Each host selects for itself the SPI it wants to see in packets received from its peer. One method for SPI creation that avoids a replay attack is to concatenate the HIT with a 32 bit random or sequential number, hash this (using secure hash algorithm (SHA)-1), and then use the high order 32 bits as the SPI [77].

Since HIP datagrams are relatively large (at least 40 bytes), and ESP already has all of the functionality to maintain and protect state, the HIP payload is 'compressed' into an ESP payload after the HIP exchange [77]. In practice, HIP packets only occur in datagrams to establish or change HIP state.

HIP utilizes the cookie mechanism to protect the Responder from a number of denial-of-service threats. The Responder may delay state creation until receiving I2. Furthermore, the puzzle included in the cookie allows the Responder to use a fairly cheap calculation to check if the Initiator is "sincere" in the sense that it has used up CPU cycles in solving the puzzle. Moreover, the Responder may keep state about the received I1s, and match the received I2s against the state. This allows the Responder to avoid the computational cost of the hash function. However, this approach has the drawback of the requirement of creating state. Finally, the Responder may set the difficulty for the Initiator in the puzzle, based on its concern of trust of the Initiator.

In HIP, the cookie exchange starts when the Responder receives an I1. The Responder supplies a random number I, and requires the Initiator to find a number J. The Responder must select the number I in such a way that the Initiator cannot guess it in order to prevent replay attacks. The Initiator creates the concatenation of I, the HITs of the parties, and a guessed J, and takes a SHA-1 hash over this concatenation to find a proper J. If the lowest order K bits of the result (SHA-1 hash) is zeros, then J is found. The Initiator generates a number of Js until one produces the hash target of zero. The Initiator should give up after trying $2^{(K+2)}$ times, and start over the exchange [77].

In addition to the cookie mechanism, with the optional use of opaque data, the Responder may include some secret data in R1 that the Initiator must copy unmodified in the corresponding I2 packet. The Responder must change this secret periodically. Furthermore, Initiators are protected against R1 replays by a monotonically increasing "R1 generation counter" included in the R1. Finally, hosts are protected against replays to R2s and UPDATEs by use of a less expensive HMAC verification preceding HIP signature verification [77].

A HIP association between two hosts may be updated over time. Reasons for the update include the need to re-key expiring security associations, add new security associations, or change IP addresses associated with hosts. HIP provides a general purpose UPDATE packet, which can carry multiple HIP parameters. This packet can be used to update the HIP state between two peers [77].

Finally, HIP does not define how to use certificates. However, it does define a simple certificate transport mechanisms that may be used to implement certificate-based security policies.

In the HIP architecture, while IP addresses continue to act as locators, the HIs take the role of end-point identifiers [78]. Note that HIP maps HIs to IP addresses. In the HIP architecture, the Host Identity Layer between Layer 3 and Layer 4 translates HIs to IP addresses. Similarly, ARP translates IP addresses to link layer addresses. HIP solves the security problem related to the identification of the hosts/end-points by incorporating a public key into the identities. This provides end-to-end security.

However, HIP does not address all sorts of possible Layer 2 attacks. ARP spoofing can still be used to mislead packets since ARP works at a lower level unprotected by HIP. Nonetheless, due to the encryption facilities of HIP, man-in-the-middle attacks are prevented. One disadvantage of HIP is that it requires changes at the end hosts and a revised secure directory system. The concepts used in HIP may be extended to between Layer 2 and Layer 3 to securely identify a network device and bind MAC and IP addresses.

2.3.4 A Layered Naming Architecture for the Internet [31]

Balakrishnan *et. al.* [31] propose a new layered naming architecture for the Internet arguing that there should be three levels of naming and resolution in the Internet: from user-level descriptors to service identifiers, from service identifiers to endpoint identifiers, and from endpoint identifiers to IP addresses. These additional levels of naming and resolution allow service and data to be the first class of Internet objects. They also seamlessly accommodate mobility and multi-homing. Lastly, these levels enable integration of middleboxes into the Internet. In this architecture, it is also argued that flat names are a natural choice for the service and endpoint identifiers in contrast to hierarchal identification proposed in the previous frameworks (FQDNs, etc.).

In this approach, the naming framework and architecture rely heavily on existing proposals. For instance, the transport and networking layers are decoupled to address mobility, which is the same idea used in Nimrod [42] and HIP [77] as well. Other proposals and architectures used as inspiration are the Unmanaged Internet Protocol (UIP) proposal [55], the Internet Indirection Infrastructure (i3) [87], Semantic-Free Referencing (SFR) [89], and Distributed Hash Tables (DHTs) [30].

In this study, a new layered naming system is proposed to address architectural problems in the current Internet. This system has four layers: user-level descriptors, such as search keywords, e-mail addresses, etc.; service identifiers (SIDs); endpoint identifiers (EIDs); and IP addresses or other forwarding directives. In addition, this study describes four general design principles about the nature and use of names.

The first design principle addresses the role of names in protocols.

> "*Principle #1: Names should bind protocols only to the relevant aspects of the underlying structure; binding protocols to irrelevant details unnecessarily limits flexibility and functionality.*"

This first principle is often violated in today's architecture. For instance, today's DNS-based names for services and data force applications to resolve service and data names down to an IP address. This binds the application request to a particular end host, to a network location, as expressed by an IP address. This violation is rectified by the introduction of two new naming layers. First, in order to enable applications to refer services with persistent names (that are not tied to endpoints) service identifiers (SIDs) are used as a new class of names. Second, topologically independent endpoint identifiers (EIDs) are utilized to identify hosts.

The idea of a topologically independent EID that uniquely identifies a host is utilized so that transport protocols could refer to endpoints in a manner independent of their IP addresses or network topology. Note that the two mechanisms SIDs and EIDs are logically distinct and need not be coupled.

These two new naming layers require two additional layers of name resolution: from SIDs to EIDs and from EIDs to IP addresses. For instance, in order to interact with a service the application initiates a communication session whose destination is named by the service's SID. When an application resolves that SID, it gets one or more EIDs that identify the end hosts that run the service. The session will typically involve one or more transport-layer connections between the client and service EIDs. The transport layer resolves the EID to the current set of IP addresses to which the EID is attached. More concretely, applications generally deal with SIDs, transport protocols generally deal with (or bind to) EIDs, and only the network layer deals with IP addresses [31].

The second design principle discusses how names should relate to their referents. When users care about the identity of an object rather than its location, the object's name should be persistent (it remains unchanged even when the object's location changes).

> "*Principle #2: Names, if they are to be persistent, should not impose arbitrary restrictions on the elements to which they refer.*"

The two current global namespaces, IP addresses and DNS names, are each closely tied to an underlying structure. Achieving scalable routing requires that IP addresses reflect network topology. DNS names, though more flexible, nonetheless reflect administrative structure. DNS-based names for data are inherently ephemeral. Data is likely to be replicated on, or moved to, hosts outside the originating domain. When this happens, existing references to the data become invalid. A similar problem arises when services move and there are pre-existing pointers to those services. Currently, there is no namespace that can persistently name data and services [31].

Since a flat namespace has no inherent structure and does not impose any restrictions on referenced elements, a flat namespace for SIDs and EIDs is adopted to ensure compliance with the second principle.

The third design principle addresses how these names are resolved. Considering that the typical definition of "resolving a name" is mapping a name to its underlying "location," an SID's "location" would usually be an EID, transport, and port triple; and an EID's location would be an IP address. However, since this definition is very restrictive, the following principle is adopted.

> "*Principle #3: A network entity should be able to direct resolutions of its name not only to its own location, but also to the locations or names of chosen delegates.*"

In any logical network connection, the initiator at any level intends to connect to a destination entity. However, the destination entity may not want to handle the connection directly, preferring instead to direct the connection to a chosen delegate. Delegation allows the architecture to gracefully incorporate intermediaries, which are defined as cleaner and more flexible versions of middleboxes. Moreover, delegation provides some protection against denial-of-service attacks.

The fourth principle adopts the idea of source routing in which sources are given the power to specify the path or, in the case of loose source routing, a few points along the path. This ability should be available at the endpoint and service layers as well. More specifically, the abstraction of sending to a destination should be generalized to allow sources to indicate that their packets should traverse a series of endpoints (specified by a sequence of EIDs) or that their communications traverse a series of services (specified by a sequence of SIDs) [31]. Since these various destinations are specified at the endpoint and service layers, these intermediate points may act on the packets in non-trivial ways. Combining this sentiment with Principle #3 suggests that endpoints and services should be able to have their names resolve not just to a single location but more generally to a sequence of identifiers [31]. This idea is stated in the forth design principle.

> "*Principle #4: Destinations, as specified by sources and also by the resolution of SIDs and EIDs, should be generalizable to sequences of destinations.*"

It is claimed that these general principles lead us to the following: (1) two additional sets of names (SIDs and EIDs) should exist, (2) these names should be flat, (3) the architecture should support delegation as a basic primitive, and (4) destinations, whether specified by the source or receiver, can, in fact, be sequences of destinations.

Since applications bind to SIDs and transport protocols bind to EIDs, applications must use a layer between them and transport layer that resolves SIDs to EIDs, and similarly, transport protocols must use a layer between them and IP layer that translates between EIDs and IP addresses. These layers could be separate libraries or software that is part of the application or transport protocol.

These two name resolutions provide a couple of benefits. First, naming data and services with SIDs overcomes the problems of using DNS-based URLs for that purpose. Second, naming endpoints with EIDs provides natural solutions to mobility and multi-homing.

It is expected that EIDs are carried in packets to identify the packet's logical endpoint. The case of SIDs is similar. SIDs name services or data. Therefore, the SID must often be carried in band, like the EID. However, the SID is not required to be in every packet but rather in each *logical piece of data* being communicated between the sender and the recipient.

In this architecture, one could utilize EID-level delegation to provide some protection against denial-of-service (DoS) attacks. A server could shield itself from attackers by placing a forwarding intermediary between itself and distrusted clients and by installing traffic filters at the forwarding intermediary. Nonetheless, an attacker could launch a DoS attack by sending packets directly to a server's IP address. This architecture cannot prevent this type of denial of service attacks since it does not modify current routers. However, having all incoming packets directed through the same intermediary may simplify router-level packet filtering.

Another feature of this architecture is flat names. Flat names have two disadvantages: they are hard to resolve and are not human-readable. It is believed that DNS achieves scalability through hierarchy. However, the introduction of Distributed Hash Tables (DHTs) [30] suggests that flat namespaces can be scalably resolved with a resilient, self-organizing, and extensible distributed infrastructure. Since DHTs are proposed in the context of peer-to-peer (P2P) systems, an unmanaged and distrusted P2P system may not be suitable for a crucial piece of the Internet infrastructure.

Instead, a well-managed, distributed collection of machines should provide the name resolution service using a DHT or other flat namespace resolution algorithms. Furthermore, in DNS, each name's uniqueness is ensured by the related authority. This is harder to achieve with flat names. Ensuring that no one else changes the resolution of an entity's name is challenging with flat names as well. Moreover, DHTs' typical resolution time in comparison to DNS is unacceptable for most name resolutions. Since DNS often returns results from a local name server, the latency may be very small. There are two approaches to address this latency issue in DHTs. First, it is possible to use DHT-style routing algorithms designed or use caching with better performance than regular DHTs. Second, it is possible to design a DHT-based resolution algorithm using local proxies, local replication, or two layered resolvers that enable hosts within a local network to find local instances of entries written from within the network. These schemes also provide fate-sharing in that if an organization is disconnected from the rest of the Internet, its hosts can still gain access to entries written locally [31]. Another advantage of the DNS infrastructure is that it has a built-in economic and trust model: domains provide their own name servers. The central facilities required (the root servers) are minimal and inexpensive. In the proposed resolution infrastructure names are stored at essentially random nodes. This model raises the questions of who will pay and why users should trust the infrastructure. It is envisioned that in future Resolution Service Providers (RSPs) form a competitive yet cooperating commercial market much like current ISPs. Customers could pay for lookups and for storing, likely a flat fee for a reasonable number of accesses. The various RSPs would have mutual "peering" relationships to exchange updates, much as the tier-1 ISPs all interconnect today. These RSPs would have incentives to process requests honestly since each RSP would be judged by how well it served its customers [31].

The other issue with a flat namespace is that it is highly versatile but provides no user-readable hints. Difficulty arises when dealing with data and services for which the human-readability of URLs has been crucial. The first question needs to be answered is how users obtain a SID. Users often find URLs through search engines rather than directly typing them into a browser. In this new architecture, search engines could continue to perform the same function to identify services and data by SIDs. Moreover, third-parties could offer directory services mapping human-readable *canonical names* to SIDs. The advantage of these canonical names is that they are not part of the infrastructure and thus can be offered by multiple competing entities [31]. In terms of security, users need some assurance that the SID they have in hand points to the intended target. For instance, a URL like http://www.nytimes.com provides hints about its target but an opaque bit string gives no such assurance. To remedy this, bit strings may be accompanied by meta-data including cryptographic statements like "Authority A says that this SID points to the newspaper New York Times." Again, authorities like Authority A would not be part of the resolution infrastructure but instead part of a competitive market of SID authenticators. In addition, users may verify the output of the resolution step using the embedded cryptographic meaning in the identifiers.

In the layered naming architecture for the Internet proposal, some security issues exist at both application and network levels. At the network level, decoupling location and identity means that using IP routing to send a packet to a given *location* (via IP) no longer means that the packet is going to the host with the intended *identity* (EID) [31]. On the other hand, because EIDs are flat, they can hold cryptographic meanings. Two communicating parties, given each other's identifiers, may authenticate each other, which is not possible by using only IP addresses. At the application level, one cannot tell by looking at a SID whether it corresponds to one's "intended" target. However, it is possible to achieve stronger security properties since one's computer can tell by looking at a SID or EID whether the accompanying meta-data is correct.

In summary, the layered naming architecture for the Internet borrows heavily from three projects - HIP [77], SFR [89], and i3 [87] - and can be seen as synthesizing these works, each of which has a narrower goal, into a larger whole. This architecture tries to accommodate middleboxes and proposes a flat name space for data and services. It also describes the binding between an identifier (SID or EID) and an IP address. In addition, it does not require significant modifications to core network elements. However, incorporating the new naming layers requires significant changes to host software, both applications and protocols. Moreover, resolving these flat names requires a new resolution infrastructure. Three significant messages are given in the proposal. The first message is that DHTs allow us to contemplate using flat namespaces in a network architecture. Once a flat namespace is established, it can be used to name anything. The second message is that the extra naming layers help to shield applications from the underlying routers. This layered naming architecture binds to IP addresses only at the lowest logical layer, thereby minimizing the extent to which the routing infrastructure constrains the protocols and applications above. The last message, the concept of *delegation* allows interposition without violating the spirit of the end-to-end principle or the semantics of IP.

2.3.5 Cryptographically Generated Addresses (CGAs)

In the Internet architecture, additional namespaces, such as end-host identities, service identifiers, and content identifiers, are needed to decouple host identities, services, and content from locations or hosts. However, there is also need for binding identities or security parameters to IP addresses to verify the authenticity of messages received from addresses or hosts. In [27], a method is described for securely associating a cryptographic public key with an IP version 6 (IPv6) address in the Secure Neighbor Discovery (SEND) protocol. In this method, a cryptographic one-way hash of a public key and auxiliary parameters is computed to generate the interface identifier (i.e., the rightmost 64 bits) of an IPv6 address. The resulting IPv6 address is called Cryptographically Generated Address (CGA). In addition, the corresponding private key may be used to sign messages sent from an address and to verify the association between a public key and a CGA.

The main advantage of this method is that no additional security infrastructure, such as a public key infrastructure (PKI), certification authorities, or other trusted servers, is needed. A host can prove the ownership of a CGA by proving that it knows the private key associated with the CGA. Moreover, any node can verify a signature if it knows the CGA and the associated public key. Nonetheless, since CGAs are not certified, an attacker can create a new CGA using a subnet prefix and anyone's public key. However, the attacker also needs to know the corresponding private key to assert its ownership of the CGA. For instance, to use another host's CGA and send messages that appear to originate from that host, the attacker needs to find another public key that produces the same hash as the host's public key (hence, the same CGA).

An IPv6 address is 128 bits long. The leftmost 64 bits of an IPv6 address is the subnet prefix, which is used to route packets in the Internet. The rightmost 64 bits of an IPv6 address is the interface identifier. The interface identifier is utilized to identify a node in a local network. Hence, an interface identifier is required to be unique within a subnet prefix. The sixth and the seventh bits (from the left) of an interface identifier are the Universal/Local bit , "u", and the Individual/Group bit, "g", respectively. In the Modified EUI-64 format, the "u" bit is set to one to indicate global (universal) scope.

In the CGA generation process, when computing an interface identifier, bits 0, 1, 2, 6 ("u" bit), and 7 ("g" bit) are ignored. Note that bit 0 represents the first bit on the left. The three leftmost bits (i.e. bits 0, 1, and 2) of an interface identifier is the security parameter (*Sec*) as presented in (1). The *Sec* of a CGA determines the

CGA's strength against brute-force attacks.

$$Sec = (\text{interface identifier } \& \text{ 0xe000000000000000}) \gg 61 \qquad (1)$$

A CGA is generated from a set of parameters that consist of a public key and auxiliary parameters. Two hash values are computed with the Secure Hash Algorithm (SHA-1) [1] from these parameters. The first hash value, *Hash1*, equals the interface identifier of the CGA address. The second hash value, *Hash2*, must have its $16 * Sec$ leftmost bits equal to zero. In addition, a *collision count*, an 8-bit unsigned integer, is utilized to recover from an address collision detected during the process. Moreover, a *modifier*, which is a 128-bit unsigned integer, is used in the CGA generation process to implement hash extensions and to enhance privacy. Finally, the generation process takes three input values: a 64-bit subnet prefix, the public key of the address owner, which is formatted as a DER-encoded [4] ASN.1 structure of the type SubjectPublicKeyInfo, defined in the Internet X.509 certificate profile [69], and the security parameter *Sec*, which is an unsigned 3-bit integer [27].

