TOWARD ADJUSTABLE LIGHTWEIGHT AUTHENTICATION FOR NETWORK ACCESS CONTROL

Henric Johnson

Blekinge Institute of Technology
Doctoral Dissertation Series No. 2005:09

School of Engineering

Toward Adjustable Lightweight Authentication for Network Access Control

Karlskrona, December 2005
Department of Telecommunication Systems
School of Engineering
Blekinge Institute of Technology
S-371 79 Karlskrona, Sweden

Blekinge Institute of Technology
Doctoral Dissertation Series No. 2005:09

ISSN 1653-2090
ISBN 91-7295-077-3

Published 2005
Printed by Kaserntryckeriet AB
Karlskrona 2005
Sweden

This publication was typeset using LaTeX.

To Vicky and Wiktor

Abstract

The increasing use of Internet access networks raises the demand for secure and reliable communication for both users and businesses. Traditionally, the aim has been to provide the strongest possible security. However, with the demand for low-power computing it has become desirable to develop security mechanisms which efficiently utilize available resources. The tradeoff between performance and security plays an important role. In general, strong security is added even if there is no attack. The implementation of strong and resource demanding security often implies more than a secure system; it may deteriorate the performance of a device with limited resources and pave the way for new threats such as resource exhaustion. It is, therefore, unwise to use strong cryptographic algorithms for devices with limited resources in the absence of an adversary. It is more efficient to begin with lightweight security, taking further measures when an attack is detected.

The overall focus of this thesis is on adjustable and lightweight authentication protocols for network access control. The thesis studies the performance degradation of strong security using empirical tests on IP security (IPSec) with a visual bottleneck indicator based on the time-discrete fluid flow model and throughput histogram differences. The results emphasize the possibility of a Denial of Service (DoS) attack against IPSec itself.

The redundant authentication performed in a Wireless Local Area Network (WLAN) also motivates the development and evaluation of novel lightweight authentication protocols for the link and network layer. The developed authentication protocols are resource efficient, per-packet based, and robust in terms of handling packet loss. The protocols are further used as part of a hierarchical defense structure, which has been implemented and evaluated in order to mitigate protocol based DoS attacks.

Finally, this thesis presents the concept of Always Best Security (ABS) and a practical decision making model based on the Analytic Hierarchy Process. The model takes a number of factors into consideration, including subjective and objective aspects of security in order to select an adequate authentication level. It is a flexible model which formalizes quantitative and qualitative considerations of a defined set of criteria, keeping Quality of Service in mind.

Preface

This thesis reports on my research in the field of network security. The work has been carried out at the School of Engineering at Blekinge Institute of Technology. Most of the work has been conducted in collaboration with the security lab at the University of California Davis (UCDavis). Parts of the thesis material have appeared in the following publications:

1. H. Johnson, A. Nilsson, and M. Fiedler, Wireless Network Security, *In Proceedings of Nordic Radio Symposium Conference (NRS01)*, ISBN 91-631-0755-4, Nynäshamn, Sweden, April 2001.

2. H. Johnson, Security in Lightweight Ad Hoc Networks, *In Proceedings of Promote IT 2002*, Skövde, Sweden, April 2002.

3. H. Johnson, A. Nilsson, J. Fu, S. F. Wu, A. Chen and H. Huang, SOLA: A One-bit Identity Authentication Protocol for Access Control in IEEE 802.11, *In Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'02)*, Taipei, Taiwan, Vol 1, pages 768–772, November 2002.

4. F. Zhao, C. Shin, S. F. Wu, H. Johnson, R. C. Guo, T. J. Liu, K. P. Fan, and J. Fu, A Framework for Data Packet Access Control (DPAC), IETF draft <draft-zhao-pana-dpac-framework-00.txt>, February 2003.

5. B. Mårtensson, S. Chevul, H. Järnliden, H. Johnson, and A. Nilsson, SuxNet - Implementation of Secure Authentication for WLAN, Research Report 2003:03, ISSN: 1103-1581, Blekinge Institute of Technology, Sweden, December 2003.

6. F. Zhao, Y. Shin, S. F. Wu, H. Johnson, and A. Nilsson, RBWA: An Efficient Random-Bit Window-based Authentication Protocol, *In Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'03)*, San Francisco, USA, pages 1379–1383, December 2003.

7. H. Johnson, A. Nilsson, S. F. Wu and F. Zhao, Lightweight Authentication for Bluetooth, *In Proceedings of the first International Conference on Mobile Computing and Ubiquitous Networking (ICMU) 2004*, ISBN 4-902523-00-0, NTT DoCoMo R&D Center, Yokusuka, Japan, pages 106–111, January 2004.

8. H. Johnson, Lightweight Authentication in Wireless Networks, Licentiate Thesis, Blekinge Institute of Technology, Sweden, ISBN 91-7295-034-x, January 2004.

9. H. Johnson and A. Nilsson, Detection and Response Policies Using a Hierarchical Scheme, *In Proceedings of Promote IT 2004*, ISBN 91-85019-93-3, Karlstad, pages 258–268, May 2004.

10. S. F. Wu, H. Johnson, A. Nilsson, SOLA: Lightweight Security for Access Control in IEEE 802.11, *IEEE CS Journal IT Professional*, Vol. 6, No.3, pages 10–16, May/June 2004.

11. H. Johnson et al., A Report on Security Concepts for Mobile and Wireless IP Networks, Chapter 5, S. J. Knapskog, and M. Fiedler (eds.), *Euro-NGI: Deliverable D.JRA.6.3.1*, December 2004.

12. H. Johnson et al., Specification of a Key Management Protocol for Mobile Networks, Chapter 3, R. Lupu, S. J. Knapskog, and M. Fiedler (eds.), *Euro-NGI: Deliverable D.JRA.6.3.3*, May 2005.

13. H. Johnson et al., Assessment of Different Security Concepts for Mobile and Wireless IP Networks, Chapter 3, A Gutscher, S Kiesel, and M. Fiedler (eds.), *Euro-NGI: Deliverable D.JRA.6.3.2*, May 2005.

14. S. Hong, F. Wong, S. F. Wu, B. Lilja, T. Jansson, H. Johnson and A. Nilsson, TCPtransform: Property-Oriented TCP Traffic Transformation, *In Proceedings of GI/IEEE SIG SIDAR Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMWA 2005)*, Vienna, Austria, LNCS, Springer, July 2005.

15. H. Johnson, B. Qaisrani, M. Fiedler, S. F. Wu, and A. Nilsson, Analysis of IPSec Performance, *In Proceedings of Promote IT 2005*, ISBN 91-44-03875, Studentlitteratur, Lund, pages 259–269, May 2005.

16. H. Johnson, L. Isaksson, M. Fiedler, and S. F. Wu, A Decision System for Adequate Authentication, *accepted for publication in the International Conference on Systems (ICONS'06)*, IEEE Computer Society Press, April 2006.

17. H. Johnson, S. F. Wu, B. Qaisrani, M. Fiedler, and A. Nilsson, Hierarchical Defense System for Mitigating DoS Attacks, *accepted for publication in the IEEE 5th International Conference on Networking (ICN'06)*, IEEE Computer Society Press, April 2006.

18. L. Isaksson, H. Johnson, S. Chevul and M. Fiedler, Toward Seamless Integration of Wireless LAN and Cellular Networks, Research Report 2005:10, ISSN: 1103-1581, Blekinge Institute of Technology, Sweden, November 2005.

19. H. Johnson, M. Fiedler and S. F. Wu, Passive Overload Detection for VPN tunnels, *to be submitted for Journal publication.*

20. H. Johnson, M. Hansson, and Nicklas Isaksson, Efficient Image Recognition for Access Control, *to be submitted for publication.*

21. H. Johnson, M. Fiedler and S. F. Wu, Performance Evaluation of an Hierarchical Defense Structure, *to be submitted for publication.*

Acknowledgements

This journey would not have been possible to do by myself and I am eternally indebted to a great many people who helped to make this thesis a reality. I would like to thank my advisor Professor Arne Nilsson for talking me into this in the first place and for accepting me to the Ph.D program, introducing me to the world of research. I am sincerely grateful to my co-advisor Professor Shyhtsun Felix Wu at the University of California Davis (UCDavis) with whom most of this work has been conducted. Professor Wu has been both a personal and an academic inspiration, contributing many useful and productive ideas. Professor Wu provided an invitation to the UCDavis security lab as a visiting scholar, which proved to be a very fruitful experience. I am also grateful and happy for all the times Professor Wu visited me in Sweden, I can not thank him enough. I am also grateful to Dr. Markus Fiedler for always keeping his office door open for me and providing me with valuable suggestions and feedback. I would particularly like to thank Dr. Fiedler for his assistance with the latter part of my Ph.D work in which he gave me a lot of support. I also wish to thank docent Adrian Popescu for his valuable suggestions and comments during these years.

I express my thanks to all my fellow researchers in the group of telecommunication systems: Patrik Arlos, Doru Constantinescu, David Erman, Lennart Isaksson, Dragos Ilie, and Stefan Chevul. Special thanks goes to my friend Babar Qaisrani for all the long evenings in the security lab and the valuable research support.

This work was financed by the Knowledge Foundation of Sweden. I am grateful for the support I have received from them throughout the years.

As always, I am in debt to my family for their endless support and encouragement during my studies and Ph.D work. My parents, Bernth and Alva, for teaching me to always think positive and keep working, and to my brother Fredrik, his wife Sofia and their children Pontus, Jakob, and Gustav.

Finally, I express my infinite gratitude to my love and life companion, Vicky, for your endless encouragement, understanding and support. I dedicate this work to you and our son, Wiktor: you are my sunshine and energy and I love you both!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Network security is a natural and important part of today's society, playing a significant role for companies, organizations and individuals. Several network infrastructures connect to form the Internet and local access networks connect end-users with small and mobile devices in order to communicate from almost any physical location. As the need for this information technology increases, so too does the demand for adequate security measures. As a result, researchers and security engineers are struggling to address the intensified security issues.

Traditionally, the aim has been to provide the strongest security possible. By implementing strong security, in terms of sophisticated cryptographic algorithms, users feel confident in what they believe to be a secure system. However, the use of strong mechanisms may deteriorate the performance of a device with limited resources and pave the way for new threats such as resource exhaustion. The result is low Quality of Service (QoS) or even Denial of Service (DoS). This is sometimes called a *protocol based DoS attack* and is a serious threat to availability. Therefore, the tradeoff between performance and security plays an important role in digital communications and strong security should only be utilized as a last resort, in specific situations.

With the demand for low-power computing it has become desirable to de-

velop security mechanisms which efficiently utilize available resources. Bruce Schneier stated: "The future of digital systems is complexity, and complexity is the worst enemy of security" [1]. With this in mind, the overall focus of this thesis is lightweight authentication for access control which offers adjustable and resource-efficient authentication solutions in a constrained environment.

Another important aspect of security is the decision process involving which security level to select with regards to authentication. This decision is complicated since the criteria can be difficult to define and measure. A decision making model has to deal with quantitative and qualitative considerations of defined criteria. Authentication algorithms may vary in terms of security strength and required resources. Therefore, it is necessary to consider a compromise between different criteria.

1.2 Problem Statements

This section presents the problem statements and addresses the issue of efficient and adjustable authentication protocol provision for network access control.

Problem statement 1: What are the implications of strong and redundant security in a constrained environment?

Redundant security, in terms of unnecessary authentication, is usually resource-demanding [2]. With limited capacity, the security mechanism may introduce bottleneck behavior or become an easy target for DoS attacks, with devastating consequences.

Normally, a user has to decide what authentication level is necessary for a specific system. The decision process can be complicated by the number of criteria and alternatives which need to be compared and measured. This leads directly to the second problem statement.

Problem statement 2: Given a set of criteria regarding security attributes, for a set of security level alternatives, which is the best alternative for obtain-

ing the overall goal of adequate authentication?

When security is desirable in network communications, it is necessary to make a decision regarding which security level to use. A decision making model needs to be self-operating in terms of selecting the adequate security level, without introducing any unnecessary costs.

Problem statement 3: How can adjustable and lightweight authentication protocols be designed to provide efficient network access control, with minimal impact on available resources, whilst still accurately detecting ongoing attacks?

With regards to low-power computing and communication devices, the purpose of security mechanisms needs to be revised. If the objective is access control and the integrity is checked in a higher protocol layer, then there is no need to use expensive cryptographic functions with the payload as an input. Current and next generation communication networks require flexible and adjustable authentication mechanisms which utilize available resources to perform efficient access control.

Problem statement 4: How can we deploy lightweight authentication protocols for access control over a Local Area Network (LAN) and a Wireless Local Area Network (WLAN) to mitigate a DoS attack?

With the development of new access control methods, it is possible to manage and prevent a DoS attack using a hierarchical defense structure at the edge of a network. Such a structure may be utilized without becoming the target of a DoS attack and provides the following benefits:

- Prevents illegal activities by implementing a multilayer security system.
- Retains mission-critical response time during a DoS attack.
- Reduces the risk of downtime by protecting the security service.
- Ability to track normal and abnormal traffic in real-time.

Authentication features (and security mechanisms in general) may have an effect on the system capacity, as well as the QoS offered to end users.

Problem statement 5: How can efficient, proactive security be provided?

By the time an attack takes place, it might be too late to take countermeasures based on a reactive method. Instead, proactive security (which is able to detect an attack and take countermeasures) is preferable. Moreover, security is, in general, added even if there is no attack, which deteriorates the end-to-end performance. It is, therefore, unwise to use strong and resource demanding cryptographic algorithms for authentication for devices with limited resources, in the absence of an adversary.

1.3 Contributions

This thesis provides three important contributions which respond to the problem statements mentioned above:

- The first contribution deals with performance evaluation and mitigation of DoS attacks. An identification of IP Security (IPSec) [3] performance degradation is performed, in which latency and throughput are the major parameters of interest. The collected data is evaluated in order to search for bottleneck behavior for different traffic loads over an IPSec tunnel. To emphasize the obtained and critical behavior of IPSec, a visual bottleneck indicator is applied, which clearly depicts the possibility of a resource-exhaustion attack by the use of flooding traffic. This leads to the contribution of mitigating the flooding attack via a proposed hierarchical defense structure, with proactive functionality. A novel hierarchical architecture is presented with the proposed adjustable and lightweight authentication protocols acting as a classifier to deny access to harmful traffic. A prototype of the presented structure is implemented and results are reported which display the capability of the structure to filter and separate the attack traffic before reaching the target of an IPSec gateway. The considered IPSec environment is based

on IPSec gateways for the low-end market, i.e., for small businesses or private networks.

- The next contribution deals with the development of novel and lightweight authentication protocols which are payload-independent and per-packet based. These protocols function on the link layer (IEEE 802.11) and on the network layer. The proposed link layer protocols are *the blind* and *the non-blind authentication protocols* and the network layer protocol is *the Random-Bit Windows-based Authentication (RBWA)* protocol. The protocols are able to deny access to adversaries and to detect an attack by anomaly detection. Moreover, synchronization algorithms are developed to handle packet loss and create a robust and efficient authentication functionality.

- The final contribution is formulated in response to the second problem statement. A practical decision model is presented for finding the adequate authentication level based on desirable security criteria and alternatives. The process of decision making can be very complex. Therefore, the model presented by this thesis optimizes the selection of an adequate authentication level. Even though the notion of lightweight security is acknowledged, the process of knowing when to use it (in contrast to strong security) is not well developed. By defining an overall security goal and a set of criteria with corresponding alternatives the Analytic Hierarchy Process (AHP) [4] is used to finally select the most suitable and preferred authentication level.

1.4 Definitions

There is often a terminology problem in new research areas; different words appear with the same meaning. Throughout this thesis a number of concepts are used which are defined as follows.

Quality of Service: A broadly used term which refers to the performance attributes of an end-to-end connection. The general definition provided by

the International Telecommunication Union (ITU) [5] is that

> *QoS is the collective effect of service performance, which determines the degree of satisfaction of a user of the service.*

Denial of Service: Such an attack consumes the bandwidth or overloads the computational resources of the victim systems, typically resulting in a loss of network connectivity and services [6]. Most commonly, the attack attempts to "flood" a network with bogus packets, thereby preventing legitimate traffic.