A CGA address can be generated as follows [27]:

1. Set the modifier to a random or pseudo-random 128-bit value.

2. Concatenate the modifier, 9 zero octets, the encoded public key, and any optional extension (variable length) fields. Execute the SHA-1 algorithm on the concatenation. Obtain Hash2 by taking the 112 leftmost bits of the SHA-1 hash value.

3. Compare the $16 * Sec$ leftmost bits of Hash2 with zero. If they are all zero or if *Sec* is set to zero, go to step 4. Otherwise, increment the modifier by one and go to step 2.

4. Set the 8-bit collision count to zero.

5. Concatenate the final modifier value, the subnet prefix, the collision count, the encoded public key, and any optional extension fields. Execute the SHA-1 algorithm on the concatenation. Obtain Hash1 by taking the 64 leftmost bits of the SHA-1 hash value.

6. Form an interface identifier from Hash1 by writing the value of *Sec* into the three leftmost bits (i.e. bits 0, 1, and 2) and by setting the u and g bits (i.e. bits 6 and 7) to zero as presented in Figure 3.

7. Form a 128-bit IPv6 address by concatenating the 64-bit subnet prefix to the left and the 64-bit interface identifier to the right.

8. Perform duplicate address detection if required. If an address collision is detected, increment the collision count by one and go to step 5. However, after three collisions, stop and report the error.

The output of CGA generation algorithm is a new CGA address and the auxiliary parameters (the modifier, the subnet prefix, and the collision count). A node may verify a CGA address with the CGA verification algorithm as outlined in [27]. CGA verification takes an IPv6 address and the auxiliary parameters, which consist of the modifier, the subnet prefix, the collision count, the public key, and the optional extensions field, as input.

The randomly chosen initial value of the modifier in the step 1 makes addresses generated from the same public key unlinkable and enhances privacy. If the subnet prefix of an address changes, the old value of the modifier with the same public key can be reused to generate a new identifier. However, this optimization may, in some cases, make it easier for an observer to link two addresses to each other. In addition, for *Sec* values greater than zero, the CGA algorithm is not guaranteed to terminate after a certain number of iterations.

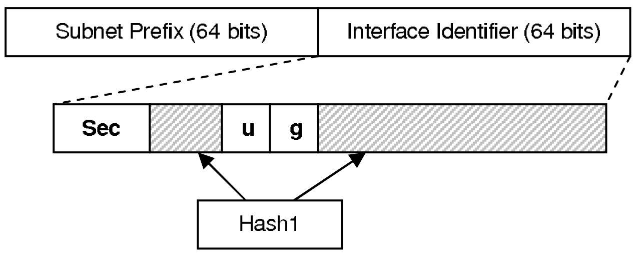

Figure 3: The Cryptographically Generated Address (CGA) format.

The main goal of CGAs is to prevent stealing and spoofing of existing IPv6 addresses. Since the public key of the CGA owner is cryptographically bound to the address, the CGA owner can use the corresponding private key to assert its ownership and to sign SEND [24] messages sent from the address. However, one of the limitations of the CGA is that there is no mechanism for proving that an IPv6 address is not a CGA. Moreover, a CGA-based authentication neither verifies that a node with the authenticated CGA exists nor that the node is permitted to use the specific subnet prefix. Furthermore, an attacker may create a CGA utilizing someone else's public key and replay messages signed by the real owner of the public key.

CGAs can provide some level of pseudonymity. A node may generate multiple pseudo-random CGAs by executing the CGA generation algorithm utilizing a different random or pseudo-random initial value for the modifier each time. In addition, to provide privacy protection, the (pseudo)random number generator used in the address generation process should be able to produce unpredictable and unlinkable values. Nonetheless, since the public key of the address owner is revealed in the signed SEND messages, the address owner should generate not only new addresses but also new public keys. Then again, a node's link-layer address, such a MAC address, is a unique identifier for the node. As long as the node has the same link-layer address, changing the public key for privacy reasons does not help. In our new data link layer security architecture, the issue of the same link layer address constraint is solved as we utilize secure identifiers instead of fixed MAC addresses. In fact, a secure identifier generated for the data link layer may be directly used as an interface identifier of an IPv6 address. However, our data link layer address generation process differs in two aspects. First, we verify the identity of a host. Second, we ensure the privacy of the host (pseudonymity) by hiding the identity of a host from observers in the generation process.

2.3.6 Anonymous Identifiers (ALIEN)

Recently, there has been some interest in anonymous identifiers, especially among mobility and network security experts in the Internet community [7, 25, 63, 62]. In the 62nd IETF meetings, the Anonymous Identifiers (ALIEN) charter held a BoF meeting in Paris, France. One of the main goals of the meeting was to consider possible approaches whether to form an ALIEN research or working group. A research group helps communities to understand the problem in a large scope whereas a working group focuses on implementation guidelines with a very narrow scope. Another consideration was whether to identify the modifications needed in existing internetworking protocols to improve their privacy properties. Although a research group was not established at this meeting, they are likely to charter a research group at the Internet Research Task Force (IRTF) [7]. We believe that it is valuable to understand the concepts discussed in this meeting because of their relevance to our data link layer identifiers and the security inter-layering concept for a flexible data link layer security architecture.

Although terminologies such as anonymity, pseudonymity, unlinkability, and privacy are related to each other, they have distinct implications. According to [63], anonymity ensures that a user may use a resource or service without disclosing the user's identity. In addition, any two anonymous acts performed by the same user may not be linked back to the user. Similarly, pseudonymity ensures that a user may use a resource or service without disclosing its user identity. However, the user can still be accountable for that use. In another related concept, if a user utilizes resources or services without allowing others to associate these uses, these events/uses are described as unlinkable. Note that unlinkability is not necessary for anonymity. Finally, privacy allows individuals, groups, and institutions to determine for themselves, when, how, and to what extent information about them is communicated to others. In essence, privacy is a more general term than anonymity [63].

In the data link layer, MAC addresses are used as device identifiers. To provide anonymity, a user's identity and the respective identifier should not be linked. For instance, a MAC address or an IP address should not be associated with a user. Since one of our objectives is to address the weak link between the network and data link layers, we are proposing to link upper layer identities to data link layer identifiers. This objective contradicts with anonymity. Nonetheless, we can provide pseudonymity in view of our objectives. Moreover, there are tradeoffs between anonymity and identifiability in a data link layer security architecture. If hosts need to be authorized to receive services in a LAN, hosts/users/devices may be required to prove their identities. In addition, our security inter-layering concept may or may not offer unlinkability depending on how data link layer identifiers are generated. For instance, generating data link layer identifiers by simply hashing upper layer identities does not provide unlinkability. On the other hand, if the identifiers are generated using hash functions that utilize pseudo-random numbers, unlinkability may be ensured.

Privacy issues are complicated in the Internet protocols and architectures. Privacy may have different definitions in different contexts. For example, in mobile environments, adversaries may locate, identify, and track targets violating the privacy of the targets by simply capturing the targets' MAC addresses. As the adversaries gather more information regarding their targets, they can build profiles, perform attacks, and misuse the information etc [62]. However, since we utilize data link layer identifiers instead of MAC addresses, this type of privacy issues may be easily prevented in wired and wireless LANs. Then again, we should not ignore unlinkability in connection with privacy. Packets/frames carry multiple identifiers for various layers. Whenever temporary data link layer identifiers are changed, other identifier sets across layers need to be replaced to avoid linking identifiers/identities [63]. Our new security inter-layering concept may provide a solution to this problem. There are also other considerations such as visibility of identifiers in a link or path, the relationship between topological and geographical locations, ability to track misbehaving devices or malicious nodes, and identifying hosts or protecting identities based on the requirements of networks and environments etc. It is obvious that addressing privacy issues only in the data link layer is not going to be sufficient. A solution which addresses privacy, anonymity, and unlinkability requires the consideration of all layers. While our objective is not to address these issues directly, our new data link layer security architecture with the security inter-layering concept will accommodate the future solutions to these problems.

2.3.7 Wireless Local Area Network Security (IEEE 802.11i)

The IEEE 802.11 (wireless LAN medium access control (MAC) and physical layer (PHY) specifications) defines an optional Wired Equivalent Privacy (WEP) protocol, which is based on a stream cipher RC4 encryption algorithm. The objective of the WEP in wireless networks was to provide comparable confidentiality to a traditional wired network. However, studies have shown that the mechanisms used in the IEEE 802.11 are insecure [16, 21, 22, 38, 47, 51, 54, 57, 88]. To improve security in wireless LANs, the IEEE 802.11i, an amendment of MAC security enhancements for the IEEE 802.11 standard, was ratified in 2004. The IEEE 802.11i, also known as Wi-Fi Protected Access 2 (WPA2), introduces a new security architecture called Robust Security Network (RSN). In this amendment, the proposed improvements to the 802.11 architecture focus on two areas: the IEEE 802.1X standard (EAPOL) and Advanced

Encryption Standard (AES), for access control and encryption, respectively. The new standard also includes enhancements to increase the security of the existing hardware (pre-RSN) with software upgrades and defines a Transient Security Network (TSN) allowing both RSN and WEP systems operate in parallel. In the IEEE 802.11i, RSN defines two data privacy protocols: Temporal Key Integrity Protocol (TKIP) for pre-RSN WLAN hardware and AES-based Counter-Mode/Cipher Block Chaining Message Authentication Code (CBC-MAC) protocol (CCMP). TKIP, the next generation of WEP, was adopted by Wi-Fi Alliance ahead of the 802.11i standard. This subset of the RSN architecture is also called WPA [14, 51].

In RSN, the beginning of the process of establishing network connectivity is very similar to 802.11. A station (STA) first authenticates itself to an access point (AP) using open-system (NULL) authentication, and then associates with the AP. During the association phase, an STA sends an association request message. This message is also used to identify the capabilities of the STA to the AP. If these capabilities are acceptable, the AP sends an association response message indicating success. However, unlike 802.11, when the AP replies with the message indicating success, the STA does not become eligible to send/receive data to/from the network. The STA should follow the 802.1X authentication or pre-shared key method, and the four-way handshake protocol before it can start sending or receiving data in the RSN. Figure 4 illustrates an overview of this process in the infrastructure mode, where each client or station communicates to an access point.

2.3.7.1 Access Control

An RSN is capable of supporting two different models of operation: the IEEE 802.1X authentication and a pre-shared key for key establishment. In RSN, the IEEE 802.1X authentication method requires an upper-layer authentication process to generate

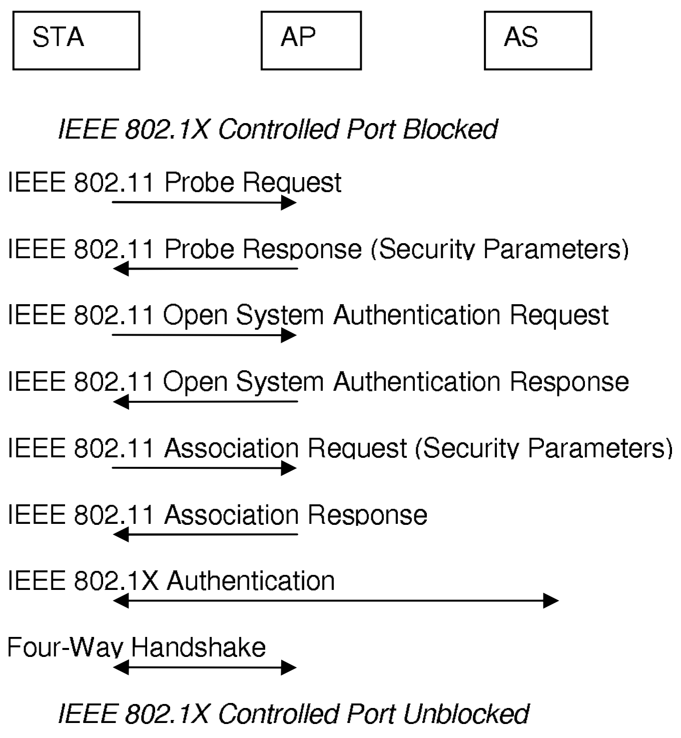
STA
AP
AS
IEEE 802.1X Controlled Port Blocked
IEEE 802.11 Probe Request
IEEE 802.11 Probe Response (Security Parameters)
IEEE 802.11 Open System Authentication Request
IEEE 802.11 Open System Authentication Response
IEEE 802.11 Association Request (Security Parameters)
IEEE 802.11 Association Response
IEEE 802.1X Authentication
Four-Way Handshake
IEEE 802.1X Controlled Port Unblocked

Figure 4: Communications that take place in the Robust Security Network (RSN) architecture in the infrastructure mode to move from the controlled port blocked state to the controlled port unblocked state [5].

matching keys (Pair-wise Master Keys (PMKs)) securely at an STA and an authentication server. The generated PMK is provided to the AP separately as well. On the other hand, the pre-shared key method does not utilize an upper-layer authentication process. As the name suggests, it uses a Pre-Shared Key (PSK) installed in advance in the STA and the AP. In this method, the authenticity of the parties is verified by proving possession of the key. In addition, the RSN architecture uses an Information Element (IE) to negotiate the type of security in a WLAN. The IE, broadcasted in an AP's beacon, identifies the authentication model (the pre-shared key or the authentication server model), a list of the supported pair-wise key mechanisms, and the group security model utilized by the AP.

The IEEE 802.1X (Standard for Port-Based Network Access Control) [3] which defines the Extensible Authentication Protocol over LANs (EAPOL), uses an authentication server to authenticate each user on the network. The purpose of this standard is to implement access control at the point which a user/host joins the network. The IEEE 802.1X is composed of three entities: a supplicant, an authenticator, and an authentication server (authorizer). A supplicant authenticates via an authenticator to an authentication server (AS) to obtain services. The authenticator allows EAP traffic before the authentication process completes by utilizing a dual-port model which consists of a uncontrolled port and a controlled port. The uncontrolled port only accepts the IEEE 802.1X packets regardless of the authorization state, whereas the controlled port accepts packets from authenticated devices (for more information, see Section 2.3.8).

In the IEEE 802.11i context, a supplicant represents a station (STA), and an authenticator represents an access point (AP). The AS can be a separate sever (e.g. RADIUS server, etc.) or built into the AP. The AS makes a decision whether to admit or block the STA and informs both the AP and STA. In the IEEE 802.1X authentication process, if an EAP Success message is delivered to the STA, then

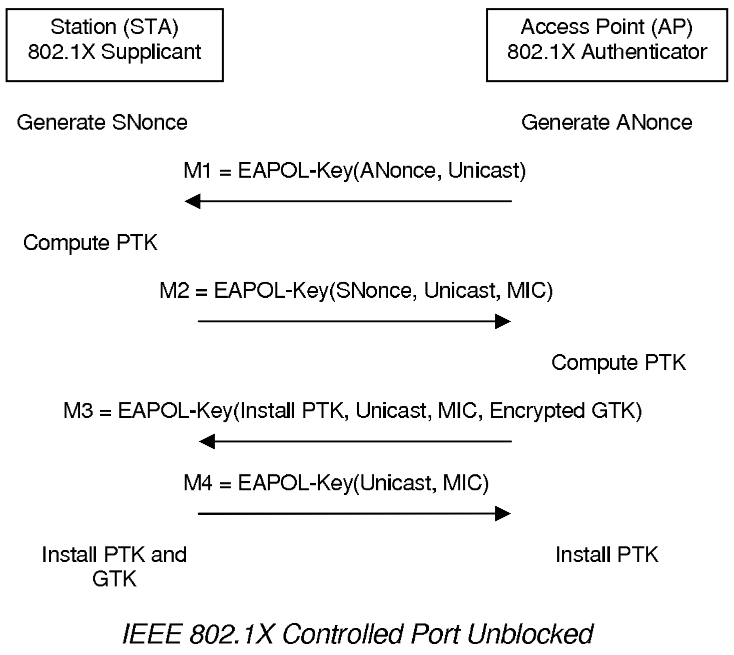
Station (STA)
802.1X Supplicant
Access Point (AP)
802.1X Authenticator
Generate SNonce
Generate ANonce
M1 = EAPOL-Key(ANonce, Unicast)
Compute PTK
M2 = EAPOL-Key(SNonce, Unicast, MIC)
Compute PTK
M3 = EAPOL-Key(Install PTK, Unicast, MIC, Encrypted GTK)
M4 = EAPOL-Key(Unicast, MIC)
Install PTK and GTK
Install PTK
IEEE 802.1X Controlled Port Unblocked

Figure 5: The four-way handshake protocol to establish pair-wise and group keys [5].

the AP can start a four-way handshake protocol. At the end of the IEEE 802.1X EAP authentication process (between the STA and the AS), both the STA and the AS possess a fresh PMK. The AS provides a copy of the key to the AP in a secure fashion. If, instead of the upper-layer authentication, the pre-shared key method is used, the PSK becomes the PMK.

In an RSN, the STA and AP must mutually authenticate each other before the STA establishes the network connectivity. Until the mutual authentication, the four-way handshake protocol, is completed, the controlled port at the AP does not accept any data packets. After the four-way handshake protocol, the mutual authentication completes and the AP starts providing services to the STA via a secure channel by enabling the IEEE 802.1X controlled port as illustrated in Figure 5 [5].

2.3.7.2 Key Hierarchy

In RSN, keys used for data privacy have a limited lifetime. RSN uses a pair-wise key for unicast traffic and a group key for multicast traffic. There are many keys used in an RSN forming a key hierarchy. The key hierarchy starts with a PMK. If the pre-shared key method is used, the PSK is the PMK. The PMK is not directly used to provide data privacy. It is used for mutual authentication of an STA and an AP, as well as for deriving the other keys in the four-way handshake protocol. A Pair-wise Transient Key (PTK), derived from the PMK, is used for data privacy and integrity. The PTK represents a set of keys derived in the four-way handshake: the Temporal Key (TK), the EAPOL-Key Encryption Key (KEK), and the EAPOL-Key Confirmation Key (KCK). These keys are never-before-used per-link keys. Every time an STA tries to associate to an AP they are re-computed. When an STA and an AP establish a fresh PTK, the AP uses it to deliver a Group Transient Key (GTK) securely to the STA. In addition, the AP utilizes a group key handshake protocol to deliver the subsequent GTKs to the STAs. After the four-way handshake, both the AP and the STA allow general the IEEE 802.11 data packets to flow. Both the four-way handshake protocol and the group key handshake protocol utilize EAPOL-Key frames.

In our data link layer security architecture, we utilize IEEE802.1X concepts for our access control. Additionally, we require hosts to generate data link layer identifiers before providing network access. Finally we utilize a similar key hierarchy for future compatibility of wired and wireless networks.

2.3.8 The Extensible Authentication Protocol over LANs (EAPOL)

The purpose of the IEEE 802.1X Standard for Port-Based Network Access Control [3] is to implement access control at the point at which a user joins the network. The standard defines the Extensible Authentication Protocol over LANs (EAPOL), which uses an authentication server to authenticate each user on the network. Extensible Authentication Protocol (EAP), defined by the RFC 2284 [37], is an extension to the Point-to-Point Protocol (PPP). PPP is most commonly used for dial-up Internet access, to transmit IP packets between a workstation or PC and an ISP.

IEEE 802.1X is composed of three entities: a *supplicant*, which is an end user system seeking to access to the network, an *authenticator*, which controls access to the network, and an *authentication server (authorizer)*, which authenticates the end user, negotiates key materials with the end user, and controls access to the network via authenticator. A supplicant authenticates via an authenticator to an authentication server. Specifically, an authenticator represents the network port at which a supplicant connects to the network. If the authentication sever confirms the supplicant's credentials, it directs the authenticator to provide services.

As illustrated in Figure 6, EAP over LAN (EAPOL) is used between a supplicant and authenticator to convey EAP packets while authentication dialog is being carried in EAP frames between the supplicant and authentication server. The authenticator and authentication server communicate using a higher layer protocol (RADUIS, TLS, etc.). The authenticator must allow EAP traffic before authentication process completes. The authenticator also relays the encapsulated EAP messages generated by the authentication server to the supplicant [51]. However, until the authentication process is completed, regular data traffic is blocked. This is achieved by a dual-port model. In IEEE 802.1X, an authenticator has two ports of access to the network: uncontrolled port, and controlled port. Figure 7 illustrates the operation of port-based access control in an authenticator. The uncontrolled port only accepts the IEEE 802.1X packets regardless of the authorization state, whereas the controlled port accepts packets from authenticated devices. The uncontrolled and controlled ports are considered to be part of the same point of attachment to the LAN. Any packet received on the physical port is made available at both the controlled and uncontrolled ports.

IEEE 802.1X may utilize EAP to support a variety of authentication schemes, including smart cards, Transport Layer Security (TLS), Kerberos, Public Key, One

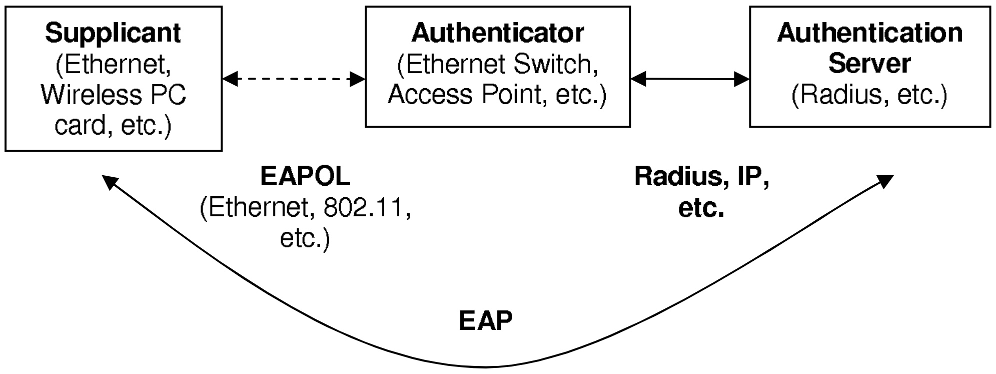

Figure 6: The IEEE 802.1X setup.

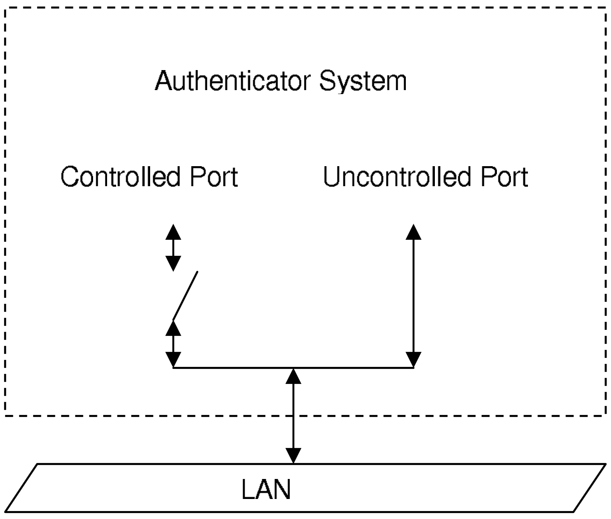

Figure 7: Port-based access control in the IEEE 802.1X.

Time Passwords, etc. EAP employs four types of messages: EAP Request, EAP Response, EAP Success, and EAP Failure. Any authentication mechanism can be encapsulated within the EAP Request/Response messages [51]. For instance, the TLS over EAP takes the advantage of the protected ciphersuite negotiation, mutual authentication, and key management capabilities of the TLS protocol [33]. Specifically, the PPP EAP TLS authentication protocol defines how to utilize a series of EAP Request/Response frames to perform the TLS handshake over EAP. The authentication method in each EAP message is identified by a Type field. Moreover, any packet exceeding the maximum size of an EAP message is fragmented and sent in several exchanges.

In 2004, the IETF EAP working group has started a new effort to standardize and improve the EAP authentication algorithms by adding new methods such as EAP-SIM, EAP-AKA, EAP-IKEv4, EAP-Double-TLS, etc. For example, EAP-SIM specifies an EAP mechanism for authentication and session key distribution using the Global System for Mobile Communications (GSM) Subscriber Identity Module (SIM) [65]. In EAP-SIM, multiple authentication triplets are combined to create authentication responses and session keys of greater strength than the individual GSM triplets. Moreover, EAP-SIM includes network authentication and optional user anonymity support. GSM subscribers are identified with the International Mobile Subscriber Identity (IMSI). Since the IMSI of a subscriber is sent in clear text in each authentication, observers may easily track the user. In EAP-SIM, EAP servers are allowed to generate pseudonym usernames to provide identity privacy. Another EAP authentication method, EAP-Double-TLS extends the full TLS handshake, used to mutually authenticate a peer and server, by establishing a secure connection based on the use of Pre-Shared Keys (PSK) [29]. EAP-Double-TLS is composed of two phases. In the first phase, a Shared-key handshake is used for mutual authentication and cryptographic key generation. In the second phase, a full TLS handshake with mutual authentication, only server-side authentication, or anonymous authentication is performed. This method allows identity protection and the EAP Success/Fail message protection which is not provided in EAP-TLS. EAP-AKA, on the other hand, defines an EAP mechanism for authentication and session key distribution using the Authentication and Key Agreement (AKA) mechanism employed in the 3rd generation mobile networks Universal Mobile Telecommunications System (UMTS) and CDMA2000. The EAP-AKA allows the use of the third generation mobile network authentication infrastructure in the context of wireless LANs. Moreover, it allows the use of the AKA also as a secure PPP authentication method in devices that already contain an identity module [23].

One of the advantages of utilizing EAP in LANs is that no IP address is needed before being granted access to a network. We utilize the IEEE 802.1X access control with an EAP authentication method in our new data link layer security architecture. However, in our data link layer security architecture, hosts are expected to generate data link layer identifiers as well. Each host needs to identify itself to obtain an identifier. Then again, a host, which does not have any type of identifier (Layer 2 or Layer 3), needs an identifier to communicate in the first place. This is analogous to the question of "Which came first, the chicken or the egg?". The problem can be solved by utilizing temporary identifiers. Nevertheless, since our objectives include secure data link layer identifier generation, alleviation of DoS attacks, and identity protection during the authentication process, we are not able to directly use any of the EAP methods as they exist at present.

The literature review and analysis of related work yielded a large quantity of relevant information. Figure 8 shows an overview summary of the related work and indicates that concepts and ideas from these related efforts have been modified and synthesized into our new data link layer architecture with security inter-layering.

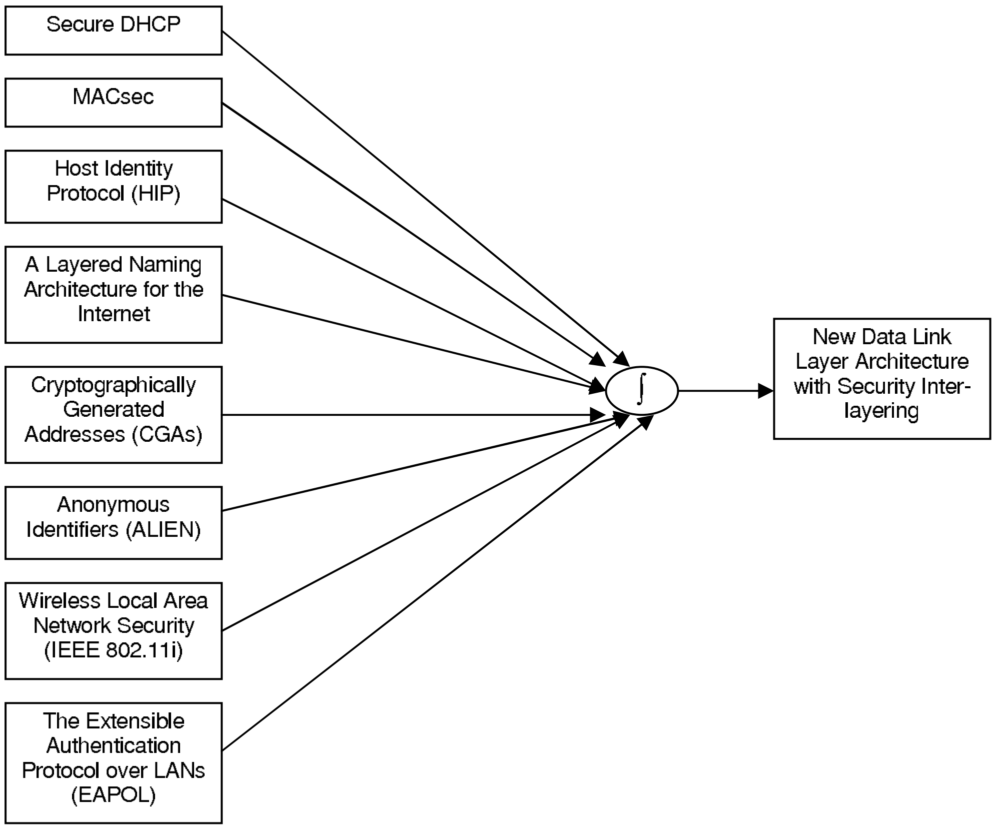
Secure DHCP
MACsec
Host Identity Protocol (HIP)
A Layered Naming Architecture for the Internet
Cryptographically Generated Addresses (CGAs)
Anonymous Identifiers (ALIEN)
Wireless Local Area Network Security (IEEE 802.11i)
The Extensible Authentication Protocol over LANs (EAPOL)
∫
New Data Link Layer Architecture with Security Inter-layering

Figure 8: Summary overview of the impact of related work on our new data link layer architecture with security inter-layering.

CHAPTER 3

ARCHITECTURAL FRAMEWORK

In this overview chapter, we first discuss possible approaches to address the open system interconnection (OSI) Layer 2 and Layer 3 binding problem identified in Chapter 2. Second, we introduce a new security inter-layering concept. We also provide a discussion of the relationship between layers/namespaces and security. Third, we present an overall diagram of our framework architecture.

3.1 Addressing the Layer 2 and Layer 3 Binding Problem

To address the weak link between Layer 2 and Layer 3, one solution is to utilize a new naming architecture instead of IP and MAC addresses in networks. However, in practice, it is difficult to implement a new naming architecture instead of IP addresses in the current Internet without creating a service disruption. It is possible to use cryptographic identities for network devices to secure identities without directly affecting the current IP naming architecture. This approach also separates identities from locations. For instance, HIP is such an architecture. HIP provides cryptographic identifications for end-points/hosts. However, HIP is designed to work between Layer 3 and 4 in the OSI reference model. This type of identification may be carried into Layer 2 as well [19]. A public/private key pair may be used to identify machines instead of MAC addresses. For example, a MAC address could be a hash of a public key at Layer 2 (similar to HIP). An advanced ARP mechanism could use such a property to give a host the ability to prove that it is the legitimate user of a MAC address by employing public/private keys. Then again, public/private key computations are too expensive to utilize in Layer 2 authentications. Instead, public/private keys may be used to establish symmetric keys to authenticate Layer 2 frames. In addition, in

HIP, the host, which desires to initiate a connection, is required to solve a puzzle to prevent denial of service attacks. A related approach could be used to prevent ARP flooding attacks.

An alternative approach may be to improve MACsec. To provide some level of visibility for Layer 2 devices and protect the original MAC and IP address binding, we propose to add a hop/link count field and an original source MAC address field into the SecTAG in MACsec [20]. Since MACsec is end-to-end in Layer 2, in the case that the router is the end point for a transmission in Layer 2, all the information regarding the origin of security (original source MAC address) is removed and a new link layer (Layer 2) header is added into the frame. This limits the capability of tracking spoofed IP/MAC packets/frames because the IP and MAC address pair in the outgoing packet/frame at the router does not provide any information regarding the original source MAC address used. In addition, it removes the source IP address and the source MAC address binding created in the original frame. When MACsec is used with a SecTAG, it provides security for data frames and binds the MAC and IP addresses via the integrity check value (ICV). The ICV is calculated over the destination MAC address, source MAC address, SecTAG, and user data. Thus, it prevents unauthorized modifications of the MAC and IP addresses. The binding between the original source IP and MAC addresses is critical to provide security and aid in billing procedures. Note that a DHCP server may assign a different IP (Layer 3) address to the same subscriber each time it joins the network even though the MAC address of the subscriber stays the same.

We transmit the original source MAC address in the Ethernet frame, specifically, when a router transmits the frame in a LAN/MAN. This feature helps to prevent ARP attacks as well, especially if MACsec is not used to secure the ARP messages. However, it also introduces an extra complexity to routers and requires link layer data transfers between the ports of routers. Moreover, the maximum size of the data

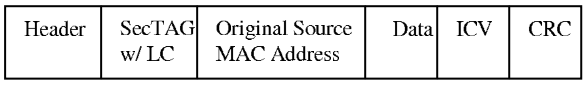

Figure 9: Proposed Ethernet frame format with MACsec.

field in a MAC frame should be reduced to convey the 48 bits original source MAC address field.

In addition, we add a hop/link count field into the SecTAG to track the number of hops/links that a frame travels. This is necessary because switches/bridges do not have MAC addresses. They are invisible in Layer 2/3. Even though the port number is contained in the SecTAG, it is not possible to identify the number of switches/bridges that a frame passes in a network. The port number in the SecTAG identifies the port number of the last end-point bridge/switch. A hop/link count field makes end devices aware of intermediate switching/bridging devices. Moreover, it helps to track the traffic in a network providing the information regarding the number of Layer 2 devices on the path. In conjunction with topology plans, network administrators can also examine whether frames take the expected path, and IDSs can utilize this field to recognize spoofed or misguided frames. The hop/link count field used here denotes the number of hops/links in Layer 2 instead of Layer 3. Note that Layer 3 already has the IP time-to-live field. The proposed new hop count field has a fixed size to prevent frame fragmentations later in the network. Each SecY should increment the hop/link count before transmitting a frame. Routers require additional features to transfer this link layer information between the ports, as well. However, it may not always be desirable to reveal the number of Layer 2 devices in a network. In such a case, the tag control information (TCI) field in the SecTAG may be used to indicate whether the hop/link count (LC) field is being utilized. A diagram of the new Ethernet frame used in this thesis is shown in Figure 9.

3.2 Cryptographic Identities at the Data Link Layer

Our approach to securing the data link layer is based on separation of identities and locations in networks. We believe that decoupling an end point identity from its network location is necessary to improve network and security services with mobility in the Internet. We choose to utilize cryptographic identities in the data link layer instead of using traditional MAC addresses. Moreover, we introduce a new security inter-layering concept to allow employing various upper layer identities in the data link layer. We argue that the data link layer should use a secure namespace of upper layers instead of MAC addresses, thus avoiding the overhead that a new secure namespace for the data link layer would create. This also prevents the risk of the introduction of possible weaknesses with a new namespace. Note that secure identities are incorporated into MACsec to provide security in the data link layer. The rest of this thesis focuses on using cryptographic identities for the data link layer with security inter-layering. Cryptographic identities and identifiers for the data link layer are further discussed in Chapter 4.

3.3 A New Concept: Security Inter-layering

We propose a new security inter-layering concept where layers make the information regarding the security protocols utilized in each layer available to other layers. Our main motivation behind the security inter-layering concept is to create top-to-bottom secure and flexible network architectures. This concept is envisioned as a security inter-layering control plane in a reference model as shown in Fig. 10.

The current implementations of security protocols in various layers provide a modular approach to security. Since each layer offers security services independent of other layers, security in one layer may provide a sufficient level of assurance against security weaknesses present in other layers. However, this approach also generates a computational overhead and increases the bandwidth usage in networks. The redundant use of security measures in various layers may be prevented if layers are informed regarding the security implementations in other layers. In fact, some information is available to lower layers in IP headers when IP Security (IPSec) is utilized [73], [86]. On the other hand, in lower layers, it is difficult to keep track of the security associations of the transport layer or upper layers as security associations may have states and/or detailed message/segment/data analysis may be required. In addition, security protocols in various layers rarely interact with each other to consider the security requirements and possible exploitations introduced by other protocols. For instance, in a recent IPSec vulnerability, an attacker modifies sections of an IPSec packet to cause a network host to generate an error message. When this error message is relayed via the Internet Control Message Protocol (ICMP), because of the design of ICMP, the message directly reveals segments of the header and payload of the inner datagram in cleartext. Consequently, an attacker intercepting the ICMP messages can retrieve the plaintext data [6]. Moreover, the layers of reference models change dynamically possibly rendering fixes introduced at present to be insufficient for future architectures and protocols. We believe that a more capable method is required to create a comprehensive and flexible security control mechanism. We propose a new security inter-layering concept to inform each layer regarding security protocols and features utilized in other layers.