Decision Making: Decision making is the process of sufficiently reducing uncertainty and doubt about alternatives and selecting a course of action among multiple alternatives. It should be noted that uncertainty is reduced rather than eliminated [6].

Adequate: As much, or as good as necessary for some requirement or purpose [7].

IPSec: IPSec [3] is a standard for securing Internet Protocol (IP) communications by encrypting and/or authenticating IP packets. IPSec provides security at the network layer and is further described in Chapter 4.

IEEE 802.11: IEEE 802.11 denotes a set of WLAN standards developed by working group 11 of the IEEE LAN/MAN Standards Committee (IEEE 802). The 802.11 family currently includes six over-the-air modulation techniques, all of which utilize the same protocol. The most popular (and prolific) techniques are those defined by the a, b, and g amendments to the original standard; security was originally included, and was later enhanced via the 802.11i amendment. Other standards in the family (c-f, h-j, n) include service enhancement and extensions, or corrections to previous specifications. 802.11b was the first widely accepted wireless networking standard, followed by 802.11a and 802.11g [6].

False Positive: A false positive exists when a test reports, incorrectly, that it has found a signal where none exists in reality [6]. Detection algorithms of

all kinds create false positives. For example, if an Intrusion Detection System (IDS) is configured to be highly suspicious, it will generate a number of false positives, incorrectly indicating an intrusion.

False Negative: A false negative, also called a miss, exists when a test reports, incorrectly, that a signal was not detected when, in fact, it was present [6].

Authentication Protocols: Authentication protocols are used to ensure that the entity which supposedly sent a message to another party is indeed the legitimate entity. The introduction to the international standard ISO/IEC 9798-1 [8] states the general purpose of an authentication protocol as follows:

> *Entity authentication mechanisms allow the verification, of an entity's claimed identity, by another entity. The authenticity of the entity can be ascertained only for the instance of the authentication exchange.*

This definition is drafted more in terms of personal interactions than in terms of communication between devices in a network. One has to pay particular attention to the word 'identity'. As cryptographic protocols are discussed in computer networks, the translation of 'identity' into 'cryptographic key' is more relevant for this thesis.

Adjustable: Having the ability to adapt and change the authentication level either by changing the number of authentication bits used per packet or the authentication protocol. The terms adjustable and tunable are in this thesis equivalent.

Lightweight: The term *lightweight* as used in this thesis refers to resource efficient authentication solutions which suit systems with constraints, such as low-power microprocessors with small processing capability, small memory, limited battery, and limited bandwidth. Sometimes these constraints make it impractical to use the majority of the current secure algorithms, which were developed for powerful devices. These constraints normally occur in mobile

environments with small communication devices. There is a growing interest in techniques that avoid strong cryptographic authentication and instead promote lightweight authentication [9].

1.5 Research Methodology

The research methodology described in this thesis is chosen in response to the problem statements. The research has been conducted with an awareness of both practical and theoretical concerns. Empirical tests and studies have been made in order to confirm the theoretical arguments. The research aimed to be systematic, replicable and logical, moving from questions to answers and was based on data collection, data analysis and data verification. Statistical methods have been used to analyze the collected data. In particular, statistical measures such as mean value, standard deviation, and confidence intervals for validation purposes were determined throughout the thesis. Information about the research methods is also included in the individual chapters.

1.6 Thesis Structure

Chapter 2 studies related work regarding performance measurements, decision processes, lightweight authentication, and DoS attacks and countermeasures. Chapter 3 emphasizes the redundant authentication exerted over a WLAN (IEEE 802.11), in which both IPSec and additional authentication can be used over the wireless link. Chapter 4 is devoted to a performance evaluation of IPSec degradation, with an implemented lab system able to generate and collect necessary data for further analysis. To illustrate the empirical tests, data analysis is performed using a proposed visual bottleneck indicator based on the *time-discrete fluid flow model* and throughput histogram difference plots. In Chapter 5, the adjustable and lightweight authentication protocols are presented with an evaluation of the proposed protocols following in Chapter 6. The evaluation is based on both the behavior of the integrated algorithms and various security considerations. The lightweight authentication protocols are applied as a first line of defense to mitigate DoS attacks. This

hierarchical structure is described by Chapter 7. The evaluation of the hierarchical architecture is performed, using an empirical test bed with connected IPSec gateways, forming a Virtual Private Network (VPN) between two local networks. Data is collected, with no influence on the monitored traffic, and is later analyzed with respect to throughput and CPU utilization. Chapter 8 addresses the selection of an adequate authentication level providing a solution to the problem of which authentication level to choose in the presence of different criteria. Chapter 9 concludes the thesis and outlines future work.

Chapter 2

State of the Art

This chapter presents prior and related work. It is divided into four parts: performance measurements, decision processes, adjustable and lightweight authentication, and DoS attacks and countermeasures.

2.1 Performance Measurement

Security related performance measurement is a broad research area. The focus in this section is limited to IPSec related performance evaluation.

In [10], Okhee and Montgomery examine the relative performance characteristics and dynamic behavior of large scale VPN environments based upon IPSec and the Internet Key Exchange protocol (IKE). They further introduce the NIST IPSec/IKE Simulation tool (NIIST) and use its detailed packet level model to characterize, for instance, the performance impact of cryptographic algorithms. Their results highlight the significant performance impact of IPSec implementation.

[11] presents a performance analysis of IPSec and a comparison of encryption algorithms (e.g., Data Encryption Standard (DES)) and one-way hash functions (e.g., Message Digest 5 (MD5) [12] and Secure Hash Algorithm 1 (SHA1) [13]) for authentication. The comparison is performed in terms of time and space complexity. Parameters considered in the study include process-

ing power and input size. The IPSec analysis revealed that HMAC-MD5 can be sufficient for authentication purposes, rather than the more complicated HMAC-SHA1.

A paper [14] presented by Barbieri, Bruschi and Rosti reveals the results of an experimental analysis of voice-over secure communication links implementing IPSec. In addition, the group presents the critical parameters characterizing the real-time transmission of voice over an IPSec connection, as well as techniques which could be adopted to overcome some of the limitations of Voice over IPSec (VoIPSec). Their results show that the effective bandwidth can be reduced by up to 50 % with respect to Voice over IP (VoIP) in case of VoIPSec. Furthermore, the cryptographic engine may damage the performance of voice traffic.

An interesting study is conducted in [15], in which an analysis of IPSec is made in high and low powered devices. The paper presents the performance cost brought up by the increased processing overhead.

In [16], Ronan et al. investigate the performance implications of IPSec deployment over a Wide Area Network (WAN). A similar investigation is performed in [17] with the focus on overhead issues for local access points in IPSec enabled VPNs.

2.2 Adjustable and Lightweight Authentication

The research area related to combining adjustable and lightweight authentication is limited.

Lindskog introduces interesting research regarding tunable security in his doctoral thesis [18]. In the thesis, methods are suggested for achieving different security levels for networked applications. Lindskog's work is restricted to the issue of confidentiality through selective encryption schemes which can be set to a desired security level in order to optimize the total system performance. The thesis contains both theoretical and practical methods and results.

In [19], Schneck and Schwan propose and implement an adjustable authentication protocol named Authenticast. It offers variable levels of secu-

rity throughput execution to client-server platforms, with varying numbers of clients and varying resource availabilities. Authenticast works with various heuristics used to enable adjustable authentication. The heuristics include: percentage-based authentication, delayed authentication, secret key connection and algorithm change. Experimental results are attained for a streamed Moving Picture Experts Group (MPEG) video sequence.

In [20], Ong et al. propose a Quality of Protection (QoP) framework that resolves the inadequacies of the one-or-nothing approach by providing differential security levels and quality awareness. The QoP parameters is dependent on security goals such as confidentiality, integrity, and authentication. However, these security goals are not easily measured [21]. The parameters of interests are the key length, block length, type of content, and the time interval during which the data must be kept secured.

Finally, Hager presents some interesting research in his doctoral thesis [22], in which he proposes a methodology which improves the efficiency of security mechanisms for wireless networks. This methodology can be used to define the relevant operational parameters of different wireless network applications, classify wireless networks into distinct categories, incorporate appropriate security protocols to a category, and analyse the security protocols through metrics. Another key contribution of the thesis involves the implementation and evaluation of a context-aware and adaptive security manager.

2.3 DoS Attacks and Countermeasures

DoS attacks have become an increasing security threat over the past two decades, meriting extensive attention by researchers and practitioners. This may be due to the fact that it is normally easier to disrupt the operation of a network or a service than to actually gain access.

The DoS attack (described by the attack model in Chapter 4) is one form of resource-exhaustion attack. This particular attack and its countermeasures (described by Chapter 7) are performed over a LAN and WLAN. There are, however, a range of other related DoS attacks. The corresponding countermeasures are as follows: protocol improvements, cookie-based approaches,

cryptographic puzzles, and theoretical work.

2.3.1 Protocol Improvements

The behavior of protocols may be a security problem; it is possible for an adversary to take advantage of malfunctioning protocols and launch a DoS attack.

SYN flooding [23] is a typical form of attack against the Transmission Control Protocol (TCP). The purpose of such an attack is to exhaust the victim through resource allocation. The attack could affect the HyperText Transfer Protocol (HTTP) and the File Transfer Protocol (FTP), two widely-used TCP-based protocols that form a significant part of the Internet. The attack works as follows: an adversary sends many TCP connection requests with forged source addresses to the victim's server. Each request packet causes the targeted host to draw on a limited resource pool. Finally, all the resources in the pool are exhausted and, as a result, no incoming TCP connections can be established. The attack has been analyzed in detail by Schuba et al., [24]. As a countermeasure, they suggest that firewalls can be configured as a semi-transparent gateway or a firewall with a relay functionality.

In the semi-transparent case, the firewall passes the SYN packets to the server. The next step is for the server to respond with a $SYN+ACK$ packet. The firewall forwards this packet to the client and then sends an ACK packet to the server. After that, the firewall waits for a legitimate ACK packet from the client. If the firewall does not receive a legitimate ACK packet within a timeout period, a reset (RST) packet is sent for termination to the server. If a legitimate ACK packet arrives at the firewall before the timeout is reached, all subsequent packets flow normally through the firewall. The advantage of this mechanism is that no extra delay is introduced for legitimate connection. The weakness of such a mechanism is that it may be difficult to find the correct timeout period, since connections with long response times could appear.

In the relay approach, the firewall contacts the server after the three-way TCP handshake is successfully completed and establishes a second connection. If an attack occurs, the firewall answers the SYN flooding from the adversary and since the ACK packet never arrives, the firewall terminates the

connection. However, this mode of protection is only useful if the firewall itself is not vulnerable to resource exhaustion. The drawback of this solution is that the firewall introduces new delays for legitimate connections, the obvious advantage being that the server is completely shielded from the attack and never receives spoofed *SYN* packets.

In [25], an admission control mechanism is described which drops a pending request with the goal of optimizing resource utilization. This mechanism can select a request at random if the connection-pending data structure is full. The authors of [25] further revised an analytical model for the random-drop mechanism. They used simulations to compare the random request dropping with the three other cookie-based, SYN flooding defense mechanisms described by Section 2.3.2.

In RFC 2267 [26], Ferguson and Senie present *ingress filtering*, which may prevent an adversary from using forged source addresses to launch a DoS attack.

The SANS Institute recommends that network administrators adopt a method called *egress filtering* [27]. This ensures that only IP packets with valid source addresses leave the network. However, one problem with this solution is that it might be difficult for ISPs to forward legitimate traffic which is not part of its own address space.

The *smurf* attack [28] is an attack in which a network acts as an amplification site to flood other networks with packets, as demonstrated by Figure 2.1. In order to defeat this attack, Senia [29] recommends that administrators block the receipt and forward *network-prefix-directed* broadcast on routers, since it is the nature of the broadcast mode which makes the attack powerful. A typical example of a *smurf* attack occurred at the University of Minnesota in 1998. Aimed at the University, the attack set off a chain reaction throughout the state, shutting down computers completely and, in other cases, causing data loss and network slowdowns.

2.3.2 Cookie-based Approaches

The primary goal of cookie-based protocols is to verify the authenticity of TCP connection requests since some attacks exploit an inherent weakness of

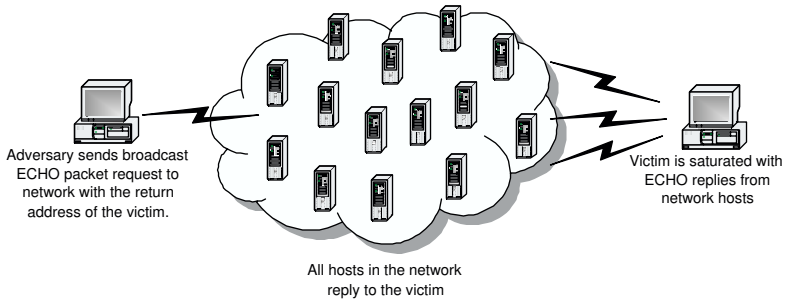


Figure 2.1: *Smurf attack.*

the TCP signaling behavior (i.e., the three-way handshake). Three cookie-based methods have been proposed: the *Berkeley cookies* [30], the *Linux cookies* [31], and the *reset cookies* [32]. The *Linux cookies* method was initially proposed by Bernstein and Bona. They suggested an approach in which a *SYN* packet is transmitted to a server which calculates a one-way hash of the *SYN's* sequence number and the addresses, combined with a secret number (changed at regular intervals). The server then sends the hash value back to the client as a sequence number within the *SYN+ACK* packet. So far, no record is kept of the connection request. If an *ACK* packet is subsequently received at the server, as a third step in the three-way handshake, the sequence number is used to authenticate the client. If properly authenticated, the connection is established. Otherwise the *ACK* packet is discarded.

Cookies have been further suggested by the Photuris protocol, developed by Karn and Simpson [33, 34], in which an authenticator is attached to the packets and must be verified before any other processing is done.

In [35], Zúguete improves the functionality of cookies. The current implementation of cookies does not support the negotiation of TCP options. The improvements suggested by Zúguete allow connections negotiated with SYN cookies to set up and use any TCP options, which reproduce the possibility to control TCP throughput performance.

2.3.3 Cryptographic Puzzle Approaches

The main idea with cryptographic puzzles is to require the client of a communication to commit its resources first. This idea was first introduced by Dwork and Naor [36], who suggested a computational approach to combatting electronic junk mail, by asking the client to solve a moderately difficult cryptographic puzzle for each received message. Juels and Brainard [37] further introduced the puzzle approach for connection attacks, such as the TCP SYN flooding attack. If the server is not under attack, the client of the communication has no puzzle to solve. However, if the server is under attack it will send a puzzle to be solved within a specified interval of time. The puzzle involves computing the reverse of a secure, one-way function. The cost of this computation for the client is determined by giving a number of input bits to the client and then it has to calculate the remaining ones.

The cryptographic puzzle solutions mentioned so far have concentrated on DoS attacks, using electronic junk mail and TCP SYN flooding attacks. They do not consider DoS attacks aimed at authentication protocols. However, in [38] they use the puzzle approach to design a DoS resistant mechanism useful to any authentication protocol, as illustrated by Figure 2.2.

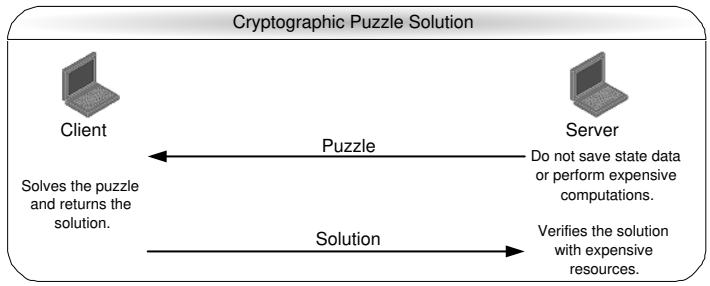


Figure 2.2: *Server suspecting a DoS attack sends a puzzle for the client to solve.*

In order to create new puzzles, the server periodically generates a *nonce*, which is random and unpredictable, like time stamps. This is done in order to prevent an adversary from precomputing the solution.

2.3.4 Theoretical Work

Meadows [39] has been working on formalizing the idea of gradually strengthening authentication and constructing a formal framework for network DoS. The purpose of the framework is to make protocols more resistant to DoS attacks by formalizing them. The goal, whilst designing a cryptographic protocol, is to specify the amount of resources a server allocates when its assurance of the client's identity and honesty increases.

Finally, Meadows [40] presents the way in which protocols can become more resistant to DoS attacks by trading off the cost to the defender versus the cost to the adversary. She also describes how this approach could be extended to protocols which do not make use of strong authentication. This formalization is based on the Gong and Syverson fail-stop model [41] of cryptographic protocols.

2.4 Decision Processes

The AHP [4] is a commonly used approach in the area of Multi-Criteria Decision Making (MCDM). The MCDM is a branch of general class-of-operations research models, which deal with decision problems in the presence of a number of criteria. There are several other methods under the MCDM and each method has its own characteristics. Two of the methods include the Weighted Sum Method (WSM) and the Weighted Product Method (WPM) [42], in which the WSM method is a commonly used approach.

The AHP method used in Chapter 8 has previously been applied in the area of risk management applications, such as: air traffic control [43], environmental risk assessment [44, 45, 46], national risk security [47] and information technology [48, 49, 50].

Another paper presenting challenges associated with selecting security technologies using decision theory is [51], in which the focus lies on software engineering practice. The paper describes two considerations that a researcher may have to address when adopting decision-theory techniques to make design choices. The first consideration depends upon whether the goal of the method was to replicate the security manager's security selections or to improve them.

The second consideration involves deciding which level and detail of information is necessary in order to make reasonable selections. Other publications in which decision making has been used in several software engineering settings include [52, 53, 54, 55].

Traditionally, security has only been expressed in a qualitative manner. However, with regards to decision making it is of interest to quantify security because of the possibility to add a security dimension to the QoS architecture. Several papers have been published on the quantification of security: [56, 57, 58, 59, 60, 61]. In [56], game theory is suggested as a method for modeling and computing the probabilities of the expected behavior of attackers in a quantitative model of security. An interesting example of the possible use of the model is provided by calculating the Mean Time to the First Security Breach (MTFSB) for a root privilege attack on a Unix system.

Chapter 3

Lightweight Security for Access Control in IEEE 802.11

3.1 Introduction

Network access technology raises new concerns regarding security; those who manage the communication networks must ensure that these networks do not introduce new vulnerabilities into the corporate network. The focus of this chapter is on lightweight authentication for network access control in WLAN (IEEE 802.11). The threat of unauthorized access in WLANs is significant because of uncontrollable signal propagation. This means that potential intruders, physically located outside a company, could access the company's internal information and network services. For instance, as Figure 3.1 illustrates, an unauthorized laptop can access local and remote critical servers via the WLAN.

Today, most enterprises are deploying a wireless infrastructure which is based on the IEEE 802.11 standard [62, 63, 64, 65], which, in turn, offers access to intranet or Internet data services.

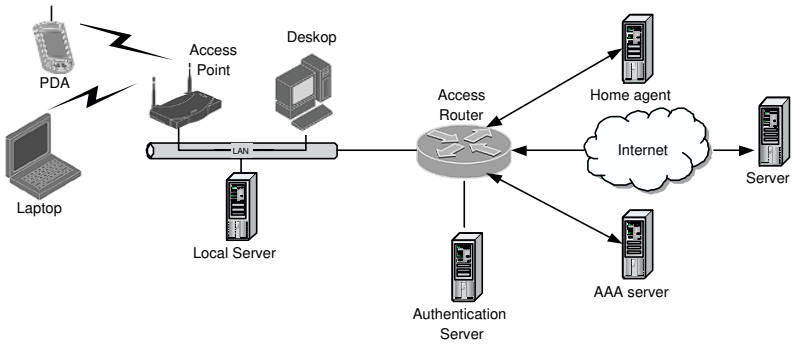


Figure 3.1: *Access route to local and remote servers via a WLAN.*

3.2 Today's Wireless Secure Access Options

Network administrators have many options when it comes to defending wired and wireless networks against unauthorized access. This section specifies several methods of securing access in IEEE 802.11. These include: the Wired Equivalent Privacy (WEP), Media Access Control (MAC) Address filtering, Service Set Identifier (SSID), and IEEE 802.11i.

3.2.1 Wired Equivalent Privacy

WEP is specified for encryption and authentication between the wireless client and the access point, and is based on the Rivest Cipher 4 (RC4) [66] encryption algorithm. The 802.11 standard describes two types of authentication services for 802.11 networks, the *Open System Authentication* and the *Shared Key Authentication*.

- *Open System Authentication:* A wireless network can use this service if it is not necessary to validate the identity of the sending mobile device. This is the default authentication protocol for 802.11, an authentication process requiring no key material.
- *Shared Key Authentication:* Shared Key Authentication provides a more secure authentication scheme than the open system procedure. However,

for a station to use shared key authentication, it must implement WEP. The 802.11 standard does not specify how to install the keys.

802.11 does not provide per-packet authentication, only encryption using WEP. Therefore, packet authentication is optional. However, a series of theoretical and practical attacks against WEP have been published [67, 68].

3.2.2 Media Access Control Address Filtering

Each access point can be configured with a list of MAC addresses which are allowed to access it. Furthermore, each MAC address is associated with a wireless client. The access point will deny access if the wireless client's MAC address is not on the aforementioned list. This is only a practical security solution if the network is small, since the work of manually updating lists of all the MAC addresses limits the scalability of this approach. Unfortunately, MAC addresses are easily eavesdropped, due to the fact that they appear in clear text. Also, most wireless cards provide the service of changing the MAC address via software, which makes it very easy to forge a valid MAC address.

3.2.3 Service Set Identifier

The SSID allows a network to be divided into multiple networks. In order to be able to access any of these networks, the wireless client must be configured with the correct SSID identifier. Several management frames contain the SSID identifier, and are broadcasted in clear text by the access point or the wireless client. However, this leaves the network vulnerable, as an attacker can easily forge the SSID, using it to gain access.

3.2.4 IEEE 802.11i

IEEE 802.11i provides better protection for wireless communication than which is displayed by Figure 3.1. IEEE 802.11i employs an authentication server, an entity which participates in the authentication of two or more wireless nodes, including the access points. The authentication server can authenticate the nodes itself, or it provides material for use by wireless nodes

to authenticate each other.

After authentication, using 802.11i, a device must also gain authorization for further service access. The core requirement for wireless network access is the verification of a wireless client's authorization to send and receive IP packets. Therefore, wireless networks need a back-end authorization infrastructure; one example is the Authentication, Authorization, and Accounting (AAA) server demonstrated by Figure 3.1. However, this chapter will discuss only the basic network access problem.

To summarize, the authentication and authorization process includes three basic stages, as Figure 3.2 demonstrates:

1. An initial authentication mechanism used in order to identify the valid user or client;
2. A key exchange and distribution procedure to mutually agree on a secret key between the access point and the client, which will then be used for subsequent activities, and;
3. A data packet authentication protocol (based on the secret key) for subsequent data communication.

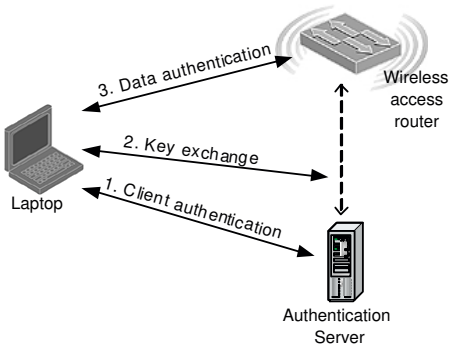


Figure 3.2: *Basic authentication and authorization process.*

3.3 Protecting the First Hop

If the wireless client has the correct credential for gaining access to the wireless network (stage 1), then all data packets passing to and from the authorized wireless node need some form of authentication. Otherwise, other unauthorized clients can eavesdrop the wireless media and discover what it takes to impersonate the client in order to gain access. Because such data authentication will apply to every bit in the communication, efficiency is an important consideration. Two options are considered. Firstly, the 802.11 working group is defining a link layer mechanism for use in authenticating all 802.11 frames with the Advanced Encryption Standard (AES) [69], and specifies the use of Counter for Cipher Block Chaining Message Authentication Code (CCM) [70]. CCM is a method for encrypting blocks of packet information. Without using efficient authenticated-encryption modes, such as CCM, a client-server pair would need to encrypt and authenticate the packets separately, thereby paying the cryptography-related cost twice. Another option is to authenticate the IP packets in layer 3 (the network layer), using the IPSec protocol suite. If the authentication process enforces access control in the network layer (instead of link layer), we usually call the enforcement point "a layer 3 access router" (instead of a layer 2 access point).

As Figure 3.3 demonstrates, in this form of authentication, the wireless client (following the IPSec standard) protects the TCP/UDP header and payload by encrypting or authenticating them. If only authentication is necessary, the client authenticates the packet payload (including the transport header) and adds an authentication header to the original packet. On the other hand, if both authentication and encryption are necessary, the client encrypts the packet payload (again, including the transport header), and the server authenticates it following the IPSec ESP standard. In IPSec/ESP, the encryption set adds an ESP trailer and an ESP header. The client subsequently authenticates the encrypted payload, plus the ESP trailer and header all together, and appends an ESP authentication header after the trailer.

Both options *only* protect the packets within the first hop of an end-to-end communication; that is, from the wireless client to the access point (in the case of 802.11i) or to the access router (in the case of IPSec).

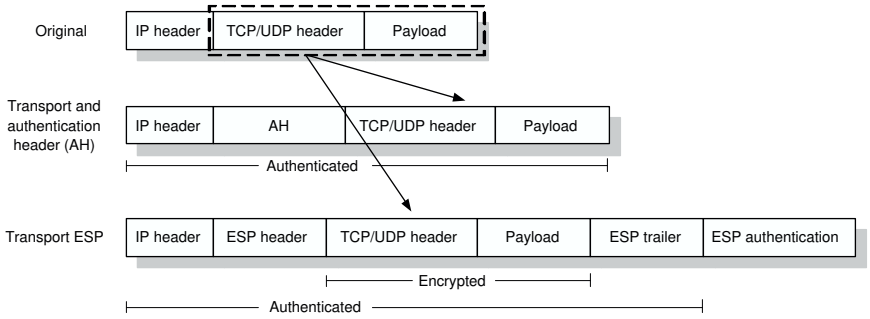


Figure 3.3: *Network-layer IPSec packet authentication: original packet, packet with added authentication header, and final packet.*

3.4 End-to-End Security

For many Internet-based enterprise applications – wired or wireless – it is necessary to provide end-to-end security in terms of encryption and authentication. Or, at the very least, these applications must have some kind of a VPN from the wireless or mobile client to its home network gateway. If an employee of a financial firm reads her confidential e-mail using the *XYZ Mobile* service provider in a particular airport, then the communication channel must ensure that all the messages are protected as they pass from the wireless client to the enterprise Simple Mail Transfer Protocol (SMTP) server. Having only first-hop protection is insufficient for almost all critical applications because, today, adversaries can actually possess accounts on routers in the Internet Service Provider's (ISP's) network. Rob Thomas reported that, as in June 2003, around 5,310 commercial routers were compromised [71]. Popular options for end-to-end security or VPNs include (at least): IPSec, Secure Sockets Layer-Transport Layer Security (SSL-TLS) [72, 73], and Secure Shell (SSH) [74]. Combining first-hop (802.11i) and end-to-end (IPSec) security considerations produces two options, as demonstrated by Figure 3.4:

1. In the first option, the wireless client encrypts the data packet using IPSec and then encrypts the IPSec encrypted packet again using 802.11i. With a proper key exchange, the access point can decrypt the packet

using 802.11i. If the decryption and authentication process is successful, the access point will forward the packet to the Internet and towards the destination host. Otherwise, it will consider the packet illegal and drop it.

2. In the second option, the client encrypts the packets only once, using 802.11i. The access point will perform the same decryption procedure. If successful, the access router will use IPSec to protect the packet before sending it to the Internet.

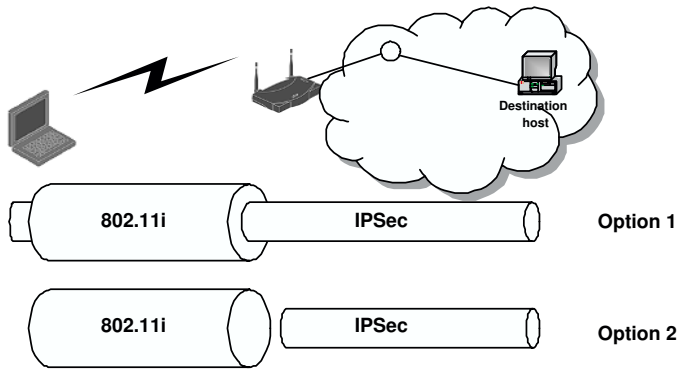


Figure 3.4: *802.11i and IPSec/VPN solution in an 802.11 network.*

The second option, in many applications, might not be desirable from a security point of view. It, essentially, assumes that the wireless client (and its corresponding server) can trust the wireless access router or access point because the wireless access router can access the original unencrypted messages. It is quite obvious that *XYZ Mobile* should use option one.

The first option makes two security associations. The outer association (802.11i) serves to protect the first hop, from the wireless client to the access router. Without end-to-end protection (the inner association: IPSec), this is important for the client in order to protect privacy and authenticity due to the great concerns surrounding security over wireless media. What if the application requires end-to-end security? From the client's perspective, the

802.11i protection might well be redundant, because many important enterprise applications require end-to-end security. This first-hop protection is still necessary from the viewpoint of *XYZ Mobile*. Otherwise, nonpaying or unauthorized users can still illegally access the service, and *XYZ Mobile* might lose significant revenue. In some sense, wireless clients are helping *XYZ Mobile* to ensure that only paying users can utilize its wireless infrastructure and Internet access. The drawback, at least for wireless or mobile clients, is that they must encrypt all data packets and authenticate them twice.

Under the three-stage authentication model, described earlier by Figure 3.2, client authentication (stage 1) and secure key exchange (stage 2) are both necessary to initially grant network access to wireless or mobile clients. However, after these two stages, offering efficient, high performance end-to-end communication is very important. For instance, a cellular phone with an 802.11 module lets its user conduct telephone calls using IP packets when 802.11 access points are nearby. To conduct the call using IP, it might be necessary to run IPSec from the client cellular device to an IP telephone gateway. Under QoS and processing considerations, the extra or redundant 802.11i encryption might unnecessarily degrade the end-to-end performance. In general, redundant protection in the first hop is undesirable because many mobile or wireless clients have few CPU resources.

The need for first-hop protection is also arguable since certain applications need end-to-end protection while others do not. The former already have well protected confidentiality and authenticity, and first hop protection is used only for network access control. For example, *XYZ Mobile* needs this mechanism (even at the cost of user resources) to ensure revenue collection for the service. In contrast, it is reasonable for less sensitive applications, such as Web surfing, to use options such as 802.11i or IPSec during the first hop.

3.5 Adequate Security

It is desirable to develop another security option for the first hop which doesn't produce unnecessary costs for mobile devices, and still permits wireless access service providers to collect the information that they need to reliably charge

for their services. The main problem is how the access point will determine whether or not a particular data packet is from a valid user. Strong cryptographic mechanism such as AES can determine whether a packet is good or bad with almost 100 percent probability. The question is whether 100 percent accuracy is really necessary when the weak wireless nodes need to increase the cost in transmitting every data packet. In fact, it is acceptable to have a small percentage of packets which originate from invalid users; especially in order to save resources. Hypothetically speaking, if no adversary attempted to steal bandwidth from *XYZ Mobile*, then all of the AES work in the first hop was unnecessary. Therefore, an inexpensive scheme to detect the presence of such adversaries permits *XYZ Mobile* to shift gears, i.e., returning to a more expensive protection mechanism only when necessary.