The security inter-layering concept also allows the use of the same namespaces in various layers in networks. For instance, a lower layer may choose to utilize a different secure namespace each time depending on the applications, user parameters, or network settings. Security inter-layering may be utilized to create secure bindings among namespaces and to protect against misbindings as well. A lower layer may create identities using a secure namespace of upper layers or vice versa. Moreover, depending on the applications or the security requirements imposed by users/networks, security inter-layering may coordinate security implementations in

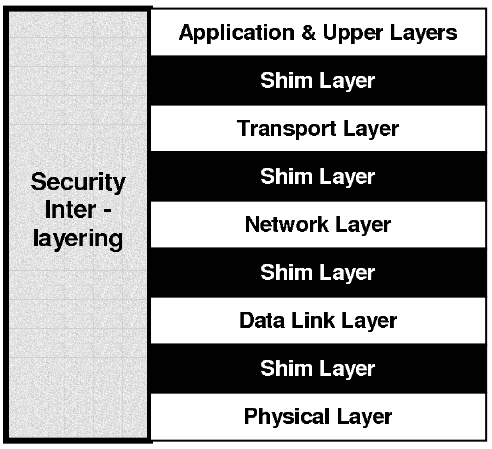

Figure 10: A reference model with security inter-layering.

each layer to provide efficient and comprehensive network security. For example, if data confidentiality is provided by the Transport Layer Security (TLS) protocol, lower layers may not be required to encrypt the messages. Furthermore, security focus in each layer may/should be different and dependent on the functionalities of layers. While confidentiality may be important in the upper layers, the focus may be anonymity issues or the authentication of a source in lower layers. Finally, this concept easily adapts to future architectures or namespaces since it is not a specific security architecture limited to a certain layer or a network architecture.

The security inter-layering process, as it is conceptually shown in Fig. 10, is expected to be protective and trustworthy. It relies on many aspects of the underlying environment. For example, defects in operating system security, design and implementation errors in security protocols, etc. will affect the overall security. The design of the security inter-layering control plane and the methods the upper layers use to communicate the identities with the lower data link layer are beyond the scope of this research. Instead we focus on generating identifiers and security parameters for the data link layer security architecture.

3.4 The Flow Diagram

In our new data link layer architecture, since hosts utilize cryptographic identities to create data link layer identifiers instead of fixed MAC addresses, hosts are expected to prove their identities and generate data link layer identifiers before receiving LAN access services. Figure 11 presents a simple flow diagram of this process at a host. Each host first generates a global data link layer identifier employing an upper layer identity. The identifier generation process and selection of identities are described in Chapter 4. Then, it negotiates security parameters and an identifier with an authentication server utilizing the key establishment protocol as explained in Chapter 5. Finally, the host computes session keys and enables message authentication. The key hierarchy, computations, and message authentication algorithms used in this step are explained in Chapter 6. At the other end of this process, an authentication server responds to the authentication requests by utilizing the key establishment protocol as discussed in Chapter 5. Figure 12 illustrates a simple flow diagram of this process at an authentication server. In addition, the authentication server is responsible for keeping a list of data link layer identifiers and distributing the keys to other network devices. For a detailed description of the key distribution scheme, refer to Chapter 6.

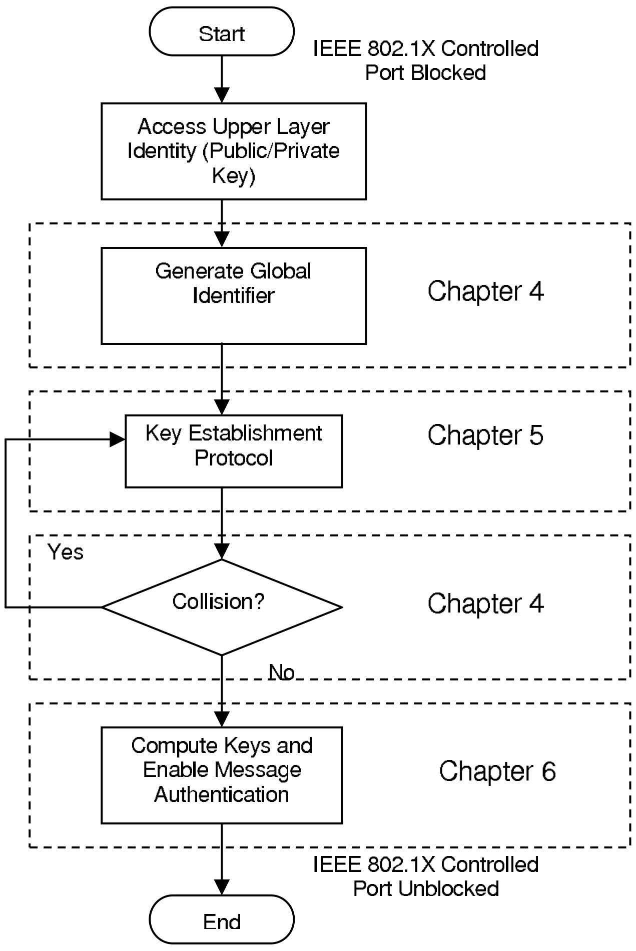
Start
IEEE 802.1X Controlled Port Blocked
Access Upper Layer Identity (Public/Private Key)
Generate Global Identifier
Chapter 4
Key Establishment Protocol
Chapter 5
Yes
Collision?
Chapter 4
No
Compute Keys and Enable Message Authentication
Chapter 6
IEEE 802.1X Controlled Port Unblocked
End

Figure 11: A simple flow diagram of the data link layer identifier and security parameters generation process at a host.

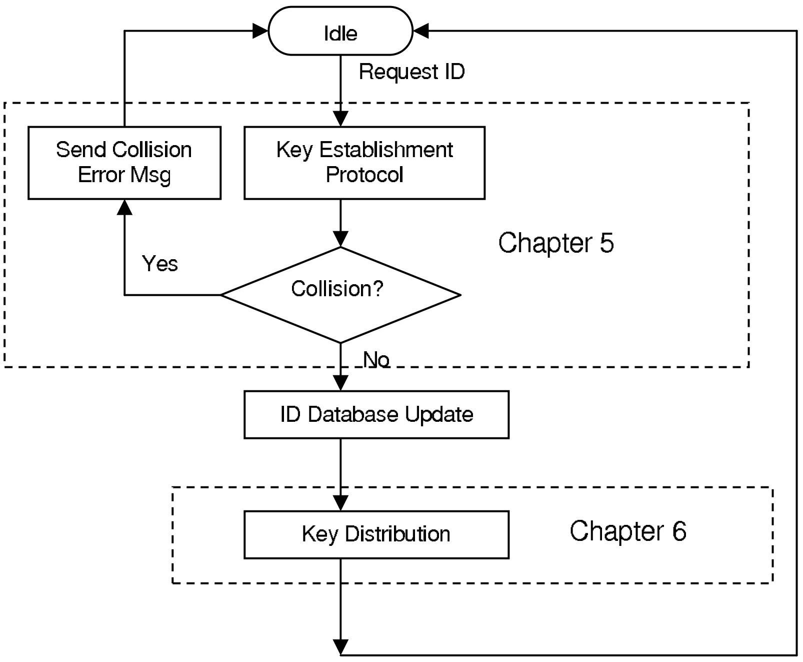

Figure 12: A simple flow diagram of the data link layer identifier and security parameters generation process at an authentication server.

CHAPTER 4

PUBLIC AND PRIVATE KEYS-BASED IDENTITIES AND IDENTIFIERS

4.1 Identities: From Upper Layers to Lower Layers

In the data link layer, we choose to utilize upper layer cryptographic identities instead of lower layer identities to generate data link layer identifiers, as illustrated in Figure 13. The main motivation behind transporting identities in this direction is to provide flexibility in selecting identities. While the lower (physical) layer lacks security, there are several security applications and services available at upper (network, transport, etc.) layers. In addition, applications at upper layers may easily generate/modify identities. On the other hand, any security parameter or identity embedded in the physical layer may be hard to change and may have a more limited scope. Moreover, a compromise of lower (physical) layer identities will prevent network access in our data link layer security architecture. Finally, if it becomes necessary to utilize physical layer identities in future networks, upper layer applications may discover physical layer identities and present them to the data link layer as upper layer identities.

4.2 Identifiers

In our data link layer security architecture, we utilize public keys as data link layer identities assuming that the data link layer has access to upper layers' public and private keys. However, since public keys may be too long to include in each frame/packet and public key lengths may differ for each host, we employ the hashes of public keys to identify hosts in the data link layer. We use the term "identifier" for the hash value since the hash of a public key is a representation of the real identity. In addition,

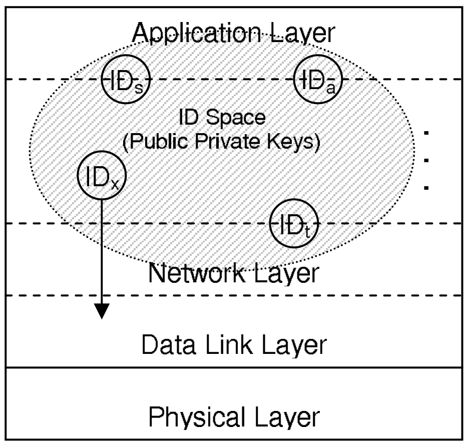

Figure 13: Data link layer identity selection.

we utilize fixed size hash values as identifiers to provide flexibility in data link layer identities. In this manner, changing the type of identity or the identity itself will not affect how identifiers are used in LANs. Moreover, it may be desirable to hide real identities from passive attackers by using dynamic identifiers. For instance, each time a host connects to a LAN, it may choose to generate and negotiate a different identifier without changing its public and private key pair. This is achieved by using pseudo-random values as additional inputs to the hash functions.

Utilizing fixed size hash values as identifiers introduces the possibility of identifier collisions. This is also known as Birthday Paradox. For instance, for a m bits long hash value, where $N \equiv 2^m$, the probability that the hash values of k random hosts are distinct is $e^{-k(k-1)/2N}$, as derived in (2).

$$(1-1/N)(1-2/N)\ldots(1-(k-1)/N) \equiv \prod_{i\equiv 1}^{k-1}(1-i/N) \cong \prod_{i\equiv 1}^{k-1}\left(e^{-i/m}\right) \equiv e^{-k(k-1)/2N} \tag{2}$$

Hence, the probability of at least one collision is $\left(1-e^{-k(k-1)/2N}\right)$, whose value is above 0.5 when $k \equiv 1.17\sqrt{N}$. We allow limited identifier collisions by assuming that the identifiers of hosts in local networks are not globally unique. In our new data link

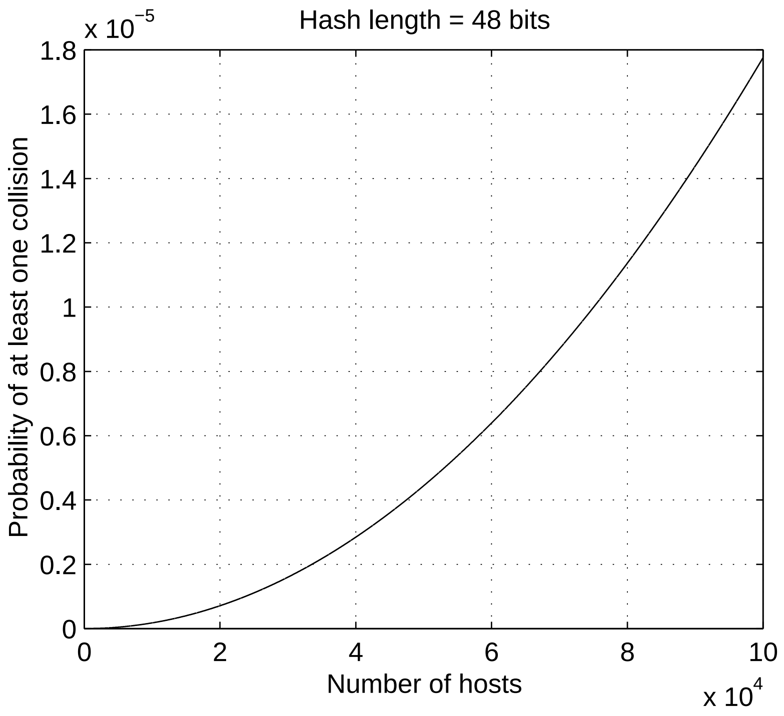

Figure 14: Probability of at least one hash collision for a hash of 48 bits versus number of hosts.

layer security architecture, authentication servers provide each host an IP address. Nevertheless, servers negotiate host identifiers to prevent collisions in each subnet.

The probability of at least one collision for a hash of 48 bits for various numbers of hosts is presented in Figure 14. As expected, the probability of a collision increases as the number hosts grows. Our hash size of 48 bits for data link layer identifiers has a very low probability of a collision even for a large number of hosts. For instance, the probabilities of a collision are 1.77458×10^{-9}, 1.77617×10^{-7}, and 1.77632×10^{-5} for 1000, 10000, and 100000 hosts, respectively.

Figure 15 presents the probability of at least one collision versus various hash sizes for 10000 hosts. According to the results, the probability of a collision decreases as the hash size increases for a fixed number of hosts. Thus, both an increase in the number of hosts and a decrease in the hash size increase the probability of a collision. We can not realistically estimate the number of hosts may be allowed in a collision-free local

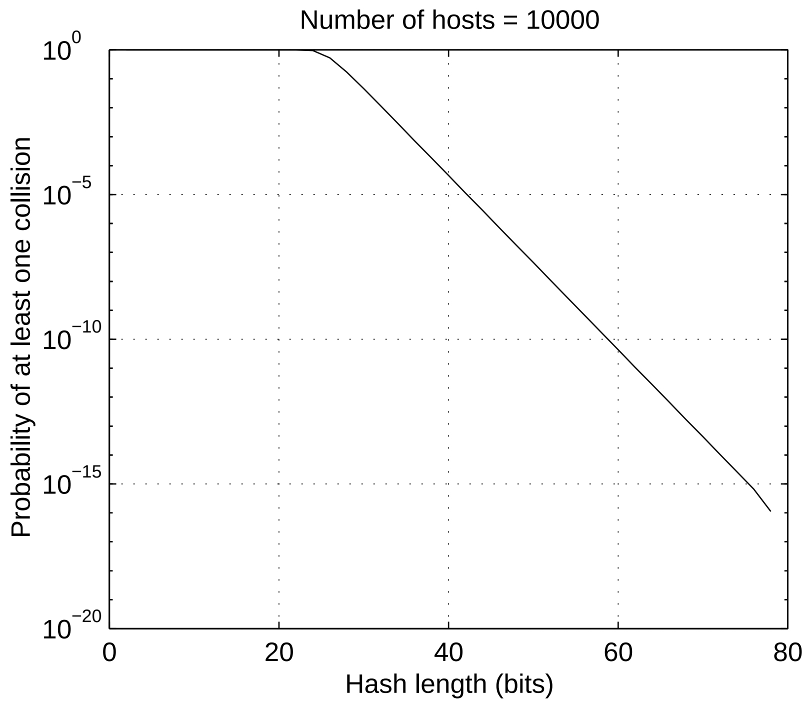

Figure 15: Probability of at least one hash collision for 10000 hosts versus hash length in bits.

identification realm in the future. However, since our data link layer architecture does not require local data link layer identifiers to be globally unique, utilizing a hash value of 48 bits is adequate for a local data link layer identifier.

In addition to local data link layer identifiers, we utilize hash values of 120 bits for global end host identifiers. Consider 10^{15} hosts, the probability of a collision is 3.76158×10^{-7} when hash values of 120 bits are used. In addition, the theoretical address space for hash values of 120 bits is 2^{120}, which is about 1.3×10^{36} addresses. Finally, we assume that global end host identifiers are globally unique with no collisions.

4.3 Global and Local Data Link Layer Identifiers

We utilize a local data link layer identifier (L2ID) for each host as a MAC address. Each L2ID is valid for one IP address only. Mobile hosts renegotiate local identifiers each time they move to other network segments. In addition, hosts may be required to negotiate different local identifiers when they reconnect to the same LAN. Moreover, hosts may periodically change their local identifiers. This architecture, using dynamic local identifiers (L2IDs), provides anonymity at the data link layer.

Global end host identifiers, G2IDs, are used to generate dynamic local data link layer identifiers, L2IDs. Since G2IDs identify end points, they are globally unique. First, a host computes a long hash value of its public key and generates its G2ID. Then, the same host computes a shorter hash value, its L2ID, using the G2ID and pseudo-random numbers (PRNs). Short hash values are utilized for L2IDs since they are incorporated in each frame. Longer hash values are used for G2IDs as they are long-lived and globally unique. G2IDs are registered and stored in databases, allowing backtracking afterwards. Specifically, G2IDs are utilized as the short representations of public keys to reduce memory and bandwidth requirements in our architecture.

4.3.1 Generating Global Data Link Layer Identifiers (G2IDs)

We compute global identifiers for end hosts in our data link layer security architecture with an algorithm similar to the keyed hash identifiers (KHI) method described in [79]. Inputs to our algorithm and the KHI method are the public key information of an end host (*bitstring*) and a *context identifier* (ID) as in the KHI algorithm. *bitstring* is a presentation of the public key information (identity) of an end host. *context ID* is a randomly generated value defining the expected usage context of the particular global identifier. *context ID*s allow generating different global identifiers while utilizing the same public key (*bitstring*) for different usage contexts, mechanisms, or protocols.