Such a lightweight security option should serve two purposes. Firstly, this inexpensive security mechanism will successfully authenticate most of the packets in the first hop, even in the presence of malicious attackers. Secondly, the same mechanism must also offer high accuracy attack detection, so that users and service providers can shift security modes in order to handle the problems.

Normally, any authentication mechanism will need to take all bits from the payload as input in order to produce a valid message authentication code which will protect the payload's integrity. However, if the receiver will check payload integrity at the final destination, it might not be necessary to perform a strong check at the first hop, because the main purpose of the first hop is authenticating the origin identity, *not* the payload. Therefore, unlike 802.11i and IPSec, which use every bit in the data packet to compute the final message authentication code, the developed and novel authentication protocols in Chapter 5 are *payload independent*.

Chapter 4

Evaluation of IPSec Performance Degradation

This chapter evaluates IPSec [3] performance degradation and describes different performance testing procedures. It further contributes to the understanding of the tradeoff between performance and security.

4.1 Introduction

IPSec has been developed to provide end-to-end security at the IP level. It is an extensible and complete security service that uses authentication and encryption to ensure non-modification and secrecy of the contents. IPSec can protect any protocol running above the IP layer such as TCP, UDP, and Internet Control Message Protocol (ICMP) among others.

During recent years there has been a rise in the number of available security services and solutions. Various security services and mechanisms appear as patches for software, since the IP stack has not been designed with security in mind. On one hand, they eliminate immediate, forseen threats, but on the other hand they pave the way for new threats in the system. Increased security requirements have a high cost in terms of resources. Therefore, it is

essential to develop security mechanisms which require a comparably small amount of resources to perform necessary calculations.

An optimal communication link should be transparent and not introduce any loss or delay. However, this is not the case after an IPSec tunnel is established. Over an IPSec link, the computational overhead in the gateways brings about the end-to-end throughput and adds significant latency or packet drop. The IPSec gateways encrypt outbound traffic and decrypt incoming traffic. Encryption, decryption and key generation algorithms are by nature computationally intensive. For service providers and individual users, the computational overhead directly affects the QoS and may bring down the network performance in general. Therefore, the objective is to analyze IPSec performance and further determine if IPSec is vulnerable to a DoS attack due to demanding processing overhead for each packet. The results presented in this chapter are based upon empirical studies with implemented test beds which measure latency, throughput and bottleneck behavior. A bottleneck indicator is also introduced which helps to visualize the impact over an IPSec tunnel, for different traffic loads. The IPv4 protocol stack is used, and a number of user categories are presented in terms of bandwidth consumption. Different cryptographic algorithms and operating systems such as Windows XP Professional Edition and Linux/Debian are employed.

This chapter is organized as follows: In Section 4.2 the attack model is described. Section 4.3 gives an overview of IPSec and Section 4.4 describes the measurements, results and testing procedures used to evaluate IPSec latency performance and throughput performance. Finally, Section 4.5 concludes the chapter.

4.2 Attack Model

The target system of this study is an IPSec gateway for the low-end market. These devices are gaining importance as the number of network-enabled home devices increases [75, 76]. The performance evaluation is based on the IPSec tunnel mode. The interest in the tunnel mode is mainly due to the possibility of launching a DoS attack against the IPSec gateway, due to resource

exhaustion (as presented by Chapter 7).

The attack model is described as follows: an adversary is associated to the same WLAN or LAN as the targeted IPSec gateway (GW1), as illustrated by Figure 4.1. Between GW1 and GW2 an IPSec tunnel is established to secure traffic over the public network between the different LANs or WLANs. If an adversary transmits a large number of malicious packets to the IPSec GW1, these must be verified and authenticated, which will drain the constrained resources in GW1. Therefore, in securing the system against one type of attack, another form of attack is promoted; namely, resource exhaustion.

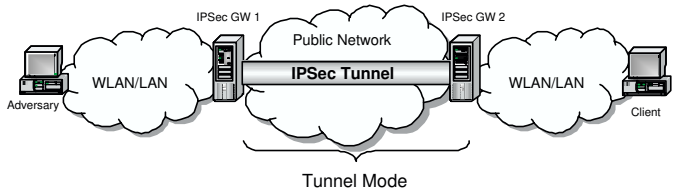


Figure 4.1: *Attack model.*

4.3 IPSec Overview

Traditional approaches to network security have been application-dependent solutions. Nevertheless, a network-layer security solution has been accepted as a necessary complement to a multilayer security architecture. There have been early attempts to network-layer solutions in [77, 78, 79]. In 1992, the Internet Engineering Task Force (IETF) began designing a protocol well suited for the Internet environment, in which IP [80] is used as the network protocol. One of the first experimental attempts was [81], in which the objectives of network-layer security were tested. A few years later (1995), the IETF IPSec Working Group developed a set of specifications [3]. These specifications were also adopted by the IPv6 [82] Working Group for the next generation Internet network protocol. The IETF developed IPSec, which is a framework of different open standards which provide data confidentiality, data authenticity,

and data integrity between communicating devices. There are a number of Requests For Comments (RFCs) and Internet Drafts that describe the IPSec architecture [83, 84, 85, 86, 87].

4.3.1 IPSec Fundamentals

IPSec establishes standards for a range of services to address security risks for IP traffic across the network. IPSec is able to handle confidentiality, access control, authentication, and rejection of replayed packets.

If the IPSec mechanism resides on an intermediate host, that host is termed an IPSec security gateway. The IPSec mechanism applies a Security Policy Database (SPD) to decide how to handle packets. Possible causes of action include: forwarding the packet with no change, applying security services to the packet, and discarding the packet. The SPD determines which action to take based on the characteristics of the packet. These characteristics include, for instance, the source and destination port and address, and the transport layer protocol involved.

IPSec contains several mechanisms with various functionality. The following fundamentals are designed by IPSec: security protocols, key management, security associations and algorithms, as described by the following:

Security Protocols: Within the IPSec protocol suite, two security protocols are defined for traffic security: the Authentication Header (AH) protocol [88] and the Encapsulating Security Payload (ESP) [89, 90]. The AH provides data authentication and an optional anti-replay service to discard replayed packets. Furthermore, the AH ensures the integrity and data origin authentication of the IP packet as well as of the invariant fields in the outer IP header. The ESP provides encryption and/or integrity protection. The set of services provided depends on options selected at the time of a Security Association establishment and on the location of the implementation in the network. It is only the headers and data behind the IP header that ESP authenticates.

Security Associations (SA): The concept of an SA is fundamental to IPSec. An SA is a relationship between two or more entities which describes how security services will be used in order to communicate securely. The SA includes an authentication algorithm, an encryption algorithm and a shared session key. An SA is unidirectional, i.e., two SAs are required for secure communication between two entities. The SA is uniquely identified by the Security Parameter Index (SPI) [89], which is located in the IPSec header. There are two types of SAs: *Transport mode SA* and *Tunnel mode SA*. Transport mode refers to adding the IPSec information between the IP header and the remainder of the packet, as illustrated by Figure 4.2.

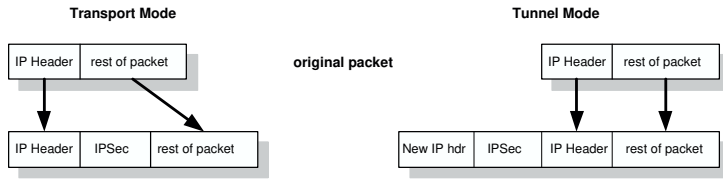


Figure 4.2: *Transport mode and tunnel mode.*

Transport mode is mostly used when IPSec is being applied end-to-end and tunnel mode is commonly used to protect the data along a part of the path between the endpoints. This is further depicted by Figure 4.3, which shows two IPSec hosts and two IPSec gateways. An IPSec gateway can be implemented in software on a server, router, firewall or a security appliance. The IPSec gateway should handle the high-speed encryption/decryption, the IPSec policies negotiation, and the tunneling services.

Key Management: IPSec supports two different methods of key establishment and SA management, the *Manual Key Management* and *Automatic Key and SA Management*. Manual key management is suitable for a small and static number of hosts in a network. For larger scenarios, Automatic Key and SA management is achieved by using the Internet Key Exchange (IKE) protocol, which combines parts of the defined protocols in [91, 92, 93].

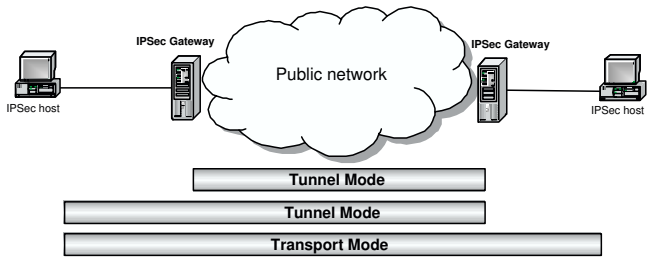


Figure 4.3: *AH and ESP operational modes.*

Algorithms: There are no defined authentication or encryption algorithms that are mandatory for IPSec. However, there are individual RFCs for algorithms that could easily be used in the IPSec protocol suite. For instance, [17] describes how ESP [89] using AES [94] is recommended in terms of security and performance. There are, however, other algorithms (i.e., 3DES or SHA1) that could be used to best fulfill the requirements for a specific context transfer.

4.4 Measurement Procedures and Test Bed Configurations

The key metrics involved in evaluating IPSec performance degradation are latency and throughput. A bottleneck indicator test is performed to describe the input and output traffic over an IPSec tunnel. Moreover, a visualization of the measurements is introduced to better describe the bottleneck behavior of IPSec.

4.4.1 Latency Test

To examine the effect and latency of IPSec over a wired link, an experimental test bed is configured (Figure 4.4). The test bed consists of three IPSec enabled gateways. An IPSec tunnel is established from GW1 (10.10.1.1) to the other two gateways, namely GW2 (10.10.1.2) and GW3 (10.10.1.3). The

equipment and configuration involved in the latency test bed are as seen in Figure 4.4: GW1 and GW2 contain a 2 GHz processor and 512 MB of RAM and GW3 (10.10.1.2) includes a 300 MHz processor and 64 MB of RAM. All gateways are running Windows XP Professional Edition. The client (192.168.1.2) also contains a 2 GHz processor and 512 MB of RAM with Linux/Debian (kernel 2.4.6) and is connected to GW1 through and Ethernet switch. The reason for using different end gateways is to compare the effect of processing power on IPSec. A 100 Mbps Ethernet local area network is used to connect the devices.

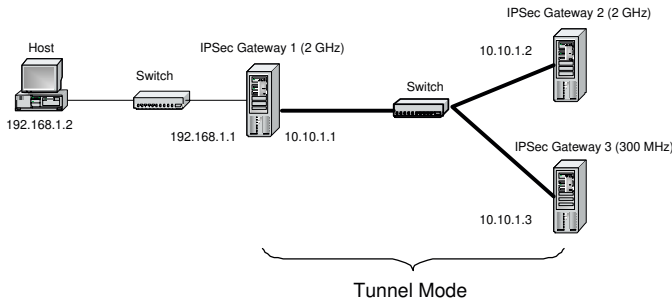


Figure 4.4: *Latency test bed.*

An IPSec rule is configured in GW1 to encrypt/decrypt all traffic from and to the client. In effect, the client's traffic is secured after passing through GW1. Manual keying is used to avoid certificates, and IP forwarding is enabled in GW1 so that it could act as a router. An ICMP (ping) script is started from the client to measure the latency with varying packet sizes of 512, 1024, 4096, and 8192 bytes of payload. To obtain the presented results each test outputs a mean result X_i of the response time obtained from 100 ping packets transmitted from GW1 to GW2 or GW3. The confidence interval is then given by

$$\hat{X} \pm t_{n-1,1-\alpha/2} \frac{\delta}{\sqrt{n}} \tag{4.1}$$

where $\hat{X}$ (the value plotted in Figure 4.5 and Figure 4.6) is the average of

$n = 40$ mean results X_i, δ is the standard deviation and $t_{n-1,1-\alpha/2} \simeq 2.02$ for a confidence level of $1 - \alpha = 95$ %. The 95 % confidence intervals are presented in Appendix A.

The ICMP script does not measure the first few packets that experience excessive delays because of the establishment of the SA. Several encryption (DES and 3DES) and authentication (SHA1 and MD5) algorithms are tested, as illustrated by Figure 4.5 and Figure 4.6. It is also noted that Windows XP Professional Edition does not have the Advanced Encryption Standard (AES) implemented as compared to the Linux counterpart Openswan [95].

Figure 4.5 and Figure 4.6 illustrate the increased response time due to the performance cost induced by the processing overhead of each packet. As expected, the combination of encryption and authentication causes a major bottleneck. The response time of GW3 (300 MHz) is larger (especially for packets with increased size) as compared to GW2 (2 GHz). The maximum response time for the 2 GHz gateway is about 7.7 ms and about 19 ms for the 300 MHz gateway.

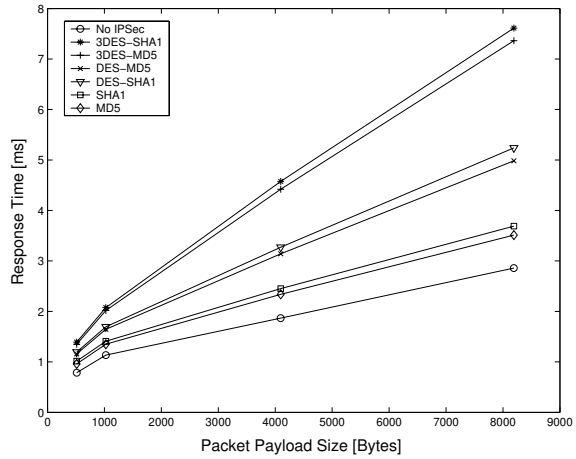


Figure 4.5: *Ping response time with a 2 GHz IPSec gateway.*

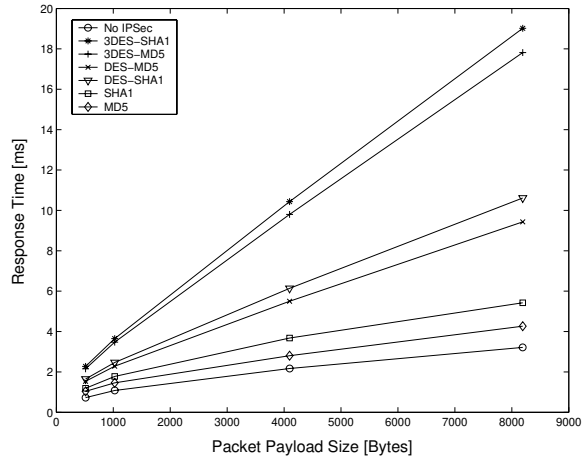


Figure 4.6: *Ping response time with a 300 MHz IPSec gateway.*

4.4.2 Throughput Test

The throughput tests are performed as described by section 10.1.1 of [96] and are defined as the obtained maximum rate through an IPSec tunnel at which none of the offered packets are dropped by the device. The test bed consists of two IPSec GWs, namely GW1 (10.10.1.1) and GW2 (10.10.1.2), as illustrated by Figure 4.7. To measure the throughput, Netperf [97] is used due to its flexibility to configure and control the message size and the socket buffer size. Netperf measures throughput based on UDP and TCP traffic.

Netserver is installed to receive the Netperf traffic at 10.70.1.2 and the client is established at 192.168.1.2. All traffic from the Netperf client to the netserver passes through GW1 and GW2. The equipment and configuration involved in the throughput test bed are as follows: the two IPSec gateways contain a 2 GHz processor running Windows XP professional with 512 MB of RAM, the client includes a 1 GHz processor running Linux/Debian (kernel 2.4.6) with 512 MB of RAM and the 10.70.1.2 (Netserver) contains a 2 GHz processor with the same configuration as the client.

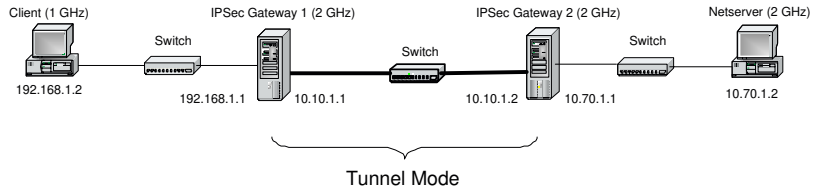


Figure 4.7: *Throughput test bed.*

IPSec is configured between GW1 and GW2. An IPSec rule is established in GW1 to encrypt/decrypt all traffic sent to and from the client. Mirrored parameter rules are configured in GW2. Moreover, the Network Time Protocol (NTP) [98] is used to synchronize the system clocks between the sender and the receiver.

In Figure 4.8, the TCP throughput over the 100 Mbps Ethernet link is illustrated for the following cryptographic algorithms: 3DES-SHA1, 3DES-MD5, DES-SHA1, DES-MD5, SHA1, and MD5. In the test a packet payload size of 1000 bytes is used. The test is performed until a 95 % confidence

interval is obtained [97]. The throughput is dependent on which algorithm IPSec uses. Therefore, the degradation of the 100 Mbps throughput occurs due to the overhead introduced by the resource-demanding algorithms which handle the packets for the IPSec tunnel. The results illustrate that 3DES-SHA1 has the highest influence on the throughput and MD5 the lowest.

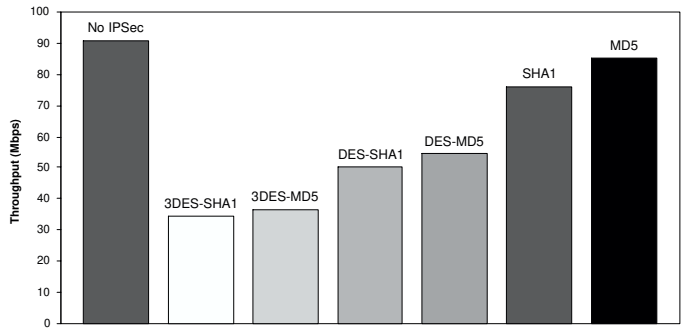


Figure 4.8: *TCP throughput for the 100 Mbps link.*

To avoid packet fragmentation, the optimal payload size for TCP and UDP packets is determined as follows:

Optimal Payload Size (OPS): Before IPSec is enabled, the Maximum Segment Size (MSS) of an IPv4-over-Ethernet packet amounts to 1460 bytes and the Maximum Transfer Unit (MTU) is 1500 bytes. IPSec adds additional bytes to the packet, which decreases the payload size of the packet. If tunnel mode AH is used, the packet appears as depicted by Figure 4.9; this includes 20 bytes for the new IP header, X bytes for the AH field (AH consists of five fixed-length fields and a variable-length authentication data field), 20 bytes for the original IP header and finally 20 bytes for the TCP header. The maximum and optimal payload size TCP could transfer in tunnel mode AH is then

$$\mathrm{OPS}_{\mathrm{TCP\_AH}} = 1440 - X \text{ bytes.} \tag{4.2}$$

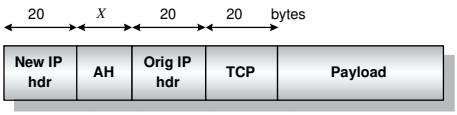


Figure 4.9: *Tunnel mode AH for IPv4.*

A similar calculation can be made for UDP with a header of 8 bytes. The optimal packet payload size for UDP is then

$$\mathrm{OPS}_{\mathrm{UDP\_AH}} = 1452 - X \text{ bytes.} \tag{4.3}$$

If tunnel mode ESP is used, the packet looks as illustrated by Figure 4.10.

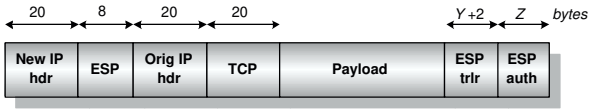


Figure 4.10: *Tunnel mode ESP for IPv4.*

The packet includes 20 bytes for the new IP header, 8 bytes for the ESP header (consisting of a Security Parameter Index (SPI) and a sequence number), and 20 bytes for the original IP header. Furthermore, the ESP trailer contains Y padding bytes (0–255 bytes), 1 byte for indicating the length of the padding and, 1 byte for the next header field. Finally, the ESP packet includes the authentication data of variable length, Z bytes. Therefore, the maximum and optimal packet payload size for TCP within tunnel mode ESP is

$$\mathrm{OPS}_{\mathrm{TCP\_ESP}} = 1470 - (Y{+}Z) \text{ bytes} \tag{4.4}$$

and for UDP the payload size amounts to

$$\mathrm{OPS}_{\mathrm{UDP\_ESP}} = 1482 - (Y{+}Z) \text{ bytes.} \tag{4.5}$$

4.4.3 Bottleneck Indication Test

IP network administrators use different tools, like *ping* or *traceroute*, to evaluate the network performance. A comprehensive overview of monitoring tools can be found in [99]. The objective of the bottleneck measurements presented by this chapter is to indicate the performance behavior of the IPSec tunnel mode between two gateways, as depicted by Figure 4.7. The bottleneck indicator test was performed as a passive measurement method, in which no probing traffic is injected into the network. Compared to active measurement methods it does not impose any additional load on the network or interfere with regular IPSec traffic.

An optimal communication link should be transparent and not introduce any loss or delay. However, if an IPSec tunnel is enabled, the throughput performance changes. Through a tunnel, packets can be delayed or dropped due to resource exhaustion in the IPSec gateways.

Because of the finite resources of the network entities, the IPSec tunnel may not be sufficient to accommodate a security attack. It is therefore necessary to provide, in analogy to [100], a definition of the term bottleneck in the presence of IPSec:

Definition: A *bottleneck*, in terms of IPSec, is a permanent lack of capacity (depending on the cryptographic function used and available computational resources) compared to the requirements of the traffic. As a result, the bottleneck introduces delay and/or packet loss to packets passing the IPSec tunnel.

Measurement Methodology

The test bed used for bottleneck detection is the same as the one used for the throughput performance test, as illustrated by Figure 4.7. The objective is to monitor incoming and outgoing traffic for the IPSec link and compare their statistics. In essence, the IPSec gateways and the IPSec link are to be treated as a single black box. For this purpose, two wiretaps are placed on the outer ends of the IPSec servers (GW1 and GW2). This is done in order to make the measurement procedure totally independent from the test bed environment.

Furthermore, the test procedure does not interfere with the measurement, nor does it influence the results. For capturing and analysis of the traffic, Ethereal [101] is used. It is important that Windows NT/XP based Ethereal is used, as it allows a change in buffer size for capturing incoming packets, which is not allowed by the Linux-based version. The possibility to change the buffer size is necessary, since at high data rates of incoming packets, Ethereal would start dropping packets itself if the capture buffer is too small. According to [102], the Windows NT/XP based Ethereal has problems regarding time stamp accuracy. However, since the throughput is determined by an averaging interval of one second, the errors are negligible.

A traffic generator is established at the client which generates a UDP flood with the desired throughput. Initially, a flood is sent at a rate of 10 Mbps. During the measurements, the traffic is varied from 10 Mbps to 100 Mbps with a step size of 10 Mbps. The time duration is 60 seconds for each observation interval and the link capacity for the IPSec tunnel is 100 Mbps. The traffic captured at the measurement points is extracted into a separate file for further analysis. For each received packet, the throughput is measured based on the captured traffic and the corresponding time stamps (as generated by Ethereal). For each second, a throughput value R_s is determined by counting the number of received bits and dividing the result by the duration of the interval (one second). $\bar{R}_{in}$ (incoming average throughput) and $\bar{R}_{out}$ (outgoing average throughput) are determined by calculating the mean value of the $N = 60$ individual throughput values spanning one second each. This can be summarized with Equation 4.6, in which $\bar{R}$ is defined as the average throughput:

$$\bar{R} = \frac{\sum_{s=1}^{N} R_s}{N} \tag{4.6}$$

The corresponding standard deviation δ_{in} and δ_{out} is then calculated as

$$\delta = \sqrt{\frac{1}{N-1}\sum_{s=1}^{N}(R_s - \bar{R})^2}. \tag{4.7}$$

Windows XP Professional Edition and Linux/Debian (Openswan) operating systems are chosen for the bottleneck tests. The same algorithms are

evaluated for both operating systems except for AES-SHA1 and AES-MD5, which can only be tested within Linux.

Results

Analyzing the reported results in Table 4.1 and Table 4.2, some observations can be pointed out. The results reveal that with a rising throughput $\bar{R}_{in}$, the cryptographic algorithm more or less has a diminishing effect, in which $\bar{R}_{out}$ becomes considerably smaller than $\bar{R}_{in}$. Such a permanent bottleneck implies packet loss in the long run. The evaluated algorithms used in IPSec are 3DES-SHA1, 3DES-MD5, DES-SHA1, DES-MD5, AES-SHA1, AES-MD5, SHA1, and MD5.

A single test without IPSec is performed with the same test bed. These results, compared with the results in which IPSec is enabled, illustrate the influence of IPSec with regards to throughput behavior. Moreover, Linux/Debian (Openswan) is in general more efficient than the Windows XP Professional operating system, as it supports higher throughput values. The difference in R_{out} between the operating systems is possible to detect in the results. When a bottleneck occurs, the R_{out} values stagnate at higher throughput values for Linux/Debian as compared to Windows XP Professional. To easily reveal the bottlenecks in Table 4.1 and Table 4.2, R_{out} is written in *italics* in the presence of a bottleneck ($R_{in} \gg R_{out}$). For instance, the results for 3DES-SHA1 (Windows XP) in Table 4.1 clearly indicate a saturated bottleneck for 50 Mbps and more since $\bar{R}_{out}$ is less than 500 kbps. Compared to $\bar{R}_{in}$ this means a breakdown in throughput. Also, the standard deviations from the throughput rise heavily (i.e., $\delta_{out} \gg \delta_{in}$) as $\bar{R}_{in}$ reaches the bottleneck behavior. Furthermore, a similar breakdown is obtained when Linux/Debian uses 3DES-SHA1, but with the difference of R_{out} being about 10.5 Mbps for $R_{in} \geq 50$ Mbps. For the rest of the cryptographic algorithms, the saturated bottleneck is found at different values of R_{in}.

From the results in Table 4.1 and Table 4.2 it is possible to observe a value of $\bar{R}_{out}$ that is a bit greater than $\bar{R}_{in}$, i.e., $R_{in} < R_{out}$ for DES-SHA1 (using Windows XP professional) at about 40 Mbps and also for DES-MD5 (using Linux) at about 40 Mbps. One of the reasons for this behavior has to do with

the buffering capability of the gateway. The gateway fills up the buffer and eventually bursts out packets, generating a throughput slightly higher than the incoming throughput during the observation interval.

According to the results in Table 4.1 and Table 4.2, 3DES-SHA1 is the most resource demanding algorithm in these tests, whereas AES-MD5 requires less resources. Depending on the algorithms and operating system used, it is necessary to consider the possibility of a protocol-based DoS attack, in which the resources are consumed in the gateway and the traffic is interrupted.

4.4.4 Visual Bottleneck Indicator

The objective of this section is to further indicate the performance of IPSec between two gateways, as depicted by Figure 4.7. However, in contrast to Section 4.4.3, a visual bottleneck indicator is used. The proposed method to detect and visualize a potential bottleneck was introduced by [103], in which the performance of a video conference was studied with the aim of generating quality feedback for streaming applications, such as video conferencing or on-line gaming.

The human visual perception system is excellent for handling features in visual displays [104]. Therefore, this ability is used in this section to improve the performance evaluation from a table-based representation to an integration of the information into plots. In the security community, visual-based methods of analysing network traffic to detect anomalous activities have produced significant results [104, 105, 106, 107, 108, 109, 110, 111].

Fluid Flow Model and Throughput Histogram Differences

The model used in the test is based on the *time-discrete fluid flow model*. The feasibility of this model to detect and identify bottleneck behavior is shown in [100]. The measurement presented here is an extension of the theoretical work to analyse the performance of IPSec, and uses the terms defined in [100].

In the tests, the synchronized capture software Ethereal generates two sets of data, one for each measurement machine at the wiretaps. As a result, the sets of data for each received packet p, are the time (T_p) when a particular packet is received and the size (L_p) of its payload.

$\bar{R}_{in}$ [Mbps]	δ_{in} [Mbps]	$\bar{R}_{out}$ [Mbps]	δ_{out} [Mbps]	$\bar{R}_{in}$ [Mbps]	δ_{in} [Mbps]	$\bar{R}_{out}$ [Mbps]	δ_{out} [Mbps]
No IPSec, Win XP				**No IPSec, Linux**			
10.25	0.0057	10.24	0.0048	10.23	0.0045	10.23	0.0045
20.44	0.0064	20.39	0.0069	20.43	0.0045	20.42	0.0063
30.05	0.0011	30.03	0.0011	30.04	0.013	30.04	0.014
41.01	0.0018	41.01	0.0057	40.81	0.017	40.81	0.064
50.65	0.0068	50.65	0.0068	50.35	0.058	50.35	0.059
61.37	0.079	61.37	0.079	61.24	0.045	61.23	0.067
70.40	0.018	70.38	0.019	70.33	0.013	70.33	0.013
81.74	0.048	81.69	0.077	81.03	0.033	80.03	0.073
90.22	0.028	90.18	0.027	90.03	0.028	90.00	0.025
98.02	0.0097	*95.69*	*0.33*	98.01	0.0096	*97.78*	*0.22*
3DES-SHA1, Win XP				**3DES-SHA1, Linux**			
10.35	0.0043	10.35	0.0050	10.23	0.0012	10.23	0.013
20.35	0.0016	20.34	0.056	20.22	0.0093	20.22	0.062
30.05	0.011	30.05	0.012	30.60	0.010	30.11	0.012
41.05	0.017	*26.38*	*12.30*	40.66	0.029	40.66	0.022
50.65	0.028	*0.50*	*1.23*	50.93	0.025	*10.75*	*1.20*
60.31	0.018	*0.49*	*1.16*	60.72	0.028	*10.67*	*0.97*
71.42	0.11	*0.47*	*1.01*	71.68	0.010	*10.51*	*3.32*
80.45	0.16	*0.49*	*1.08*	80.47	0.026	*10.23*	*1.21*
90.56	0.027	*0.48*	*1.03*	91.46	0.045	*10.82*	*4.36*
98.13	0.013	*0.48*	*1.11*	98.13	0.011	*10.33*	*1.92*
3DES-MD5, Win XP				**3DES-MD5, Linux**			
10.59	0.0035	10.59	0.065	10.10	0.0036	10.10	0.013
20.63	0.0039	20.63	0.014	20.42	0.0053	20.42	0.0052
30.43	0.016	30.43	0.077	30.15	0.0097	30.15	0.013
40.69	0.014	40.69	0.060	40.36	0.043	40.36	0.044
50.42	0.018	*0.99*	*1.44*	51.20	0.026	51.20	0.057
61.02	0.023	*0.61*	*1.28*	60.71	0.038	*51.83*	*0.22*
70.05	0.064	*0.59*	*1.29*	70.81	0.072	*51.88*	*0.22*
81.69	0.038	*0.52*	*1.13*	80.94	0.047	*51.87*	*0.30*
91.24	0.043	*0.50*	*1.14*	91.43	0.074	*51.86*	*0.29*
96.49	0.12	*0.59*	*1.28*	98.03	0.080	*50.99*	*4.51*
DES-SHA1, Win XP				**DES-SHA1, Linux**			
10.59	0.0039	10.59	0.0040	10.59	0.0039	10.59	0.042
20.63	0.0079	20.63	0.024	20.63	0.0075	20.63	0.0067
30.43	0.016	30.43	0.072	30.43	0.018	30.43	0.020
40.69	0.016	40.70	0.13	40.69	0.016	40.69	0.021
50.42	0.024	50.41	0.14	51.19	0.030	51.19	0.040
61.08	0.059	*45.13*	*18.67*	61.11	0.067	61.11	0.069
70.41	0.066	*10.50*	*1.86*	71.44	0.076	71.44	0.090
80.28	0.26	*0.85*	*1.46*	81.11	0.075	81.08	0.095
91.24	0.066	*0.90*	*1.46*	91.40	0.012	*78.90*	*0.80*
94.80	0.12	*0.94*	*1.54*	98.13	0.0093	*69.69*	*9.28*