A KHI is generated as follows:

$$KHI \ = \ prefix \mid extract_l \left(SHA1 \left(expand \left((context\ ID) \ \mid \ (bitstring)\right)\right)\right), \qquad (3)$$

where *expand* is an expansion function, which is designed to overcome recent attacks on SHA1 [1], [94], $extract_l$ is an extraction function, and | denotes concatenation. Output is obtained by extracting a $\prec l \succ$-bits-long bitstring from the argument bitstring and concatenating a $prefix$ value.

Our data link layer global identifier is generated using the algorithm below:

$$G2ID \;=\; context\ ID \mid extract_{120}\left(SHA1\left(expand\left((context\ ID) \;\mid\; (bitstring)\right)\right)\right), \tag{4}$$

where the *context ID* is utilized interchangeably with the $prefix$.

As in KHIs, G2IDs in our architecture are designed to serve as identifiers rather than locators. While, in KHIs, the $prefix$ is used to distinguish KHIs from IPv6 addresses, we utilize the *context ID* instead of the $prefix$ to distinguish global identifier contexts (naming methods). For instance, in our algorithm, the *context ID* may be used to distinguish global identifiers for the data link layer from global identifiers for the application or transport layers. In addition, in our algorithm the *context ID* is limited to 8 bits in length to generate 128 bits long G2IDs. Note that the hash function may be selected differently. In that case, there is no need for the *expand* function.

We assume that data link layer identities (public keys from upper layers) belong to the Rivest Shamir Adelman (RSA) public key algorithm. In RSA, the public key consists of the modulus, n, and the public exponent, e, as defined in [83]. For instance, the ciphertext (C) of a message (M) can be generated with the RSA public key as follows:

$$C \;=\; M^e \bmod n. \tag{5}$$

The ciphertext in (5) can be decrypted using the RSA private key, which consists of the modulus, n, and the private exponent, d, as follows:

$$M \; = \; C^d \, mod \, n. \tag{6}$$

To generate the G2ID, we encode the *bitstring* for the RSA public key utilizing four information fields as defined in [49]: exponent length, exponent (e), modulus length, and modulus (n). The public key exponent length is one or three octets depending on its value. If the exponent length is in the range of 1 to 255, it is represented as one octet. Otherwise, the exponent length is represented as one zero octet followed by a two octet unsigned length. Moreover, both the exponent and modulus are each limited to 4096 bits in length. The *bitstring* value for a RSA public key is calculated as follows:

$$bitstring_{RSA} \; = \; exponent \mid exponent\ length \mid modulus \mid modulus\ length\ . \tag{7}$$

Note that all the fields in (7) are encoded in network byte order.

4.3.2 Generating Local Data Link Layer Identifiers (L2IDs)

To generate local identifiers, we utilize G2IDs and two PRNs (PRN1, PRN2), each 64 bits long. Initial PRNs are selected randomly. However, for the subsequent L2ID computations, PRNs are exchanged during the key establishment protocol. A simple method of generating a L2ID is to concatenate the G2ID with PRNs and hash the result using SHA1 as in (8).

$$L2ID \; = extract_{48}\left(SHA1\left(expand\left(G2ID \mid PRN1 \mid PRN2\right)\right)\right). \tag{8}$$

Another method of generating L2IDs is to utilize the Advanced Encryption Standard (AES). This method may be preferred if an efficient implementation of AES is available in hardware/software. In this method, a single-length rate-one modification detection code (MDC) based on block ciphers is used to generate hash values [76].

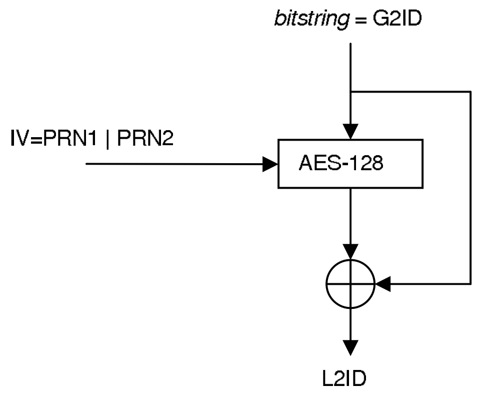

Figure 16: The L2ID generation method based on an AES-128 block cipher.

We use Matyas-Meyer-Oseas hash algorithm with AES. AES specifies the Rijndael algorithm, which is a symmetric block cipher that can process data blocks of 128 bits, using cipher keys with lengths of 128, 192, and 256 bits [2].We use a G2ID value (128 bits) as the input bitstring and PRNs as the initial vector (IV) of the MDC. To create a 128 bits long IV, we concatenate PRN1 and PRN2. The L2ID generation process in this method is illustrated in Figure 16.

Each host computes its L2ID and negotiates it with a server in the network. Servers provide pseudo-random numbers (PRN2s) to hosts and check for L2ID collisions. However, to limit the computational overhead and reduce DoS attacks, servers compute hash values towards the end of the key establishment process. Assuming that servers support inter-connected LANs, they inform hosts regarding the IP address(es) as well. This approach gives the control of the locations of hosts to servers. Finally, servers employ a probing mechanism to ensure the aliveness of each host in the network. Note that, in contrast to ARP, our naming architecture verifies the identities of hosts before binding the identities and locations.

4.4 The New Ethernet Frame Format

Fig. 17 presents an Ethernet frame format utilizing L2IDs. Existing source and destination MAC address fields in the frame header, each 48 bits, are used to carry source and destination L2IDs. In essence, we are changing fixed MAC addresses to random identifiers enabling backward compatibility. However, since network devices change source MAC addresses (L2IDs) at each link, we incorporate an original source identifier field into the frames as well. An Ethernet type field is included in the frame format to distinguish secure frames and allow coexistence of other systems in the same environment. In addition, a link count field is used to improve data link layer visibility in networks. Switches and bridges are invisible in the data link layer, because they do not have MAC addresses. A link count field is used to make end devices aware of intermediate switching/bridging devices. A security parameters field is also incorporated into the frames to convey additional information such as counter values for replay protection, cipher suites used, data length, key indexes, whether confidentiality or integrity alone are used, etc. Finally, an integrity check value (ICV) field is added into each frame to authenticate messages. Message authentication codes are used to compute the ICV to verify the source of a message and its integrity. The ICV field authenticates the destination and original source L2IDs, Ethernet type, link count, security parameters, and data fields. With authenticated source and destination L2IDs, network administrators can identify and locate the origin of data traffic in LANs. Note that the maximum size of the data field in a frame should be reduced to allow carrying the original source L2ID, Ethernet type, link count, security parameters, and ICV fields. MACsec has also adopted a similar approach by changing the Ethernet frame format and reducing the maximum service data unit size available to users of the insecure MAC Service [11].

An intentional side effect of our new security architecture for the data link layer is the separation of network locations (IP addresses) and end points (identities). The argument of whether network locations and end points should be separated in the Internet architecture is controversial. We believe that it is a necessary approach to improve network and security services with mobility in the Internet and data link

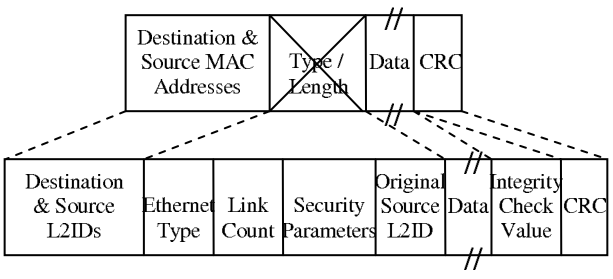

Figure 17: A modified Ethernet frame format utilizing data link layer identifiers.

layer architectures. In addition, inter-layer communication between the data link and network layers is required to link identities and locations in this architecture. For instance, source and destination identities or identifiers should be included in the options field of IPv4 headers to locate mobile hosts.

CHAPTER 5

THE KEY ESTABLISHMENT PROTOCOL

5.1 Preliminaries: On the Security Models and Requirements

Key establishment protocols are utilized by two or more parties to negotiate and agree on security parameters. Since we aim to link message authentication and identification processes in our data link layer security architecture, negotiations for the security parameters (key establishment) take place at the same time as the data link layer identifier generation process.

In security models, if a trusted third party (TTP) is not involved in the authentication process, it is difficult to mutually authenticate parties and to prevent man-in-the-middle (MITM) attacks. In in our data link layer security architecture, the authentication process takes place between end hosts and servers during the key establishment protocol. If the identities of hosts are not registered or certified, servers cannot verify these identities. Similarly, if the identities of servers are not publicly known and registered, hosts cannot distinguish impersonators from the authorized servers. Unregistered hosts or servers make key establishment protocols vulnerable to identity misbinding attacks as well. In identity misbinding attacks, an attacker will be able to change the binding between a session key and the parties in the session to create an authenticated binding between himself and one of the parties [35]. There are two methods available to provide mutual authentications: utilizing a TTP to authenticate parties and utilizing pre-distributed pairwise secret keys. We leave the mutual authentication process as an implementation choice. However, we address identity misbinding attacks in the design of the key establishment protocol, assuming that a method of identity verification is available.

In our data link layer security architecture, since hosts are not authorized or authenticated before they can communicate to a server, we assume that the Extensible Authentication Protocol over LANs (EAPOL), defined in the IEEE802.1X (Standard for Port-Based Network Access Control) [3], is utilized to carry Extensible Authentication Protocol (EAP) frames (control messages) between the hosts (supplicants) and data link layer switches (authenticators). Switches allow EAP frames, which convey the key establishment protocol messages, between hosts and authentication servers during the key establishment process. However, until the key agreement and authentication process is completed between hosts and switches, regular data traffic is blocked (refer to Chapter 6). This is very similar to the Robust Security Network (RSN) model of the IEEE 802.11i (MAC Security Enhancements) wireless standard [5], [9]. As illustrated in Figure 18, servers, switches, and hosts in our model will correspond to the authentication servers, authenticators, and supplicants in an IEEE 802.1X setup, respectively. The IEEE 802.1X standard may utilize EAP to support a variety of authentication schemes. However, since in our key establishment protocol, we incorporate the data link layer identifier generation process, a puzzle mechanism to alleviate DoS attacks, and an identity protection feature, we are not able to directly use any of the EAP methods as they exist at present. Existing EAP methods may be modified or a new EAP method may be proposed to carry our key establishment protocol messages in the future.

5.2 The Key Establishment Protocol

A general form of the key establishment protocol in our data link layer security architecture is illustrated in Fig. 19. Our key establishment protocol is based on the Just Fast Keying (JFK) protocol [17] and the Sign-and-MAC (SIGMA) protocol [74] with appropriate modifications to provide identity protection for the initiator. The

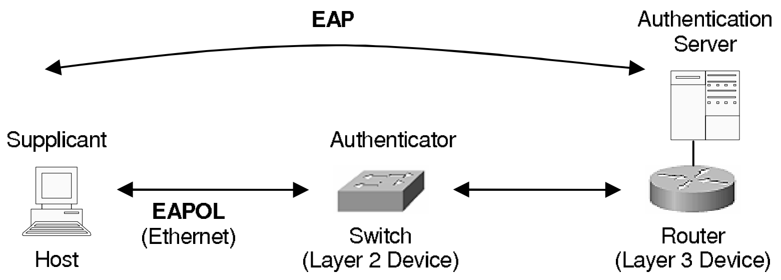

Figure 18: The IEEE802.1X EAP authentication model of the data link layer security architecture.

key establishment protocol utilizes the Diffie-Hellman exchange. In the exponential notations g^x and g^y, x and y are random exponents, and g is a Diffie-Hellman group generator. We assume that the host (initiator) knows an acceptable group generator of the server (responder). In the first message, the alias $\hat{A}$ is computed by the initiator, A, as $\hat{A} = hash(A, r)$, where r is a random number. In the third message, the initiator reveals its identity to the responder by encrypting both its real identity, A, and the random number, r. The notation $K_e\{\}$ is used to denote that the data between the brackets are encrypted with the symmetric key of K. The symmetric key K is derived from the Diffie-Hellman value (master key) g^{xy}. However, in the key generation process, session keys are derived from the master key independently of K. Finally, the notations $sig_A()$ and $sig_B()$ are used to denote that the messages between the parentheses are signed with the private keys of A and B, respectively.

In our key establishment protocol, public-private keys, which are long-lived identities for hosts, are utilized to help create short-lived encryption and authentication keys for the data link layer. Specifically, a host and a server generates a master key, g^{xy}, using a Diffie-Hellman key agreement protocol. After completing the key establishment protocol, both the host and the server calculate encryption and message authentication keys from the master key. We employ public keys in the key establishment protocol for three purposes: verifying identities, creating data link layer identifiers, and generating session keys (refer to Chapter 6).

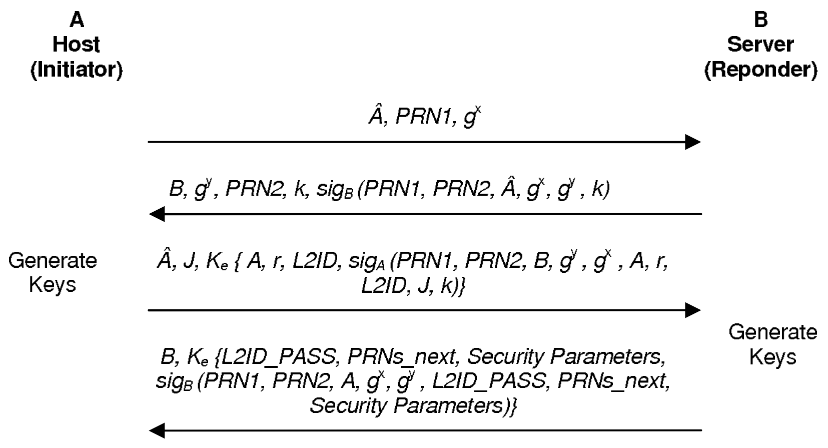
A
Host
(Initiator)
B
Server
(Reponder)
Â, PRN1, g^x
B, g^y, PRN2, k, sig_B (PRN1, PRN2, Â, g^x, g^y, k)
Generate Keys
Â, J, K_e { A, r, L2ID, sig_A (PRN1, PRN2, B, g^y, g^x, A, r, L2ID, J, k)}
Generate Keys
B, K_e {L2ID_PASS, PRNs_next, Security Parameters, sig_B (PRN1, PRN2, A, g^x, g^y, L2ID_PASS, PRNs_next, Security Parameters)}

Figure 19: The key establishment protocol for the data link layer security architecture.

5.2.1 Observations

The following observations can be made for our key establishment protocol. First, in the key establishment protocol, the inclusion of the identities under the signatures prevents identity misbinding attacks. In identity misbinding attacks, for instance, an active (man-in-the-middle) attacker may replace or modify messages between two parties to create an authenticated binding between the attacker and one of the parties (without knowing or influencing the secret key). The basic station-to-station (STS) protocol is not used here because of its vulnerability to identity misbinding attacks [74]. Second, in the key establishment protocol, the signed message part by the initiator ($sig_A\{PRN1, PRN2, B, g^y, g^x, ...\}$) is incorporated into the encrypted message to prevent any third party from checking the signatures to prove that the communication took place. Third, the key establishment protocol incorporates a puzzle to mitigate DoS attacks. The responder tries to ensure that the initiator is sincere by allocating resources to solve the puzzle. Our puzzle mechanism utilizes $PRN1$, $PRN2$, B, $\hat{A}$, J, and k. For the details of the puzzle mechanism, see Section 5.2.2. Fourth, the key establishment protocol protects the identity of the initiator from active attacks. The initiator does not reveal its identity until it can generate keys for encryption and message authentication. Fifth, the key establishment protocol does not require the responder (server) to calculate the session and message authentication keys until it receives the answer of the puzzle. This prevents unnecessary state generations and key computations at the server.

In the key establishment protocol, the server verifies the solution of the puzzle, J, and checks for L2ID collisions before sending the last message. If the server does not receive a correct solution from the host, $\hat{A}$, it sends a $PUZZLE\_FAIL$ message in cleartext and discards all the information. If the server receives a correct solution of the puzzle, but there is a L2ID collision, it replaces the $L2ID\_PASS$ with the $L2ID\_FAIL$ in the last message. In addition, the server may still provide the $PRNs\_next$ in the last message to reduce the number of handshakes required for the key establishment protocol next time with the same host. However, this approach requires the server to register the K, PRNs, $\hat{A}$, A, and the security parameters of the host. While this approach may reduce the load at the server, it demands more memory space.

Since this implementation does not utilize a TTP, the identity of servers (public keys) should be registered and advertised to hosts. Otherwise, MITM and server impersonation attacks may not be prevented. Alternatively, servers and hosts may mutually authenticate each other utilizing pre-distributed pairwise secret keys. If a TTP is utilized in the key establishment protocol, both hosts and servers should communicate with the TTP to establish mutual session keys or to verify the authenticity of public keys. However, the involvement of a TTP may cause delays and increase the complexity of the key establishment protocol.

As illustrated in Fig. 20, the role of a TTP depends on the security model. A TTP may deliver session keys and other security parameters to both servers and hosts. Another model may utilize a TTP to verify the identities (public keys) of hosts and servers. In both TTP models, since, in our data link layer security architecture,

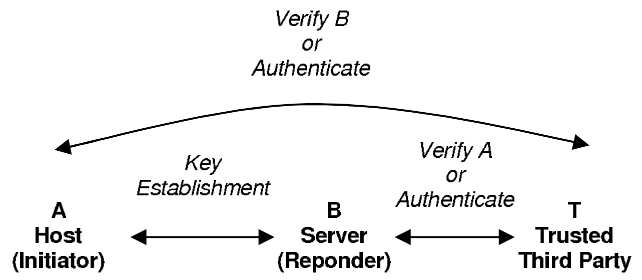

Figure 20: A trusted third party-based authentication and key establishment.

hosts are not authorized or authenticated before they can communicate to a TTP, an additional protocol, such as the Extensible Authentication Protocol over LANs (EAPOL), should be utilized to allow control messages to pass.