Table 4.1: *The average incoming and outgoing throughput ($\bar{R}_{in}$ and $\bar{R}_{out}$) and the corresponding standard deviation (δ_{in} and δ_{out}) in Mbps for different algorithms and operating systems.*

$\bar{R}_{in}$ [Mbps]	δ_{in} [Mbps]	$\bar{R}_{out}$ [Mbps]	δ_{out} [Mbps]
DES-MD5, Win XP			
10.60	0.0035	10.60	0.0038
20.63	0.0055	20.63	0.011
30.43	0.016	30.43	0.080
40.71	0.018	40.70	0.19
50.42	0.022	50.41	0.31
61.08	0.060	61.07	0.42
70.91	0.040	*32.22*	*15.80*
80.48	0.058	*11.91*	*1.77*
91.29	0.993	*1.05*	*1.59*
96.50	*9.40*	*1.03*	*1.54*
AES-SHA1, Linux			
10.47	0.0047	10.47	0.0048
20.63	0.0065	20.63	0.0078
30.43	0.011	30.43	0.011
41.01	0.016	40.70	0.023
51.19	0.066	51.19	0.068
61.10	0.062	61.10	0.062
71.40	0.065	71.40	0.075
81.10	0.073	81.07	0.089
91.48	0.015	*24.18*	*5.02*
94.86	*17.03*	*22.97*	*3.05*

$\bar{R}_{in}$ [Mbps]	δ_{in} [Mbps]	$\bar{R}_{out}$ [Mbps]	δ_{out} [Mbps]
DES-MD5, Linux			
10.60	0.038	10.60	0.0042
20.63	0.024	20.63	0.024
30.43	0.018	30.43	0.018
40.69	0.016	40.71	0.019
51.18	0.027	51.19	0.030
60.33	0.047	60.33	0.064
70.42	0.063	70.42	0.24
81.01	0.073	81.04	0.10
91.44	0.12	91.31	0.22
96.50	*10.64*	*89.94*	*7.39*
AES-MD5, Linux			
10.35	0.0035	10.35	0.0032
20.44	0.0024	20.44	0.0067
30.53	0.010	30.53	0.015
40.02	0.015	40.02	0.023
50.42	0.026	50.42	0.027
61.08	0.058	61.08	0.057
70.42	0.064	70.42	0.072
81.14	0.082	81.11	0.10
91.46	0.15	91.40	0.40
96.49	*11.63*	*91.72*	*9.73*

Table 4.2: *The average incoming and outgoing throughput ($\bar{R}_{in}$ and $\bar{R}_{out}$)and the corresponding standard deviation (δ_{in} and δ_{out}) in Mbps for DES-MD5, DES-MD5, AES-SHA1, and AES-MD5.*

The fluid flow model works on throughput values, where R_s denotes the average bit rate observed during the interval $](s-1)\Delta T, s\Delta T]$. The throughput values are derived from the measurements of the packet arrival time process $\{T_p\}_{p=1}^{k}$ and payload length process $\{L_p\}_{p=1}^{k}$. Assume a time window, W, which has a *time resolution* ΔT. This gives $n = \lfloor W/\Delta T \rfloor$ throughput values.

A single sampling interval $s = \lceil T_p/\Delta T \rceil$ can include complete packets as well as parts of incomplete packets. For simplicity, we make the decision that a packet belongs to the interval in which the packet started. All the bits of payload p then belong to one sampling interval. Upon initializing, the time series $\{R_s\}_{s=1}^{n}$ with $R_s = 0 \; \forall \, s$ are calculated as follows:

$$R_s = R_s + \frac{L_p}{\Delta T} \quad \forall \, p \tag{4.8}$$

where each packet p gives a contribution to the interval s.

The time series $\{R_s^{in}\}_{s=1}^{n}$ respectively $\{R_s^{out}\}_{s=1}^{n}$ describe the packet streams that enter and leave the bottleneck, i.e., the IPSec tunnel, in terms of throughput. If the time series match each other, the network is transparent except for a constant transmission time, and the equivalent bottleneck remains empty. Based on these time series, the amounts of traffic X_s in the tunnel at the end of interval s is determined by

$$X_s = X_{s-1} + (R_s^{in} - R_s^{out})\Delta T \,, \tag{4.9}$$

in which we define $X_0 = 0$. The first observed packet serves as a starting point to synchronize the time series $\{R_s^{in}\}_{s=1}^{n}$ and $\{R_s^{out}\}_{s=1}^{n}$.

The next step is to define a representation of $\{R_s\}_{s=1}^{n}$ in the form of throughput histograms, which are denoted as $\mathcal{H}(\{R_s\}_{s=1}^{n}, \Delta R)$, in which ΔR defines the throughput resolution. Furthermore, the histogram values are given by

$$h_i = \frac{\text{number of } R_s \in](i-1)\Delta R, i\Delta R] \text{ in window } W}{n} \quad \forall i \,. \tag{4.10}$$

From the throughput histograms (at the input and output of the IPSec tunnel) the throughput histogram differences $\Delta\mathcal{H}(\{R_s^{out}\}_{s=1}^{n}, \{R_s^{in}\}_{s=1}^{n}, \Delta R)$

are calculated by subtracting the output histograms value with the input histogram values as follows:

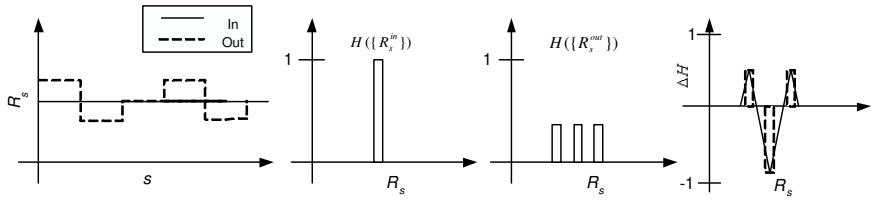
$$\Delta h_i = h_i^{out} - h_i^{in} \,. \tag{4.11}$$

This method of representing the information (by comparison of the individual throughput histograms at the input and output of the IPSec tunnel), demonstrates the performance behavior of IPSec and the presence of a potential bottleneck. The histogram difference plots are depicted by plotting Δh_i versus the upper bounds $i\Delta R$ of the corresponding throughput intervals.

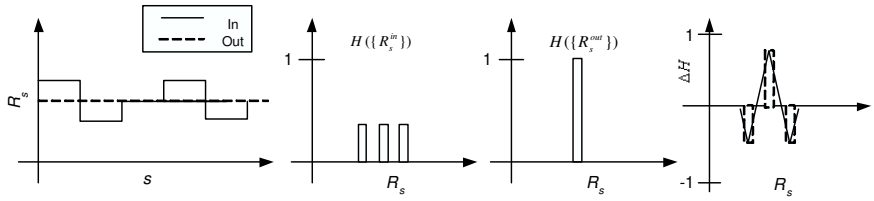
By looking at the shape of the histogram throughput difference plots, information regarding the impact of IPSec can be obtained. In Figure 4.11 a few simplified examples are illustrated which describe the general shape of the histogram plots. The shape of the plots are dependent on the behavior of the bottleneck. In Figure 4.11 (a), the incoming stream has a constant throughput and the outgoing stream displays a varying throughput. When the total demand for resources exceeds the available capacity, packets are queued or dropped, which means that the throughput of a particular stream is temporarily reduced. As soon as the demand goes below the capacity, the queued packets are released, and conceptually, the throughput of the particular stream increases. With this behavior, the resulting $\Delta\mathcal{H}$ depicts an "M" with a negative value at the incoming throughput and positive values around. This shape of $\Delta\mathcal{H}$ refers to a *shared bottleneck*.

Figure 4.11 (b) is the opposite of the shared bottleneck. It illustrates a *shaping bottleneck* on a stream. The bottleneck, in this case, reduces the throughput variations. This can be seen as the traffic bursts decrease. The resulting $\Delta\mathcal{H}$ now depicts a "W" with a higher peak at the output throughput and negative values above and below the peak.

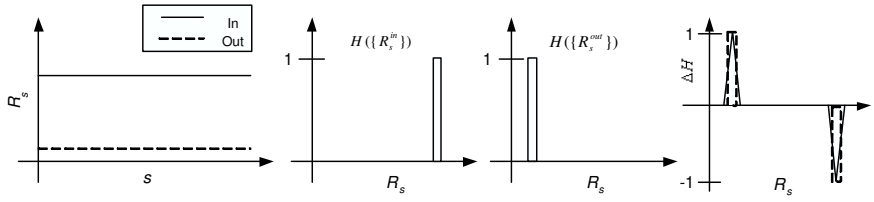
The shape of Figure 4.11 (c) describes the effect of a *saturated bottleneck*, in which the input throughput is constant and the output throughput is also constant, but at a much lower value. In this case, the available resources are not sufficient to handle the incoming stream and packets are mainly lost due to resource exhaustion. In the continuation, the saturated bottleneck behavior is obtained in the experimental results.



(a) Shared bottleneck



(b) Shaping bottleneck



(c) Saturated bottleneck

Figure 4.11: *Anticipated time plot with corresponding input and output throughput histograms and the histogram difference plot for a shared bottleneck (a), a shaping bottleneck (b), and a saturated bottleneck (c).*

In order to characterize the histogram difference plots, the following parameters are used:

- *peak-to-peak* value $= \max\{\Delta h_i\} + |\min\{\Delta h_i\}| \in [0, 2]$;
- *width* $= \Delta R\ (\max\{i|\Delta h_i \neq 0\} - \min\{i|\Delta h_i \neq 0\})$.
- $\sigma = \frac{\delta_{out} - \delta_{in}}{\delta_{in}} \quad (\sigma \geq -1)$.

In the latter formula, σ is a relative parameter that reflects the change between δ_{out} and δ_{in}. The relationship between δ_{out} and δ_{in} is as follows:

1. If $\delta_{out} = \delta_{in}$, there is no change in the standard deviation for the incoming and outgoing traffic to the IPSec tunnel.
2. If $\delta_{out} > \delta_{in}$, an increased traffic burst appears and the distribution of the throughput is wider after, rather than before the IPSec tunnel.
3. If $\delta_{out} < \delta_{in}$, a decreased traffic burst appears and the distribution of the throughput is more compact because of traffic shaping [103].

Experimental Results

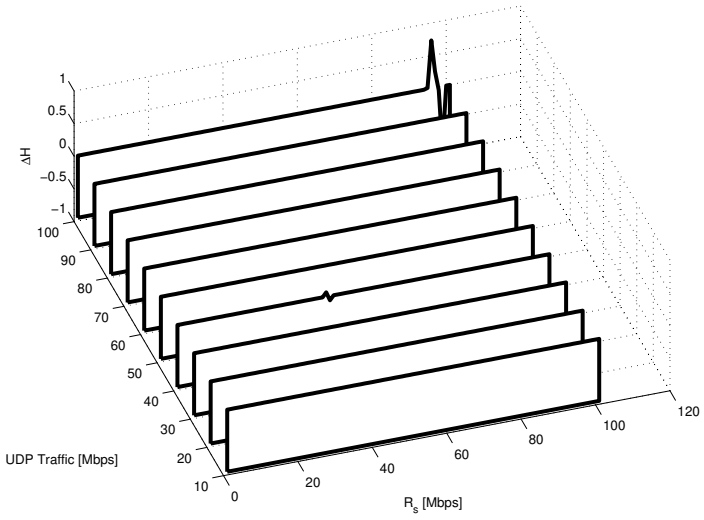
The experimental results are the outcome from the test bed as depicted by Figure 4.7. The measurement methodology is similar to that of Section 4.4.3. The time window for the histogram difference plots is $W = 60$ seconds and the time resolution $\Delta T = 1$ second, while the bandwidth resolution $\Delta R = 1$ Mbps. UDP packets are sent over the IPSec tunnel with different traffic loads of approximately 10, 20, 30, ..., and 100 Mbps. IPSec is configured with the following algorithms: 3DES-SHA1, 3DES-MD5, DES-SHA1, DES-MD5, AES-SHA1 and AES-MD5. Both Windows XP Professional and Linux/Debian operating systems are used separately in the gateways. The histogram difference plots for the separated configurations are illustrated by Figure 4.12 to 4.17. According to the results, the performance behavior decreases for several IPSec configurations. The bottleneck appears at different UDP traffic loads and depends on the cryptographic algorithm and operating system used. As indicated, the Linux based implementation of IPSec works more efficiently in terms of performance, as compared to Windows XP Professional. A detailed evaluation of the two operating systems (with regards to how IPSec is implemented) is outside the scope of this thesis.

Figure 4.12 illustrates the histogram difference plots for Windows XP Professional and Linux, without IPSec enabled and with different levels of UDP traffic. Without any cryptographic algorithms, the UDP traffic streams do not experience any change in bit rate statistics. However, for 100 Mbps traffic, a minor network influence is revealed for both Windows XP Professional

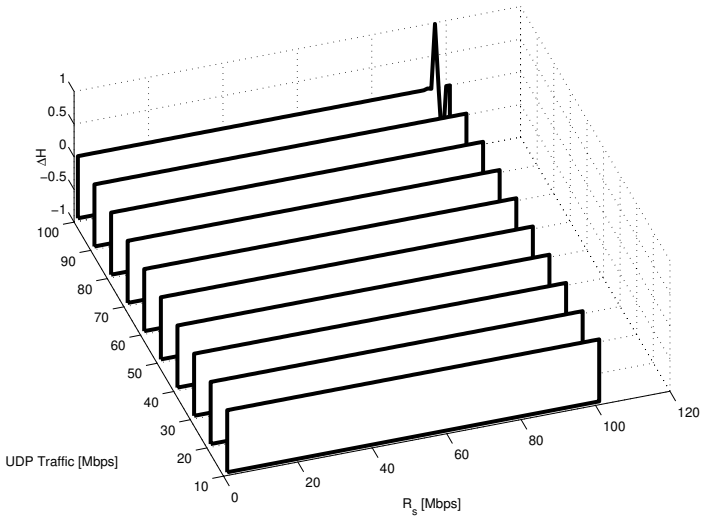
and Linux. For the 3DES-SHA1 algorithm, Figure 4.13 (a) indicates a typical shape of a saturated bottleneck which appears for the Windows XP Professional already at 30 Mbps UDP traffic. This is to be compared with the Linux operating system (Figure 4.13 (b)), in which a bottleneck emerges at 50 Mbps UDP traffic. Moreover, as seen in Figure 4.13 (a) the standard deviation at 40 Mbps UDP traffic is large for the outgoing throughput traffic. The standard deviation indicates that IPSec generates delay and packet drop with a wide spread of outgoing throughput. In the case of 3DES-MD5, a saturated bottleneck shape arises at 40 Mbps in Figure 4.14 (a) and 60 Mbps in Figure 4.14 (b). In Figure 4.15 and Figure 4.16 the DES encryption algorithm is evaluated, together with SHA1 and MD5, as authentication algorithms. Even for these algorithms a bottleneck behavior appears, but with better throughput performance than the 3DES encryption algorithm. Finally, in Figure 4.17 the AES encryption algorithm is presented with SHA1 and MD5. The results indicate that AES-SHA1 encounters a bottleneck at 90 Mbps and the combination of AES-MD5 has a minor impact on the throughput performance.

As depicted by Figure 4.12 to 4.17 the *peak-to-peak* value for a saturated bottleneck is, in several cases, close to 2, which agrees with a small standard deviation value of R_s^{out}. This is especially true for the 3DES-SHA1 and 3DES-MD5 algorithms. Appendix B contains the results of the *peak-to-peak* and the *width* parameter and also illustrates the relative parameter σ in Figure B.1 to Figure B.3. The *width* parameter is, however, dependent on which algorithm and operating system used and increases when the UDP traffic reaches higher traffic loads.

According to the user's experience, the QoS seems to decrease and a DoS attack is possible to achieve when the traffic load increases. It would be interesting to use the visual bottleneck indicator as some kind of a warning system in realtime, i.e., the throughput histogram difference plot are used, as they appear, to signal prospective performance problems in advance to the user.

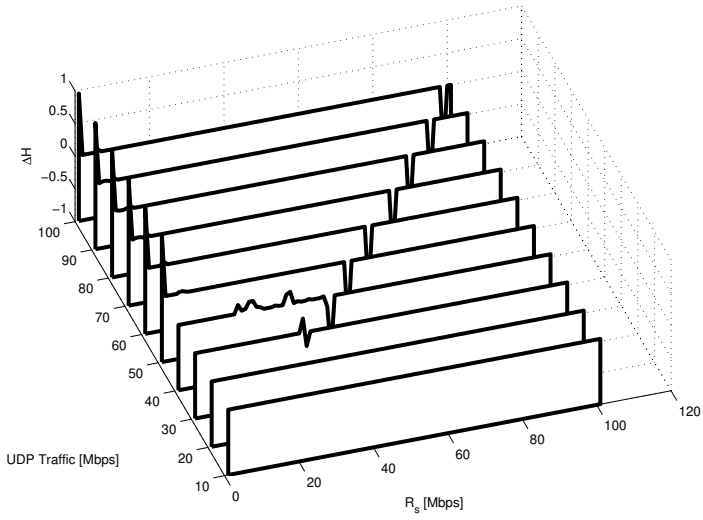


(a)

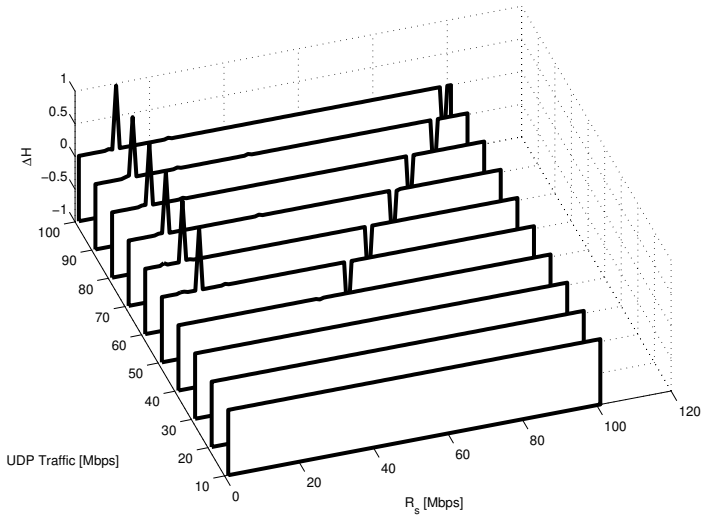


(b)

Figure 4.12: *The histogram difference plots for (a) Windows XP Professional and (b) Linux without IPSec with different levels of UDP traffic.*

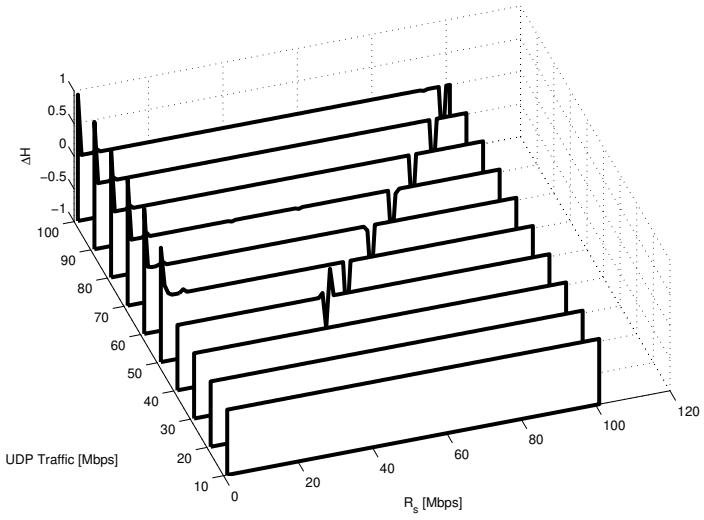


(a)

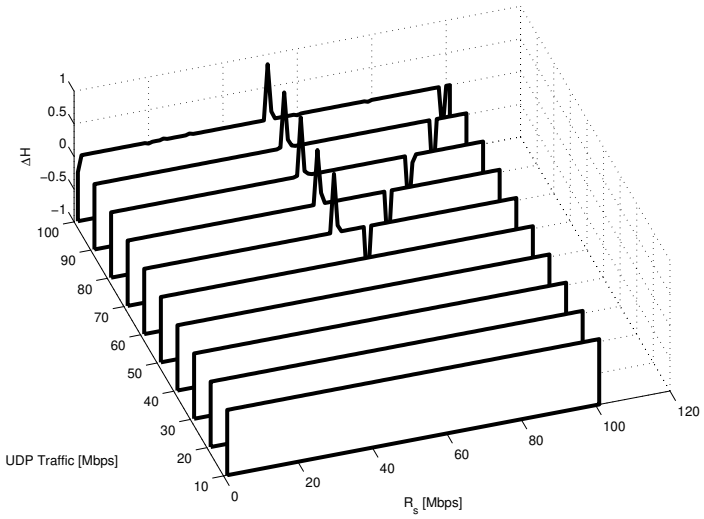


(b)

Figure 4.13: *The histogram difference plots for (a) Windows XP Professional and (b) Linux using 3DES-SHA1 with different levels of UDP traffic.*

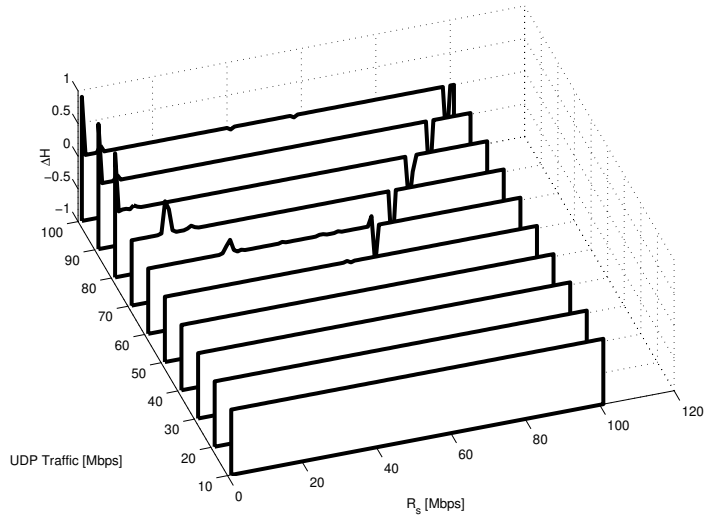


(a)

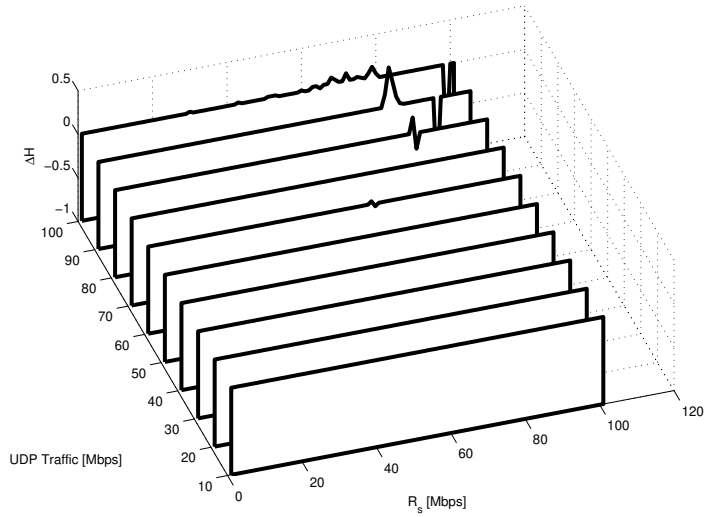


(b)

Figure 4.14: *The histogram difference plots for (a) Windows XP Professional and (b) Linux using 3DES-MD5 with different levels of UDP traffic.*

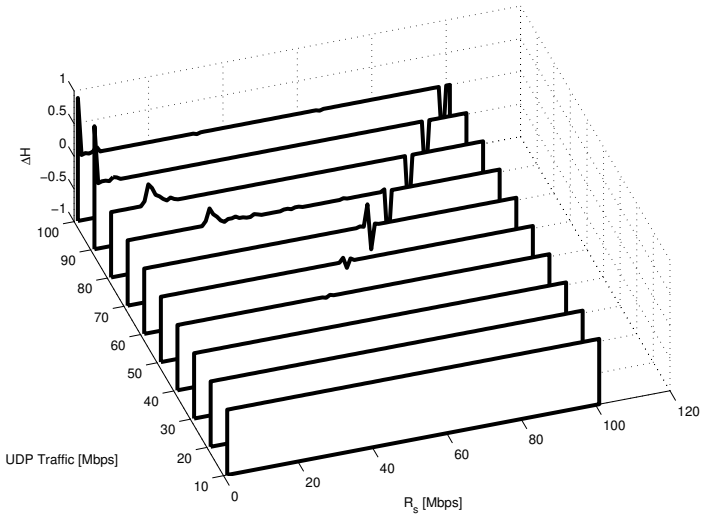


(a)

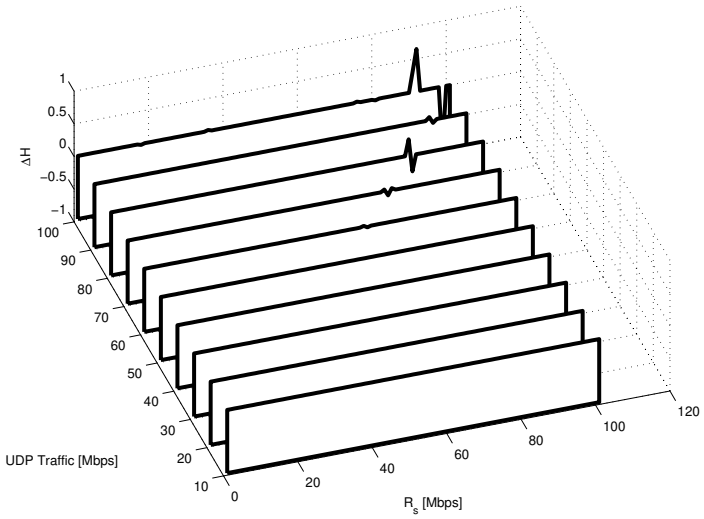


(b)

Figure 4.15: *The histogram difference plots for (a) Windows XP Professional and (b) Linux using DES-SHA1 with different levels of UDP traffic.*

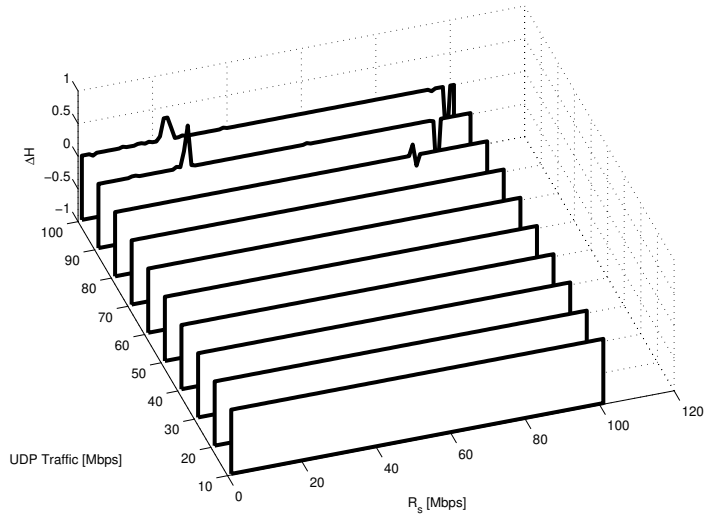


(a)

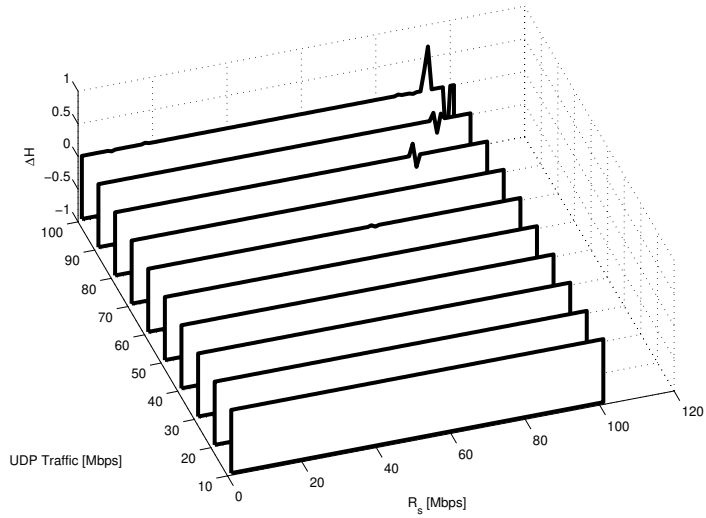


(b)

Figure 4.16: *The histogram difference plots for (a) Windows XP Professional and (b) Linux using DES-MD5 with different levels of UDP traffic.*



(a)



(b)

Figure 4.17: *The histogram difference plots for Linux using (a) AES-SHA1 and (b) AES-MD5 with different levels of UDP traffic.*

4.5 Summary

In this chapter, an IPSec performance analysis was performed, in which the effect of latency and throughput for different traffic loads were demonstrated. The tunnel mode was considered and in the tests, both Windows XP Professional and Linux/Debian were used to configure IPSec. To emphasize the bottleneck behavior of IPSec, a visual bottleneck indicator was introduced which clearly depicted the performance behavior of IPSec.

From the results, one can infer that the IPSec service mostly acted as a saturated bottleneck, which underlines the possibility of a protocol-based DoS attack against the IPSec service itself. The complexity of IPSec and the cryptographic functions used are an obvious burden. Therefore, in a constrained environment with low cost devices and rather low-end processing capability, the use of IPSec can be devastating.

Chapter 5

Lightweight Authentication Protocols

This chapter presents, in detail, three developed and novel lightweight per-packet authentication protocols that are well suited for resource-constrained environments.

5.1 Introduction

The traffic overhead of a protocol affects resource consumption by causing unnecessary energy consumption [112]. Therefore, one way to achieve resource efficiency is to use a smaller number of bits for authentication and generate these bits in a simplified way. Communication and computation resources are then used more efficiently. Moreover, resource efficient techniques do not relate only to the physical layer. Instead, it is necessary to consider the higher layers of the protocol stack and maintain resource efficiency as an important design constraint for security mechanisms.

The majority of link-layer attacks in WLANs are DoS attacks [113]. The attacks work by forging either the wireless device or the access points. Forging is possible because the IEEE 802.11 standard does not provide per-frame

source authentication. This problem can be effectively prevented if a proper authentication is added into the standard. Unfortunately, it is unlikely that commercial WLANs will support link-layer source authentication that covers both management and control frames in the near future [113]. Therefore, the fundamental question behind our research is as follows: how can adjustable and lightweight authentication protocols be designed to provide efficient network access control, with minimal impact on available resources, whilst still accurately detecting ongoing attacks? As a response, this chapter presents the development of novel and lightweight authentication protocols on a per-packet basis. The protocols are totally intellectual property free – they are open technology, meaning that any company wishing to support a lightweight authentication service (without using heavyweight authentication) may adopt them.

The objectives of the proposed authentication protocols are as follows:

- **Secure and operable:** An adversary should pass authentication with low probability and an attack should be detected by anomaly detection. Authentication mechanisms are commonly regarded as unbreakable. With this in mind, the objective is to develop authentication protocols which are lightweight but still operable.
- **Efficient:** The protocols must be efficient in terms of computational cost, memory usage, and bandwidth. This implies power efficiency and longer usage time.
- **Robust:** The protocols must be able to handle packet loss and packet reordering due to the impact of the network itself or an attack from an adversary.
- **Generic:** The protocols must be applicable to a variety of different standards.
- **Per-Packet Authentication:** The protocols should authenticate the sender continuously, i.e., for each transmitted packet. For instance, an authentication scheme which only verifies the device at the beginning of

a session encourages an adversary to perform session hijacking after the initial authentication phase has passed.