5.2.2 The Puzzle Mechanism

DoS attacks based on protocols remain a serious threat to networks and users. The 2004 Computer Security Institute and US Federal Bureau of Investigation (CSI/FBI) Computer Crime and Security Survey shows that the most expensive computer crime in 2004 was DoS attacks [59]. DoS attacks, by their nature, are difficult to prevent. A DoS attack may be characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service [68]. Key exchange and authentication protocols are vulnerable to DoS attacks that exhaust the servers' processing resources. Puzzles have been proposed as a countermeasure to DoS threats in communication networks [28, 48, 52, 64, 77, 90, 91]. In our data link layer security architecture, we utilize a client puzzle in the key establishment protocol to delay state creations at servers. Hosts (initiators) perform computations to solve a puzzle and prove that they are willing to allocate resources to access to the servers. The puzzle mechanism allows servers to check the answers by simply computing one hash function.

In the key establishment protocol, the server sends a puzzle, containing the random number $PRN2$ and the puzzle strength k, after receiving the initial message from the host without computing the Diffie-Hellman (master) key. The host must solve this cryptographic challenge to continue the key exchange. The server discards messages containing incorrect puzzle answers. The server adjusts the level of difficulty of the puzzle by setting value k.

To solve the puzzle, the host (initiator) generates a number of random numbers, Js, and computes the hash values as in (9) until the lowest order k bits of the hash are all zeros. The host gives up solving the puzzle if it exceeds the puzzle lifetime. The server verifies the puzzle by computing the same hash value once using the J provided by the host.

$$SHA1\left(PRN1|\ PRN2|\ B\ |\hat{A}\ \ |J\right). \tag{9}$$

In (9), the random numbers J, $PRN1$, and $PRN2$ are 64-bit integers whereas B and $\hat{A}$ are 128-bit integers. The puzzle difficulty, k, is an 8-bit integer. It takes, on average, $2^{(k-1)}$ hash calculations to solve the puzzle [48]. Since the output of the hash function is 160 bits long, the reasonable values of k lie between 0 and 80. Setting the k to 0 means that the puzzle mechanism is disabled. In that case, the server accepts any J value.

5.3 Discussion

Since our objective is to secure the data link layer and bind upper layers and the data link layer, we also focus on preventing design specific attacks, such as identity misbinding attacks, in addition to well-known attacks. While the authentication, confidentiality, and data integrity requirements are widely known and expected, the requirement of identity binding is usually overlooked. We should emphasize that identity binding is essential in our data link layer security architecture to authenticate messages. We prevent identity misbinding attacks in our key establishment protocol by including identities under signatures.

In key establishment protocols, identities are transmitted as a part of the protocols since each party needs to know the identity of the other party for mutual authentication. However, unprotected identities are prone to identity-probing attacks from any machine in the network. For instance, an attacker may initiate a key establishment protocol to find the identity of a machine at a certain IP address. To prevent this type of attack, the key establishment protocol may reveal the identity of the responder only after the initiator reveals its identity. On the other hand, in some cases, it may be more suitable to reveal the identity of the responder first. Note that it is not possible to design a key establishment protocol that protects the identities of both the responder and the initiator from active attacks since the first party, which proves its identity to the other party, is prone to active attacks. In our key establishment protocol, we choose to protect the identities of hosts from active and passive attacks.

A key property of secure protocols is the protection of past session keys in spite of the compromise of long-term secrets. This property is known as perfect forward secrecy. In our key establishment protocol, the Diffie-Hellman exchange provides this property for master keys. In addition, in the case that information leakage happens, where some session specific information or the value of a session key is learned by an attacker, we require that any adverse security consequence from such a compromise will affect the exposed session only. This security principle can be achieved by deriving session (temporary) keys, such as encryption and message authentication code keys, from a master key computed in the key establishment protocol, independently of the symmetric key of K (refer to Chapter 6). We utilize the Advanced Encryption Standard (AES) and the Counter with Cipher Block Chaining-Message Authentication Code (CBC-MAC) mode (CCM) for encryption and integrity check algorithms, respectively [2], [92].

Another desirable, but not necessarily required, property of secure protocols is non-repudiation. By non-repudiation property, the signer of a digital signature is prevented from denying having signed a document after signing it [76]. In general, this property prevents the denial of previous commitments and actions. However, this property comes with the price of digital signatures utilizing public-private keys. For that reason, we choose to employ this property only when it is essential in the key establishment protocol, which in our case is the last three messages.

While it may not be possible to prevent DoS attacks, key establishment protocols may utilize various techniques to reduce this type of attack. We utilize a puzzle mechanism, which is similar to the Host Identity Protocol (HIP) puzzle mechanism [77], to mitigate DoS attacks in our key establishment protocol. In our puzzle mechanism, we utilize $PRN1$s and $PRN2$s as session identifiers. We also utilize these pseudo-random numbers to generate L2IDs. An attacker can pre-compute puzzle solutions by estimating the $PRN2$s. To prevent pre-computation attacks, $PRN2$s should not be easily guessed by hosts. In addition, servers should generate new $PRN2$s once in every few minutes. Moreover, servers should verify the puzzle values in the responses. Furthermore, servers may need to remember old puzzles for a limited time to allow slower hosts to solve the puzzles. Also, utilizing $\hat{A}$ instead of real host identities prevents attackers from identifying hosts by observing messages. For that reason, our puzzle mechanism prevents attackers from pre-computing puzzle solutions for specific hosts. If the server receives a correct puzzle solution sent by an attacker, it will not be able to verify the signature in the received message. In that case, the server will send a $PUZZLE\_FAIL$ message to the host to prevent more attacks. The server should record the PRNs and the $\hat{A}$ and avoid utilizing these values in the puzzle mechanism. Attackers can send $PUZZLE\_FAIL$ messages to hosts to cause DoS. To prevent this type of attack, hosts should utilize a timer to end the sessions. Finally, since attackers can send false puzzle solutions to servers to cause DoS, servers should use a timer to wait for the correct puzzle solutions as well.

CHAPTER 6

NETWORK STRUCTURE, KEY MANAGEMENT, AND KEY HIERARCHY

6.1 Network Structure

In our data link layer security architecture, we utilize the IEEE 802.1X [3] concepts for access control. In addition, we incorporate the IEEE 802.1AE standard [11] and use a key hierarchy similar to the IEEE 802.11i standard [5] for future compatibility of wired and wireless networks. Our data link layer security architecture has three main components: authentication servers, authenticators, and hosts.

6.1.1 Authentication Servers

We utilize authentication serves to establish realms and security parameters in local networks. We assume that authentication serves are integrated into routers. Each authentication server records and manages the data link layer identifiers in its realm. Hosts negotiate security parameters and their data link layer identifiers (L2IDs) with authentication servers during the key exchange protocol. Specifically, authentication servers and hosts utilize the key exchange protocol to perform mutual authentication and to generate session keys. In addition, authentication servers assign IP addresses to hosts in their realm at the end of the key exchange protocol. Each new host moving into the realm of an authentication server is required to perform the key exchange protocol and obtain an IP address. However, authentication servers may assign the same IP address to several hosts with different L2IDs. We assume that authentication servers utilize a distributed database maintaining the list of G2IDs, L2IDs, and IP addresses for network access.

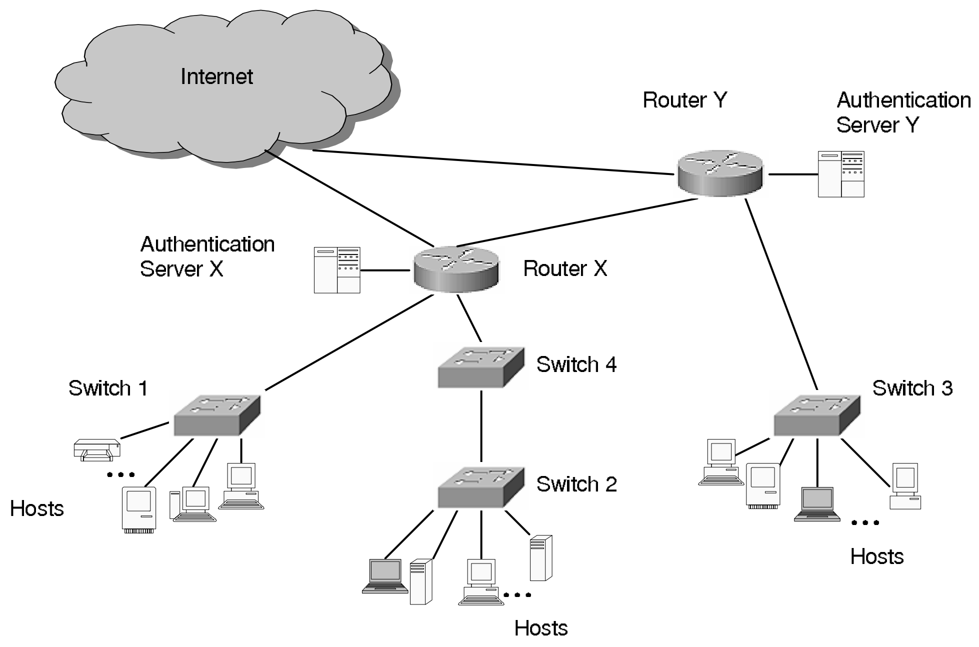

Figure 21: An illustration of the network architecture.

6.1.2 Authenticators

Authenticators are the data link layer devices that act as gateways between hosts and authentication servers. In our data link layer security architecture, authenticators function as access points similar to the security model of the IEEE 802.11i. We assume that authenticators are Layer 2 devices, such as switches. Figure 21 illustrates an example of the network architecture where switches function as authenticators. Authenticators communicate with authentication servers to receive their L2IDs, which are required to be unique among the realms, and to establish security parameters. In addition, each authenticator controls a connectivity association (CA). Each CA consists of an authenticator and a number of hosts. Each host, identified by a L2ID, participates in a single CA at any one time. However, a host with several data link layer connections (L2IDs) can participate in more than one CAs. For instance, in Figure 22, S_1 and S_2 are the authenticators in the CA_{S1} and CA_{S2}, respectively. Note that the host with two different identifiers, F and Q, is the member of both

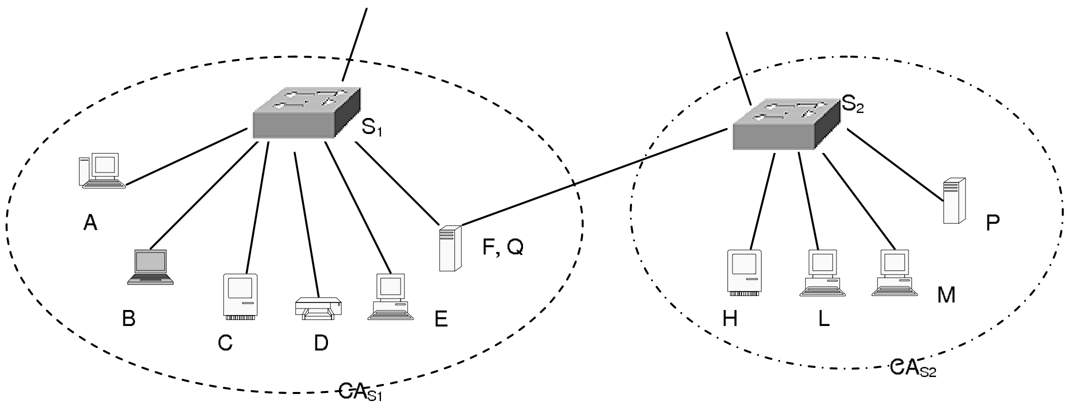

Figure 22: An illustration of connectivity associations.

CAs. Moreover, authenticators learn the security parameters of the hosts and the L2IDs of the hosts and other authenticators in the same realm from authentication servers utilizing a secure protocol.

Authenticators can limit the number of L2IDs that can participate in the CAs. Each CA is supported by security associations (SAs). For example, in Figure 22, A creates security associations with B, C, D, E, F, and S_1. Similarly, B creates security associations with A, C, D, E, F, and S_1. Figure 23 shows the security associations of the host M in the CA_{S2}. In the figure, the four SAs, SA_{MH}, SA_{ML}, SA_{MP}, and SA_{MS_2}, provide secure communication between M and the other hosts where each association is bidirectional. All the SAs in a CA use the same cipher suite at any one time. Finally, authenticators with direct links create SAs with each other, as well.

6.1.3 Hosts

In our data link layer security architecture, hosts are identified by L2IDs. Each L2ID, including the L2ID of an authenticator, corresponds to a MAC Security Entity (SecY) in the IEEE 802.1AE standard. Each host becomes a member of a CA and creates SAs with authenticators and other data link layer devices that they are connected to. Each SA represents a single value of the transient session key(s) used for a period by the cipher suite to support the communications between two data link layer

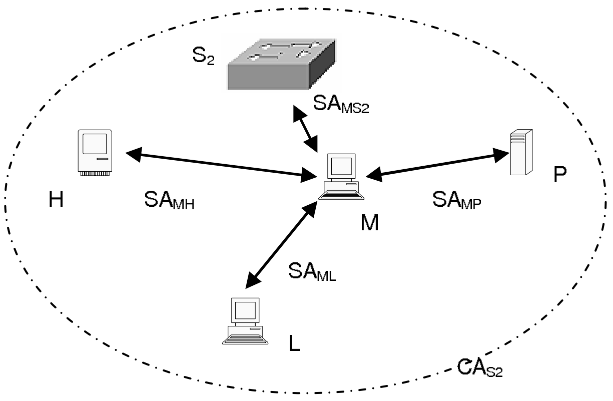

Figure 23: An illustration of the security associations of the host *M*.

devices. In addition, each host learns its IP address from an authentication server at the end of the key establishment protocol. The IP address is used to identify the home network/realm of a host. If a host moves from its home realm to another realm, it performs the key establishment protocol and learns its IP address from the authentication server of the new realm. However, before a host can send any data frames, it is required to create SAs in its CA. After creating a SA with the authenticator, the IEEE 802.1X Controlled Port is unblocked allowing the host to transmit and receive data frames. The host that completed the key establishment protocol utilizes the four-way handshake protocol, defined in the IEEE 802.11i standard [5], to establish a SA with the authenticator. After creating SAs, a host wishing to communicate with a destination host finds the location (IP address) and the identity/identifier of the destination host via a Fully Qualified Domain Name (FQDN) or another method. Note that hosts participating in different CAs communicate through authenticators.

6.2 Key Management

In our data link layer security architecture, there are four different type of links that require confidentiality, data authentication, and replay protection mechanisms: the links between authentication servers and authenticators, authenticators and hosts, hosts and hosts, and authenticators and authenticators.

6.2.1 Authentication Servers to/from Authenticators

In our data link layer security architecture, we assume that authentication servers and authenticators create security associations and utilize secure communication protocols. While the key establishment protocol messages do not require any encryption between an authenticator and an authentication server, the frames that carry the pair-wise master key (PMK) information and other security parameters between an authentication server and authenticator should be protected. An authentication server transports the information regarding a PMK, which is utilized by an authenticator and a host to derive a pair-wise transient key (PTK), to an authenticator. A PMK is derived from a master key computed at the end of the key establishment protocol between a host and an authentication server.

6.2.2 Authenticators to/from Hosts

Authenticators and hosts utilize the four-way and group handshake protocols, defined in the IEEE 802.11i standard [5], to create SAs and a CA with fresh keys. Authenticators and hosts employ EAPOL-Key frames in these protocols. The four-way handshake protocol enables an authenticator and a host to derive a fresh pair-wise transient key (PTK) from a pair-wise master key (PMK). Moreover, the authenticator confirms the liveliness of the host and that the host holds the PMK. In addition, during the four-way handshake protocol, the authenticator transports the group transient key (GTK) to the host. Furthermore, the authenticator informs the host regarding the cipher suite selection used in the CA and other hosts belonging to the same CA. At the end of the four-way handshake protocol, both parties install pair-wise encryption and integrity keys. The host installs the GTK as well. In our security architecture, a GTK and PTK represent a CA and SA, respectively. While the GTK is the same for all the data link layer devices in the same CA, the PMK is different for each SA.

6.2.2.1 Four-way and Group Handshake Protocols

Figure 24 illustrates the four-way handshake protocol utilized between an authenticator and a host and shows some of the important fields in the handshake messages. The four-way handshake protocol starts with an EAPOL-Key message from an authenticator to a host. The authenticator generates a nonce (a random or pseudo-random number) value (256 bits), called $ANonce$, and includes this value in the first message. This first message is not encrypted or protected. The host generates a nonce value (256 bits), called $SNonce$, and computes the PTK when it receives the first message. In our data link layer security architecture, the host and the authenticator utilize five inputs to compute the PTK: the PMK, $SNonce$, $ANonce$, and L2IDs of the authenticator and the host. The IEEE 802.11i standard describes the algorithm to compute the PTK. In the four-way handshake protocol, the PTK is partitioned into an EAPOL-Key Confirmation Key (KCK), an EAPOL-key Encryption Key (KEK), and a temporal key (TK) to protect unicast communication between the authenticator and the host [5]. The authenticator computes the PTK when it receives the second message from the host. The second message contains the $SNonce$ value (unencrypted) and a message integrity code (MIC) to detect any modifications in the message. The authenticator, first, extracts the $SNonce$ from the message, and then, computes the PTK and verifies the MIC over the whole message. The MIC value is calculated using the KCK. The authenticator sends an EAPOL-Key message, the third message, to the host including the $ANonce$, a starting sequence number, and a MIC check. This message informs the host that the authenticator is ready to use the TK for encryption. The last message in the four-way handshake protocol is sent by the host to the authenticator to acknowledge the completion of the four-way handshake. The host installs its keys after sending the last message. When the authenticator receives the last message, it installs its keys as well. This completes the four-way handshake.

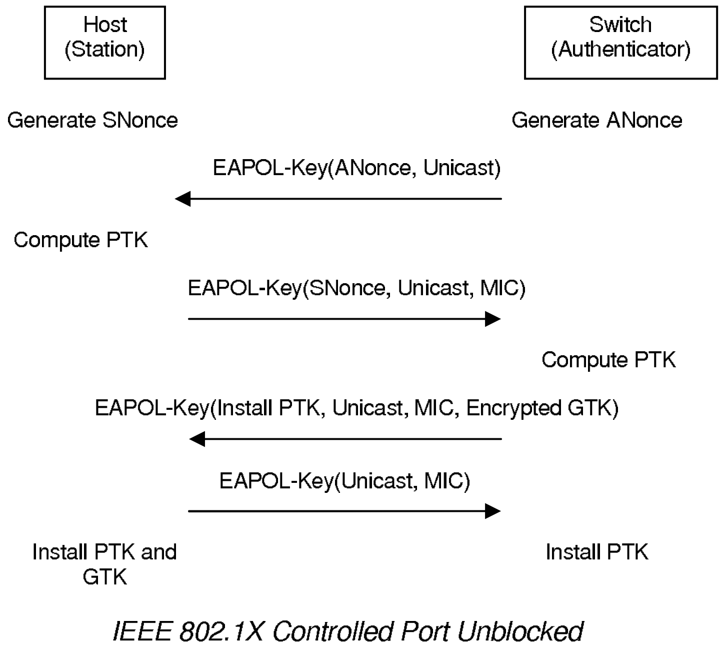
Host (Station)
Switch (Authenticator)
Generate SNonce
Generate ANonce
EAPOL-Key(ANonce, Unicast)
Compute PTK
EAPOL-Key(SNonce, Unicast, MIC)
Compute PTK
EAPOL-Key(Install PTK, Unicast, MIC, Encrypted GTK)
EAPOL-Key(Unicast, MIC)
Install PTK and GTK
Install PTK
IEEE 802.1X Controlled Port Unblocked

Figure 24: The four-way handshake protocol between a host and a switch.

During the four-way handshake, the authenticator delivers a group transient key (GTK) to the host. The GTK is used for broadcast messages in the CA. The authenticator may update the GTK as needed. The GTK is derived by the authenticator using a group master key (GMK), the L2ID of the authenticator, and a $GNonce$ (a random or pseudo-random value). The GTK is encrypted with the KEK in the third message of the four-way handshake protocol. The group key updates done after the four-way handshake protocol require two handshake messages. The first message of the group handshake protocol sent by the authenticator delivers a new GTK to the host. This EAPOL-Key message contains the encrypted GTK, last transmit sequence number for the GTK, and the MIC computed over the body of the EAPOL-Key frame. The host sends an EAPOL-Key message in response. This message acknowledges the new group key and includes a MIC code.

6.2.3 Hosts to/from Hosts

Hosts utilize SAs with authenticators to secure data frames sent to hosts in other CAs. However, hosts can use host-to-host keys to secure data frames directly to other hosts in a CA. We utilize the STAKey handshake defined in the IEEE 802.11i standard [5] to create security associations between hosts. After each host establishes a SA with an authenticator, the authenticator transfers STAKey handshake messages between hosts. The originating host requests the STAKey by sending an EAPOL-Key frame to the authenticator with the L2ID of a peer host. The authenticator sends a STAKey message 1 to the peer host with the L2ID of the originator to provide a STAKey. The peer host responds to the authenticator sending a STAKey message 2 with the L2ID of the initiator host. The authenticator (after receiving the STAKey message 2 from the peer host) sends a STAKey message 1 to the initiator host with the L2ID of the peer host and the STAKey. The STAKey message exchange ends with a STAKey message 2 from the initiator host to the authenticator containing the L2ID of the peer host. In summary, the authenticator provides the key for both hosts to use for securing the connection.

6.2.4 Authenticators to/from Authenticators

In our security architecture, we assume that authenticators utilize a secure protocol to communicate. Authenticators create security associations and utilize a secure protocol to facilitate mobility and signaling. Security associations among authenticators provide protection to data frames transferred between hosts in different CAs.

6.3 Key Hierarchy

In our data link layer security architecture, the key hierarchy utilizes hash algorithms to derive PMKs and PTKs from a master key, as depicted in Figure 25. A master key is generated during the key establishment protocol between a host and an

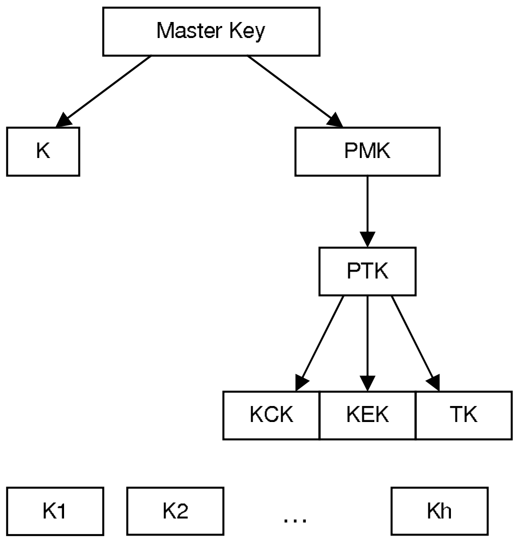

Figure 25: The key hierarchy of a host.

authentication server. A symmetric key K, which is used in the key establishment protocol between a host and autentication server, and a PMK, which is used during a four-way handshake protocol between the host and an authenticator, are derived from the master key. The PTK is composed of a KEK, KCK, and TK. The KCK and KEK are utilized in the four-way and group handshake protocols to provide data origin authenticity and confidentiality, respectively. The host and the authenticator use the TK as the CCMP key to communicate after the IEEE 802.1X controlled port is unblocked.

We utilize the HIP keying material derivation method, described in [77], to compute a PMK. The PMK is computed as in (10), where the $G2ID_{AS}$ and $G2ID_{Host}$ are 128-bit integers representing the global identifiers of an authentication server and a host, respectively. Both a host and an authentication server compute the PMK. The authentication server securely transports the PMK to an authenticator. Both the host and authenticator utilize the PMK to compute a PTK during the four-way handshake protocol, as described in the IEEE 802.11i standard. In (11), the PTK is computed using the L2IDs of the authenticator ($L2ID_{AA}$) and host ($L2ID_{Host}$).

The pseudo-random function (PRF), which is a function that hashes various inputs to derive a pseudo-random value, outputs 384 bits. A KCK is computed as the first 128 bits (bits $0-127$) of the PTK, while a KEK is computed as bits $128-255$ of the PTK. Finally, a temporal key (TK) is computed as bits 256-383 (for the counter-mode/cipher block chaining message authentication code protocol (CCMP)) of the PTK. The CCMP is based on the counter mode with cipher block chaining message authentication code (CBC-MAC) of the advanced encryption standard (AES) encryption algorithm. The counter mode with CBC-MAC (CCM), which combines counter mode for confidentiality and CBC-MAC for authentication and integrity, is a generic mode that can be used with any block-oriented encryption algorithm. CCM requires a fresh TK for every session. Note that the CCMP processing expands the original MAC protocol data unit (MPDU) size by 16 octets, which consists of 8 octets for the CCMP header field and 8 octets for the message integrity check (MIC) field (for further details, refer to [5]).

$$\begin{aligned} PMK = SHA1(masterkey \mid Min\,(G2ID_{AS}, G2ID_{Host}) \mid \\ Max\,(G2ID_{AS}, G2ID_{Host}) \mid PRN2 \mid J \mid 0 \times 01). \end{aligned} \tag{10}$$

$$\begin{aligned} PTK = PRF - 384(PMK \mid \text{“}Pairwise\ key\ expansion\text{”} \mid Min(L2ID_{AA}, L2ID_{Host}) \mid \\ Max(L2ID_{AA}, L2ID_{Host}) \mid Min(ANonce, SNonce) \mid Max(ANonce, SNonce)). \end{aligned} \tag{11}$$

In our data link layer architecture, since each host or creates security associations with other hosts in a CA, host-to-host keys are used for direct communications in the same CA. These keys are presented as $K1 - Kh$ in Figure 25. The STAKey EAPOL-Key exchange, described in [5], provides a mechanism to obtain these keys for direct host-to-host communication. A host in a CA obtains $h-1$ host-to-host

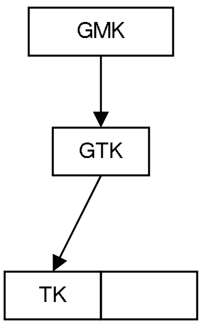

Figure 26: The group key hierarchy of an authenticator.

keys to transfer direct host-to-host frames, where h represents number of hosts in the CA.

$$GTK = PRF - 128(GMK \mid \text{“}\mathit{Group\ key\ expansion}\text{”} \mid L2ID_{AA} \mid GNonce).$$

An authenticator, in addition to the PTK, derive a GTK from a GMK as in (12), where the group nonce (GNonce) is a random or pseudo-random value. A GTK is partitioned into temporal keys used in the CCMP to protect broadcast/multicast communication. Figure 26 illustrates the relationship among the keys of the group key hierarchy.

CHAPTER 7

CONCLUSIONS

We have introduced a data link layer security architecture in IP over Ethernet networks with security inter-layering. In this architecture, we utilize data link layer identifiers which are flat names that can hold cryptographic meanings instead of fixed media access control (MAC) addresses. In addition, we modify and utilize protocols and design principles from various related work in our data link layer security architecture.

7.1 Security Analysis

In this section, we examine different links and protocols in our architecture to provide a security analysis of the introduced data link layer security architecture.

7.1.1 The Key Establishment Protocol

In our architecture, each host first performs mutual authentication with an authentication server using the key establishment protocol, as described in Chapter 5. Since authenticators provide communication between hosts and authentication servers, the link between a host and an authentication server is not a direct link.

The key establishment protocol is composed of four handshake messages. The handshake messages are carried in extensible authentication protocol (EAP) frames from a host to an authentications server and vice versa. However, regular data traffic is blocked from/to the host during the key establishment process. There are two links involved in this communication: host to/from authenticator and authenticator to/from authentication server. The link between a host and an authenticator is not secure during the key establishment protocol. While it is possible that authenticators and authentication servers utilize a secure protocol to carry EAP frames, we discuss security issues in both scenarios.

First, we assume that there is no protection for the key establishment protocol messages between authenticators and authentication serves. Note that this is the default case for the link between hosts and authenticators. The most vulnerable message of the protocol is the first message since it is not protected. However, the first message does not convey much information to a man-in-the-middle. The pseudo-random number, sent in this message, is later used with the global data link layer identifier (G2ID) of the host to create a local data link layer identifier (L2ID). Nonetheless, the key establishment protocol does not reveal the G2ID of the host to a man-in-the-middle. The first message contains an alias. After establishing a master key, the host sends its G2ID encrypted to the authentication server. Consequently, an attacker cannot learn the real identity of the host without the knowledge of the master key.

An attacker can easily observe, modify, or create the first message. For instance, an attacker can flood an authentication server with forged messages to cause denial of service (DoS) at the server. However, the authentication server does not create a state or compute the keys until it receives the third message. The authentication server responds to each message it receives by sending a message that contains a puzzle and the signature. This attack tries to overwhelm the computational resources of a server. An attacker can send forged messages to a host as well. This type of attack is impossible prevent. Nonetheless, the key establishment protocol utilizes signatures and a puzzle mechanism to detect and mitigate the attacks. Both the host and the authentication server detect forged messages by checking the signatures. In addition, the authentication server defers the keys generation process until it receives a puzzle solution. The authentication server verifies the solution of the puzzle before computing the keys and decrypting the third message. Even if a forged message contains a correct puzzle solution, the signature and decryption process will fail. Both the host and the authentication server should use timers to mitigate these attacks and allow valid messages to succeed.

Second, we assume that authenticators and authentication servers utilize a secure protocol to carry the key establishment protocol messages. In this scenario, the link between a host and an authenticator is the weakest link. The DoS attacks by message manipulations mentioned above are possible in this link as well. However, an attacker with access to the link between a host and an authenticator has a limited number of targets. Moreover, it is easier to detect the origin of the attacks if the link between hosts and authenticators is the only insecure link.

One potential attack in the key establishment protocol is replay attacks where an attacker records a series of messages and replays them at a later time. Specifically, if an authentication server is using the same pseudo-random number (PRN) and the puzzle value (k) for a time period, it should utilize a replay counter included in each message against replay attacks. Furthermore, a strong PRN generator should be used to prevent attackers from estimating PRNs.

Another concern in the key establishment protocol is man-in-the-middle attacks. The Diffie-Hellman exchange, utilized in the protocol, is vulnerable to man-in-the-middle attacks. To prevent these attacks, the public keys of authentication servers should be distributed to hosts in advance or advertised in the network. In addition, authentication servers should be capable of verifying the identities (public-private key pairs) of hosts. Finally, since the key establishment protocol utilizes the signatures of both parties, it prevents identity misbinding attacks.

7.1.2 Authenticators to/form Authentication Servers

In our data link layer architecture, we assume that authenticators and authentication servers create and utilize security associations. Security associations are necessary for transferring pair-wise master keys (PMKs) and other control/signaling information from authentication servers to authenticators. As discussed in the previous section, the key establishment protocol does not require this link to be secure. However, any data and signaling traffic should be protected with message integrity check and encryption methods. The attacks possible in the link between authenticators and authentication servers are mainly DoS and man-in-the-middle attacks. Most of these attacks are mitigated with the robust key establishment protocol and security associations.

7.1.3 Authenticators to/from Authenticators

In our data link layer architecture, security associations among authenticators are required to transfer data frames between hosts in different connectivity associations. A frame destined to another connectivity association within a realm is transferred to the corresponding authenticator. Security associations among authenticators provide confidentiality and message authentication properties to these frames. In addition, this link accommodates signaling among authenticators. For instance, events such as hosts moving among connectivity associations and new authenticators joining or leaving realms may require signaling among authenticators.

Possible attacks in this link include replay, DoS, and man-in-the-middle attacks. Authenticators should use replay counters and message authentication methods to mitigate these attacks. In this link, the most damage will be caused if an authenticator is compromised. An attacker will have access to group, pair-wise master and transient keys utilized in connectivity and security associations by the compromised authenticator. Although, an attacker can listen and modify traffic going through a rogue or compromised authenticator, an attacker will not be able to learn master keys generated by hosts and authentication servers during the key establishment protocol. Additional protection methods should be employed to prevent attackers from gaining control of authenticators.

7.1.4 Hosts to/from Authenticators

In our data link layer security architecture, the link between hosts and authenticators utilize security associations to protect frames. However, before a security association is created, this link is insecure. Specifically, during the key establishment protocol and the four-way handshake protocol the first messages are not protected. After a host and an authenticator complete the four-way handshake protocol, they utilize pair-wise transient keys to provide message authentication and data confidentiality. In addition, they should employ counters against replay attacks. Although DoS and man-in-the-middle attacks are possible in this link, they are easier to detect.

7.1.4.1 The Four-way Handshake Protocol

The four-way handshake protocol between hosts and authenticators provides means to create security and connectivity associations in our data link layer security architecture. Since the initial message in this protocol is not protected, it is vulnerable to attacks. The first message of the protocol can easily be forged. In our architecture, an authenticator learns the L2ID of a host from an authentication server securely. Since the L2ID of a host is not sent in cleartext in the key establishment protocol, an attacker does not know the L2ID of a host at the end of the key establishment protocol. The first message of the four-way handshake protocol is sent from an authenticator to a host. An attacker should be able to listen the first message from an authenticator and send the first message to a host modifying the $ANonce$ value before the third message. When the host receives more than one of the first messages, it computes a pair-wise transient key for each $SNonce$ and $ANonce$ pair. This may cause a memory exhaustion attack at the host. However, the host will be able to verify the legitimate third message by computing the message integrity check value because only the authenticator and the host possess the pair-wise master key. In [66], the DoS attack based on forging the first message of the four-way handshake protocol is further discussed and possible repairs are described.

7.1.5 Hosts to/from Hosts

In this architecture, two different types of links exist among hosts. First, hosts utilize security associations with authenticators to send frames to other hosts. Second, hosts may utilize host-to-host keys to send frames directly other hosts in the same connectivity association. The first type is not a direct link among hosts. The link between a host and an authenticator as well as the links among authenticators should be protected against replay, DoS, and message modification attacks. Utilizing security associations and employing replay counters and timers mitigate these attacks. The second type is a direct link between hosts. However, since in the STAKey handshake protocol, which is used to create security associations between hosts, an authenticator provides a key for both hosts to use for securing the connection, an authenticator can eavesdrop messages between hosts. The compromise of an authenticator enables an attacker to listen, modify, or create any host to host frames. In this architecture, we utilized the STAKey handshake protocol to create security associations between hosts to be compatible with the IEEE 802.11i standard. Nonetheless, a less vulnerable method of creating security associations between hosts can be utilized instead.

7.2 Contributions

We have introduced a new data link layer security architecture with security inter-layering. In this architecture, the security inter-layering concept allows the use of the same namespaces of upper layers in the data link layer of networks. First, instead of fixed MAC addresses, in this architecture, we utilize secure and flexible data link layer identifiers. We present methods to generate global identifiers from public keys (upper layer identities) and local data link layer identifiers from global identifiers.

Second, our data link layer architecture separates identities and locations supporting mobility and multi-homing. Note that our architecture modifies other internetworking layers as well. It requires the network layer to explicitly incorporate identifiers or identities in IP packets.

Third, the new data link layer security model with security inter-layering is incorporated into the MACsec. We address the establishment of secure associations and the key management in this architecture, which is not included in the scope of the IEEE 802.1AE standard. Nevertheless, to enable security inter-layering and secure identities at the data link layer, modifications to the IEEE 802.1AE standard are required.

Fourth, we propose the key establishment protocol to negotiate data link layer identifiers, establish security parameters, and mutually authenticate hosts and authentication servers. Furthermore, in the key establishment protocol, we address misbinding attacks protecting the identities of hosts in the data link layer. Moreover, in our key establishment protocol, we utilize a puzzle mechanism to thwart DoS attacks.

Fifth, we utilize the four-way handshake protocol and the key hierarchy of the IEEE 802.11i standard to be compatible with wireless networks addressing security between wireless and wired networks.

Sixth, we present the network structure providing link-to-link security with security and connectivity associations. This architecture requires all data link layer devices, such as switches/bridges, to own data link layer identifiers. Finally, we provide a security analysis of the network structure and the protocols.

7.3 Future Research

While the presented data link layer security architecture with security inter-layering is a significant step towards a secure and flexible network architecture, it creates new areas for further research.

- A distributed database architecture utilizing distributed hash tables (DHTs) can be incorporated into the architecture to track host identities and security associations and to enable mobility and multi-homing in local networks. In addition, directory or resolution services may be provided to map human-readable canonical names to flat names and locations.

- Performance of the algorithms and protocols may be evaluated. Specifically, the overhead created by the key establishment and four-way handshake protocols and security associations can be further studied.

- Security in the links between authentication servers and authenticators and among authenticators can be addressed in more detail. A new or modified EAP authentication method is required to accommodate the new key establishment protocol.

- The impact of the proposed architecture in networks and the Internet can be explored. Especially, the affects of the data link layer and global identifiers and implementation issues can be further investigated. For instance, VLAN double encapsulation methods can be used to accommodate dynamic data link layer identifiers.

- The public-private key distribution mechanisms are the main disadvantage of this architecture. Authentication methods utilizing pre-distributed keys as pairwise master keys can be incorporated into the architecture as an alternative approach.

REFERENCES

[1] *Secure Hash Standard*, April 1995. National Institute of Standards and Technology, Federal Information Processing Standards Publication FIPS PUB 180-1.

[2] *Advanced Encryption Standard (AES)*, November 2001. FIPS 197.

[3] *IEEE 802.1X-2001, IEEE Standards for Local and Metropolitan Area Networks: Port-Based Network Access Control (EAPOL)*, 2001.

[4] *Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*, July 2002. ITU-T Recommendation X.690.

[5] *IEEE Std 802.11i, Amendment to IEEE Std 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Security Enhancements*, June 2004.

[6] "NISCC vulnerability advisory IPSEC - 004033," May 2005. [Online]. Available: http://www.niscc.gov.uk/niscc/docs/al-20050509-00386.html?lang=en.

[7] "Anonymous Identifiers (alien)," November 2006. [Online]. Available: http://www3.ietf.org/proceedings/05aug/alien.html.

[8] "Arpwatch," November 2006. [Online]. Available: http://www-nrg.ee.lbl.gov/nrg.html.

[9] "IEEE 802.11 Standards," November 2006. [Online]. Available: http://grouper.ieee.org/groups/802/11/.

[10] "IEEE 802.1AE-Media Access Control (MAC) Security," July 2006. [Online]. Available: http://www.ieee802.org/1/pages/802.1ae.html.

[11] *IEEE 802.1AE, Standard for Local and Metropolitan Area Networks - Media Access Control (MAC) Security*, June 2006.

[12] "Netdisco," November 2006. [Online]. Available: http://www.netdisco.org/.

[13] "NewArch Project: Future-Generation Internet Architecture," November 2006. [Online]. Available: http://www.isi.edu/newarch/.

[14] "WiFi (Wireless Fidelity) Protected Access," November 2006. [Online]. Available: http://www.wi-fi.org/OpenSection/certification_programs.asp?TID=2.

[15] Aboba, B. and Dixon, W., *IPsec-Network Address Translation (NAT) Compatibility Requirements*, March 2004. RFC 3715.

[16] Aboba, B., "WEP2 Security Analysis." IEEE doc.: 802.11-00/253, May 2001.

[17] Aiello, W., Bellovin, S. M., Blaze, M., Ioannidis, J., Reingold, O., Canetti, R., and Keromytis, A. D., "Efficient, dos-resistant, secure key exchange for internet protocols," in *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, (New York, NY, USA), pp. 48–58, ACM Press, 2002.

[18] Altunbasak, H., Krasser, S., and Owen, H., "Network architecture at GaTech campus," Tech. Rep. M1, Submitted to Siemens, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, May 2004.

[19] Altunbasak, H., Krasser, S., Owen, H., Sokol, J., Grimminger, J., and Huth, H.-P., "Addressing the weak link between Layer 2 and Layer 3 in the Internet architecture," in *Proc. of the 29th Annual IEEE Conference on Local Computer Networks (LCN)*, (Tampa, Florida), November 2004.

[20] Altunbasak, H., Krasser, S., Owen, H. L., Grimminger, J., Huth, H.-P., and Sokol, J., "Securing Layer 2 in Local Area Networks," in *ICN*, vol. 2, (Reunion, France), pp. 699–706, April 2005.

[21] Arbaugh, W. A., "An Inductive Chosen Plaintext Attack Against WEP/WEP2." IEEE doc.:802.11-01/230, May 2001.

[22] Arbaugh, W. A., Shankar, N., Wan, Y. J., and Zhang, K., "Your 802.11 wireless network has no clothes," in *IEEE Wireless Communications*, December 2002.

[23] Arkko, J. and Haverinen, H., "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)." Internet-Draft, December 2004.

[24] Arkko, J., Kempf, J., Zill, B., and Nikander, P., "SEcure Neighbor Discovery (SEND)." IETF RFC 3971, March 2005.

[25] Arkko, J., Nikander, P., and Naslund, M., "Enhancing privacy with shared pseudo random sequences," in *Security Protocols, 13rd International Workshop*, (Cambridge), April 2005.

[26] Aura, T., "Cryptographically generated addresses (cga)," in *Proceedings of 6th Information Security Conference (ISC'03)*, vol. 2851, (Bristol, UK), pp. 29–43, Springer, October 2003.

[27] Aura, T., "Cryptographically Generated Addresses (CGA)." IETF RFC 3972, March 2005.

[28] Aura, T., Nikander, P., and Leiwo, J., "Dos-resistant authentication with client puzzles," in *Revised Papers from the 8th International Workshop on Security Protocols*, (London, UK), pp. 170–177, Springer-Verlag, 2001.

[29] Badra, M. and Urien, P., "EAP-Double-TLS Authentication Protocol." Internet-Draft, October 2005.

[30] Balakrishnan, H., Kaashoek, M. F., Karger, D., and Morris, R., "Looking up data in P2P systems," *Communications of the ACM*, vol. 46, pp. 43–48, Feb. 2003.

[31] Balakrishnan, H., Lakshminarayanan, K., Ratnasamy, S., Shenker, S., Stoica, I., and Walfish, M., "A layered naming architecture for the Internet," in *Proc. SIGCOMM '04*, (Portland, Oregon, USA), August-September 2004.

[32] Bashir, M. S., "ARP Cache Poisoning with Ettercap," August 2003.

[33] Bernard Aboba and Dan Simon, "PPP EAP TLS Authentication Protocol." IETF RFC 2716, October 1999.

[34] Berners-Lee, T., Fielding, R., and Masinter, L., *Uniform Resource Identifiers (URI): Generic Syntax*, August 1998. RFC 2396.

[35] Blake-Wilson, S. and Menezes, A., "Unknown key-share attacks on the Station-to-Station (STS) protocol," in *Lecture Notes in Computer Science*, vol. 1560, (Kamakura, Japan), January 1999.

[36] Blumenthal, M. S. and Clark, D. D., "Rethinking the design of the Internet: the end-to-end arguments vs. the brave new world," *ACM Transactions on Internet Technology (TOIT)*, vol. 1, August 2001.

[37] Blunk, L. J. and Vollbrecht, J. R., "PPP Extensible Authentication Protocol (EAP)." IETF RFC 2284, March 1998.

[38] Borisov, N., Goldberg, I., and Wagner, D., "Intercepting mobile communications: the insecurity of 802.11," in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 180–189, ACM Press, 2001.

[39] Bradner, S., Mankin, A., and Schiller, J. I., "A Framework for Purpose-Built Keys (PBK)." Internet-Draft, June 2003.

[40] Castelluccia, C., "Securing group management in ipv6 with cryptographically generated addresses," in *Wireless Personal Communications*, vol. 29, p. 221232, Kluwer Academic Publishers, 2004.

[41] Castelluccia, C. and Montenegro, G., "Securing group management in ipv6 with cryptographically generated addresses," in *ISCC '03: Proceedings of the Eighth IEEE International Symposium on Computers and Communications*, (Washington, DC, USA), p. 588, IEEE Computer Society, 2003.

[42] Castineyra, I., Chiappa, N., and Steenstrup, M., "The Nimrod routing architecture." RFC 1992, August 1996.

[43] Clark, D., Braden, R., Falk, A., and Pingali, V., "FARA: Reorganizing the addressing architecture," in *ACM SIGCOMM 2003 Workshops*, (Karlsruhe, Germany), August 2003.

[44] Clark, D. D., Sollins, K., Wroclawski, J., and Faber, T., "Addressing reality: An architectural response to real-world demands on the evolving Internet," in *ACM SIGCOMM 2003 Workshops*, (Karlsruhe, Germany), August 2003.

[45] Clark, D. D. and Tennenhouse, D. L., "Architectural considerations for a new generation of protocols," in *Proc. ACM SIGCOMM*, (Philadelphia, Pennsylvania, USA), 1990.

[46] Convery, S., "Hacking Layer 2: Fun with Ethernet Switches," 2002. [Online]. Available: http://www.arp-sk.org/doc/bh-us-02-convrey-switches.pdf.

[47] Convery, S., Miller, D., and Sundaralingam, S., "SAFE: WLAN Security in Depth." Cisco white paper, July 2002.

[48] Dean, D. and Stubblefield, A., "Using client puzzles to protect tls," in *10th Annual USENIX Security Symposium*, 2001.

[49] Donald E. Eastlake, "RSA/SHA-1 SIGs and RSA KEYs in the Domain Name system (dns)." IETF RFC 3110, May 2001.

[50] Droms, R. and Arbaugh, W., "Authentication for DHCP Messages." IETF RFC 3118, March 2003.

[51] Edney, J. and Arbaugh, W. A., *Real 802.11 Security: Wi-Fi Protected Access and 802.11i.* Boston, MA: Addison-Wesley, 2003.

[52] Faria, D. B. and Cheriton, D. R., "Dos and authentication in wireless public access networks," in *WiSE '02: Proceedings of the 3rd ACM workshop on Wireless security*, (New York, NY, USA), pp. 47–56, ACM Press, 2002.

[53] Floyd, S., "General Architectural and Policy Considerations." IETF RFC 3426, November 2002.

[54] Fluhrer, S., Mantin, I., and Shamir, A., "Weaknesses in the key scheduling algorithm of rc4," in *8th Annual Workshop on Selected Areas of Cryptography*, (Toronto), August 2001.

[55] FORD, B., "Unmanaged Internet Protocol: Taming the edge network management crisis," in *2nd ACM Workshop on Hot Topics in Networks*, (Cambridge, MA), November 2003.

[56] FRANCIS, P. and GUMMADI, R., "IPNL: A NAT-extended Internet architecture," in *Proc. SIGCOMM '01*, (San Diego, California, USA), August 2001.

[57] GAST, M. S., *802.11 Wireless Networks: Definitive Guide.* Sebastopol CA, USA: O'Reilly, 2002.

[58] GLAZER, G., HUSSEY, C., and SHEA, R., "Certificate-Based Authentication for DHCP," March 2003. [Online]. Available: http://www.cs.ucla.edu/ chussey/proj/dhcp_cert/cbda.pdf.

[59] GORDON, L. A., LOEB, M. P., LUCYSHYN, W., and RICHARDSON, R., "2004 CSI/FBI Computer Crime and Security Survey," 2004. [Online]. Available: http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2004.pdf.

[60] GRITTER, M. and CHERITON, D., "TRIAD: A New Next-Generation Internet Architecture," July 2000. [Online]. Available: http://www.dsg.stanford.edu/triad.

[61] GRITTER, M. and CHERITON, D., "An architecture for content routing support in the Internet," in *Proc. of the Usenix Symposium on Internet Technologies and Systems*, March 2001.

[62] HADDAD, W., NORDMARK, E., DUPONT, F., BAGNULO, M., PARK, S. D., PATIL, B., and TSCHOFENIG, H., "Privacy for Mobile and Multi-homed Nodes (MoMiPriv): Formalizing the Threat Model." Internet Draft, February 2005.

[63] HADDAD, W., NORDMARK, E., DUPONTAND, F., BAGNULO, M., PARK, S. D., and PATIL, B., "Privacy for Mobile and Multi-homed Nodes: MoMiPriv Problem Statement." Internet Draft, February 2005.

[64] HANDLEY, M. and GREENHALGH, A., "Steps towards a dos-resistant internet architecture," in *FDNA '04: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, (New York, NY, USA), pp. 49–56, ACM Press, 2004.

[65] HAVERINEN, H. and SALOWEY, J., "Extensible Authentication Protocol Method for GSM Subscriber Identity Modules (EAP-SIM)." Internet-Draft, December 2004.

[66] HE, C. and MITCHELL, J. C., "Analysis of the 802.11i 4-way handshake," in *WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security*, (New York, NY, USA), pp. 43–50, ACM Press, 2004.

[67] HORNSTEIN, K., LEMON, T., ABOBA, B., and TROSTLE, J., "DHCP authentication via Kerberos V." The Internet Society, Internet Draft, November 2000.

[68] Houle, K. J. and Weaver, G. M., *Trends in Denial of Service Attack Technology*, October 2001.

[69] Housley, R., Polk, W., Ford, W., and Solo, D., *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, April 2002. RFC 3280.

[70] Howard, C., "Layer 2 – The Weakest Link: Security Considerations at the Data Link Layer," 2003. [Online]. Available: http://www.cisco.com/en/US/about/ac123/ac114/ac173/ac222/about_cisco_packet_feature09186a0080142deb.html.

[71] Howard, C., "Layer 2 – The weakest link: Security considerations at the Data Link Layer," *PACKET*, vol. 15, First Quarter 2003.

[72] Karygiannis, T. and Owens, L., *Wireless Network Security 802.11, Bluetooth and Handheld Devices*, November 2002. Special Publication 800-48, Recommendations of the National Institute of Standards and Technology.

[73] Kent, S. and Atkinson, R., *IP Authentication Header*, Nov. 1998. RFC 2402.

[74] Krawczyk, H., "SIGMA: The 'SIGn-and-MAc' approach to authenticated Diffie-Hellman and its use in the IKE protocols," *Lecture Notes in Computer Science*, vol. 2729, pp. 400–425, Oct. 2003.

[75] Krawczyk, H., Bellare, M., and Canetti, R., "HMAC: Keyed-Hashing for Message Authentication." IETF RFC 2104, February 1997.

[76] Menezes, A. J., van Oorschot, P. C., and Vanstone, S. A., *Handbook of Applied Cryptography.* CRC Press, 1997.

[77] Moskowitz, R., Nikander, P., Jokela, P., and Henderson, T. R., "Host Identity Protocol." Internet draft, March 2006. Work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-hip-base-05.txt.

[78] Nikander, P., "Applying Host Identity Protocol to the Internet addressing architecture," in *Proc. of the 2004 International Symposium on Applications and the Internet (SAINT2004)*, (Tokyo, Japan), January 2004.

[79] Nikander, P., Laganier, J., and Dupont, F., "A Non-Routable IPv6 Prefix for Keyed Hash Identifiers (KHI)." Network Working Group Internet Draft, September 2005.

[80] Ornaghi, A. and Valleri, M., "Man in the Middle Attacks Demos." Blackhat, 2003. [Online]. Available: http://www.blackhat.com/presentations/bh-usa-03/bh-us-03-ornaghi-valleri.pdf.

[81] Plummer, D. C., "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware." IETF RFC 826, November 1982.

[82] Richardson, R., "CSI/FBI Computer Crime and Security Survey," 2003. [Online]. Available: http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2003.pdf.

[83] Rivest, R., Shamir, A., and Adleman, L., "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[84] Rouiller, S. A., "Virtual LAN Security: Weaknesses and counter measures," November 2006. [Online]. Available: http://www.sans.org/rr/papers/38/1090.pdf.

[85] Spangler, R., "Packet Sniffing on Layer 2 Switched Local Area Networks," December 2003. [Online] Available: http://www.packetwatch.net/documents/papers/layer2sniffing.pdf.

[86] Stephen Kent and Randall Atkinson, *IP Encapsulating Security Payload (ESP)*, Nov. 1998. RFC 2406.

[87] Stoica, I., Adkins, D., Zhuang, S., Shenker, S., and Surana, S., "Internet indirection infrastructure," in *Proc. SIGCOMM '02*, (Pittsburgh, Pennsylvania, USA), August 2002.

[88] Stubblefield, A., Ioannidis, J., and Rubin, A. D., "Using the fluhrer mantin and shamir attack to break wep," Tech. Rep. TD-4ZCPZZ, AT&T Labs, AUGUST 2001.

[89] Walfish, M., Balakrishnan, H., and Shenker, S., "Untangling the Web from DNS," in *Proc. of the 1st USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI '04)*, (San Francisco, CA), March 2004.

[90] Wang, X. and Reiter, M. K., "Mitigating bandwidth-exhaustion attacks using congestion puzzles," in *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security*, (New York, NY, USA), pp. 257–267, ACM Press, 2004.

[91] Waters, B., Juels, A., Halderman, J. A., and Felten, E. W., "New client puzzle outsourcing techniques for dos resistance," in *Proceedings of the 11th ACM conference on Computer and communications security*, pp. 246–256, ACM Press, 2004.

[92] Whiting, D., Housley, R., and Ferguson, N., *Counter with CBC-MAC (CCM)*, September 2003. RFC3610.

[93] Wolkstein, L., "Tussle in Cyberspace: Defining Tomorrow's Internet," September 2002. [Online]. Available: http://www.cs.duke.edu/ lmw8/cps182/tussle.txt.

[94] Xiaoyun Wang and Yiqun Lisa Yin and Hongbo Yu, "Collision search attacks on SHA1," February 2005.

[95] Ylitalo, J. and Nikander, P., "A new name space for end-points: Implementing secure mobility and multi-homing across the two versions of IP," in *Proc. of the Fifth European Wireless Conference, Mobile and Wireless Systems Beyond 3G (EW2004)*, (Barcelona, Spain), February 2004.