- **Mutual Authentication:** The proposed solution must be able to handle authentication in both directions. This feature would prevent address forging whilst also obstructing man-in-middle attacks.

5.2 Architectural Assumptions

The following assumptions were made when designing the protocols:

1. The sender and the receiver share a random bit generator called the *Authentication Stream Generator (ASG).*
2. Keys can be shared and distributed out-of-band. The problem of key exchange is deemed to be outside the scope of this thesis.
3. Wireless communication is not secure; the information is broadcasted and any adversary can eavesdrop, replay, or inject messages.
4. Protocols do not place any trust assumption on the communication infrastructure and, therefore, packets might not be delivered to the receiver or, alternatively may arrive out of order.

5.3 Notations

The following notations are used to describe the authentication protocols and security operations:

- A and B are principals, such as communication nodes.
- $K_{\langle AB\rangle}$ and $K_{\langle BA\rangle}$ are symmetric keys shared between A and B. No additional information is stored in this key so we have $K_{\langle AB\rangle} = K_{\langle BA\rangle}$.
- LAC$\{K_{\langle AB\rangle}, \alpha\}$ and LAC$\{K_{\langle BA\rangle}, \beta\}$ denote the Lightweight Authentication Code (LAC) generated by the ASG in A and B with corresponding keys $K_{\langle AB\rangle}$ and $K_{\langle BA\rangle}$. The index value in the authentication

stream is denoted by α for A and β for B, in which each device has a bit pointer at the current index value. If k bits are used for authentication the bit pointer will point at the first bit in each authentication block. This is demonstrated by Figure 5.1 with an example of four bits per block.

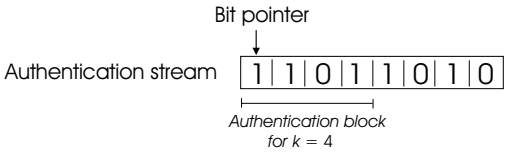


Figure 5.1: *Example of an authentication block with 4 bits per block and the bit pointer pointing at the first bit in the block.*

- {s-bit} denotes a success if the authentication bits are verified correctly. {f-bit} denotes a failure if the authentication bits are verified incorrectly. The {s-bit} and {f-bit} are piggy-backed in the reply packet that is sent back to node A from the verifier B.

5.4 Link Layer Authentication Protocols

This section aims at describing two novel and lightweight link layer authentication protocols well suited for IEEE 802.11. The first protocol is a *blind protocol*, which means that when a verification failure occurs, the sender does not know the LAC that the receiver expects. The second protocol is a *non-blind protocol*. In this case, the sender knows the LAC expected by the receiver when a verification failure takes place.

5.4.1 The Blind Protocol

The proposed blind authentication protocol is designed to obtain lightweight, per-packet authentication. This may be viewed as a new security option for the security community, determining the legitimacy of a series of continuous packets. The blind protocol was first presented in [114] and [2] as the Statistical One-bit Lightweight Authentication (SOLA) protocol, in which only

one bit per-packet was used for authentication for the IEEE 802.11 standard. The blind protocol extends the SOLA protocol to k bits for the purpose of authentication.

In the blind protocol the sender and the receiver share the same ASG with corresponding key and consequently generate the same authentication bits. Since the adversary does not have the correct key to generate the authentication bit, the probability for the adversary to correctly guess n subsequent bits is 2^{-n}, given that 0 and 1 are generated with equal probability.

The process begins with the sender generating k random authentication bits, and a pointer to the first bit of the authentication stream. These bits are added to the packet for transmission. Under normal conditions, without any packet loss or security attacks, each block of k bits is used only once. The verifier then verifies the authentication bits of the incoming packet. If the incoming authentication block of k bits matches the bits which the receiver's pointer is pointing at, a success bit {s-bit} is piggy-backed in the next reply packet. However, if the sender's bits mismatch the receiver's, a failure bit, {f-bit}, is sent back which triggers the sender's synchronization scheme. An authentication mismatch could occur for two reasons:

1. There is no synchronization between the sender's and the receiver's authentication bit pointers.

2. The sender is an adversary, who attempts to guess the authentication bits in order to gain access.

Design Description

This section describes, in detail, the blind authentication protocol and illustrates possible scenarios during packet transmission between the communicating parties A and B. Further, it outlines the synchronization algorithm, an essential part of the protocol especially in a constrained and error-prone environment.

Figure 5.2 demonstrates the operation of the blind protocol, in which the upper part of the figure describes the communication scheme from node A to node B and the lower part from node B to node A.

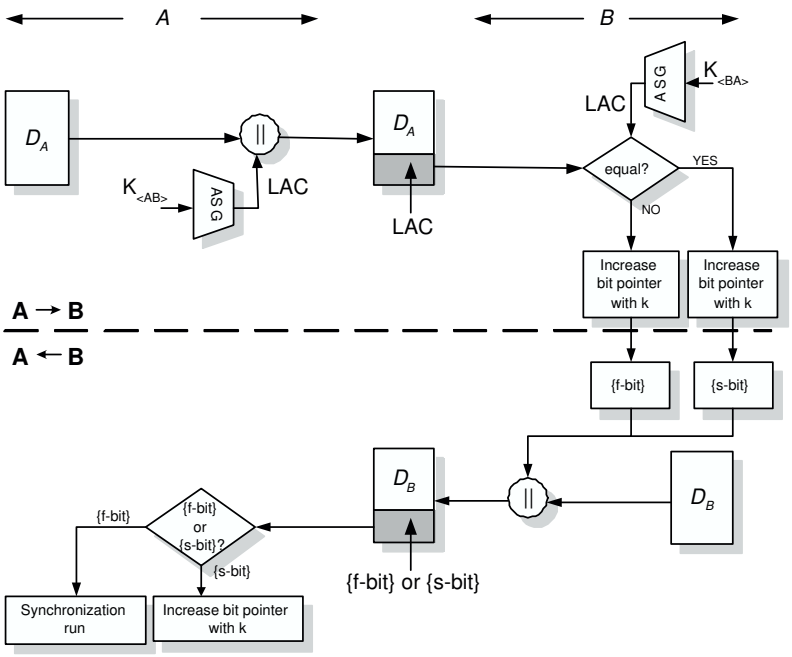


Figure 5.2: *The blind authentication protocol scheme.*

Figure 5.2 and the blind authentication protocol can further be described by five scenarios for packet transmission, in which the synchronization scheme will be triggered differently. In the following scenarios, node B verifies node A by one-way authentication. The developed protocol can, however, easily be used in both directions to obtain mutual authentication. Moreover, it is only the drop of reply packets containing the {s-bit} or the {f-bit} which affects synchronization, and not the packets containing the LAC$\{K_{\langle AB\rangle},\ \alpha\}$.

1. In the first scenario, a packet is transmitted from node A to node B and is received correctly. The authentication bits are checked as being correct. The B's bit pointer moves up k bits and a success bit, {s-bit}, is piggy-backed in the next packet sent to A. The complete messages,

with D_A and D_B as the data sent by A and B, respectively, are:

$A \rightarrow B:$ $D_A,\ \mathrm{LAC}\{K_{\langle AB\rangle},\ \alpha\}$
$B \rightarrow A:$ $D_B,\ \{\text{s-bit}\}$

2. In the second scenario, a packet is transmitted from node A to node B and is received correctly. The authentication bits are compared and found to be incorrect. B's bit pointer moves up k bits (the same as for a correct authentication bit) and a failure bit, {f-bit}, is piggy-backed in the next packet sent to A. The complete messages that A and B send are:

   $A \rightarrow B:$ $D_A,\ \mathrm{LAC}\{K_{\langle AB\rangle},\ \alpha\}$
   $B \rightarrow A:$ $D_B,\ \{\text{f-bit}\}$

3. In the third scenario, a packet is transmitted from node A to node B and the packet is dropped. Neither node A nor node B will advance their bit pointers. In this case, node A does not know if the packet from A to B or the reply packet was lost. Therefore, node A retransmits the same packet with the same authentication bits.

4. In the fourth scenario, a reply packet transmitted from node B to node A is received correctly. If it contains an {s-bit}, node A will advance the bit pointer by k bits. However, if it contains an {f-bit}, a *synchronization run* is performed and A will advance the bit pointer to the next *different authentication block*. $A's$ bit pointer is, then, further increased by k bits, in order to obtain randomness for the next set of authentication bits. The following definition applies:

   Definition: A *different authentication block* is the next authentication block with an opposite (if $k = 1$) or a different constellation of authentication bits, compared to the block which the bit pointer is currently pointing at.

   This can be formalized as follows. Given that the current authentication block is defined as

$$\vec{a}_{\alpha,k} = [a_\alpha, a_{\alpha+1}, \ldots, a_{\alpha+k-1}] \tag{5.1}$$

and the next authentication block is

$$\vec{a}_{\alpha',k} = [a_{\alpha'}, a_{\alpha'+1}, \ldots, a_{\alpha'+k-1}]\,. \tag{5.2}$$

Then if

$$\vec{a}_{\alpha,k} \neq \vec{a}_{\alpha',k}\,, \tag{5.3}$$

$\vec{a}_{\alpha',k}$ is considered to be the *different authentication block*. When the different authentication block is found, the bit pointer moves further by k bits, in which the total movement of bits equals $\Delta\alpha + k$ with $\Delta\alpha = \alpha' - \alpha$. For example, the authentication stream is 01011100... and $\alpha = 1$ and $k = 2$. The different authentication block is then found at $\alpha' = 5$. The synchronization algorithm moves the bit pointer from $\alpha = 1$ to $\alpha = 7$, as shown by Figure 5.3. If we, for the same authentication stream, have $k = 1$, the bit pointer is moved from $\alpha = 1$ to $\alpha = 3$.

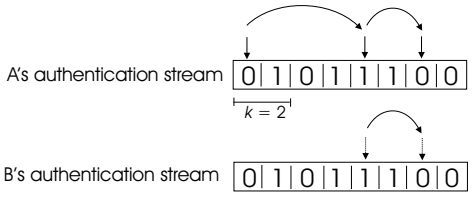


Figure 5.3: *Example of authentication stream synchronization with $k = 2$.*

5. In the fifth scenario, a reply packet, transmitted from node B to node A, is lost. Therefore, $\alpha \neq \beta$ since B's bit pointer has moved k bits, but A's bit pointer is still fixed. As for the third scenario, A will retransmit the same packet with $\mathrm{LAC}\{K_{\langle AB\rangle},\ \alpha\}$.

The synchronization algorithm explained by the different scenarios is significant for the efficiency and robustness of the protocol and can partially be

described by the pseudo code outlined in Table 5.1.

Algorithm for B
1. B receives LAC$\{K_{\langle AB\rangle},\ \alpha\}$ from A 2. If LAC$\{K_{\langle AB\rangle},\ \alpha\} ==$ LAC$\{K_{\langle BA\rangle},\ \beta\}$ then 3. B's index value = $\beta + k$ 4. $B \rightarrow A:\ \ D_B,\ \{$s-bit$\}$ 5. end 6. else if LAC$\{K_{\langle AB\rangle},\ \alpha\} \neq$ LAC$\{K_{\langle BA\rangle},\ \beta\}$ then 7. B's index value = $\beta + k$ 8. $B \rightarrow A:\ \ D_B,\ \{$f-bit$\}$ 9. end End Of Algorithm

Algorithm for A
1. A receives (D_B, {s-bit}) $\vee$ (D_B, {f-bit}) from B 2. if {s-bit} is received then 3. A's index value = $\alpha + k$ 4. end 5. else if {f-bit} is received then 6. A's index value = $\Delta\alpha + k$ 7. end End Of Algorithm

Table 5.1: *The pseudo code for the blind authentication protocol.*

Authentication Stream Generator

Technically, any random bit generator can be used as an ASG. The purpose is to generate an authentication stream which cannot be guessed by an adversary. The ASG is crucial to the security of the authentication protocol; otherwise an adversary can eavesdrop on the traffic, register the transmitted authentication bits, and guess the next k authentication bits. An adversary may also obtain the preshared key $K_{\langle AB\rangle}$ by heuristic trying.

The random authentication bits can be generated in two ways:

- by using a random bit generator and distributing the seed between A and B. An option for a good random bit generator is to implement

G-SHA [115];

- by utilizing the already implemented authentication and encryption features in the node.

A concern may arise if the authentication bit or bits are corrupted by noise, but here we depend on the Cyclic Redundancy Check (CRC) normally performed for every transmitted packet.

Analysis

Ideally, both A and B have their pointers pointing at exactly the same bit, meaning that their advancement is synchronized. However, due to packet loss, other failures or attacks, it is possible that the pointers during a session will not be synchronized. We then weaken the synchronous condition: the index value α of the A's bit pointer must be equal to or, lower than, the index value β of B's bit pointer.

Lemma 5.1: If A's bit pointer advances regularly, then B's bit pointer must have advanced.

Proof: A will not advance unless it receives the reply packet from B, piggybacked with the {f-bit} or {s-bit} and B has, at this time, already advanced.

Lemma 5.2: B and A can have $\alpha \neq \beta$, i.e., being unsynchronized, without knowing it.

Proof: Assume the following bit stream: 0001101, with two consecutive packet drops from B to A, in which $k = 1$, $\alpha = 1$ and $\beta = 3$. A and B are unaware that they are not synchronized. Both A and B realize that this is the case after two successful transmissions, moving to $\alpha = \beta = 5$, according to the synchronization algorithm.

Lemma 5.3: If A advances according to the synchronization algorithm, it will not advance further than B. This always means that $\alpha \leq \beta$.

Proof: Lemma 5.3 is proved by analysing the synchronization algorithm for all possible transmission steps with and without packet drop.

1. If the packet with LAC$\{K_{\langle AB\rangle},\ \alpha\}$ is not dropped, B increases β by k. B and A then have different index values with $\alpha < \beta$.
2. If LAC$\{K_{\langle AB\rangle},\ \alpha\}$ failed the authentication, B increases β by k, thus $\beta \geq 2 \cdot k + \alpha$. Assume that the reply packet from B to A is not dropped. A increases α by k, which gives $\alpha \leq \beta$.
3. If the packet with LAC$\{K_{\langle AB\rangle},\ \alpha\}$ is dropped, still $\alpha = \beta$. A does not increase α until the reply packet is received.
4. If the reply packet from B to A is dropped, then $\alpha < \beta$. This also holds if consecutive packets from B are dropped.
5. If LAC$\{K_{\langle AB\rangle},\ \alpha\}$ passed the authentication, B increases β by k and $\alpha < \beta$. Assume that the reply packet from B to A is not dropped, A increases α and $\alpha \leq \beta$.

Lemma 5.4: Given that B and A are not synchronized and the difference between their index value is $\epsilon = \beta - \alpha$, in which $\epsilon \geq 0$. In order to synchronize again, at least one synchronization run (and $\frac{\epsilon}{k}$ runs at the most) are needed, as demonstrated by Figure 5.4.

Proof: For each synchronization run, both B and A increase their index value. The difference between the index values of A and B is a non-negative multiple of k. In the case of not synchronized bit pointers, β is at least k greater than α. Thus at least one synchronization run makes node A increase α by $\epsilon + k$ and we obtain $\alpha = \beta$. For example, if we have the authentication bit stream 1111001... with $k = 1$, $\alpha = 2$ and $\beta = 5$, according to the synchronization algorithm, one run is needed in order to obtain synchronization at $\alpha = \beta = 6$.

In the worst case, ϵ will only decrease by one for each synchronization run. Therefore, with a difference of ϵ between the index values, $\frac{\epsilon}{k}$ runs at the most are needed. For example, the bit stream is 1010011, with $k = 1$, $\alpha = 1$ and $\beta = 4$ ($\epsilon = 3$). After one run of synchronization, $\alpha = 3$ and $\beta = 5$. After the second run of synchronization, $\alpha = 5$ and $\beta = 6$. Finally, after the third run of synchronization, $\alpha = \beta = 7$. Thus, three runs of synchronization are needed.

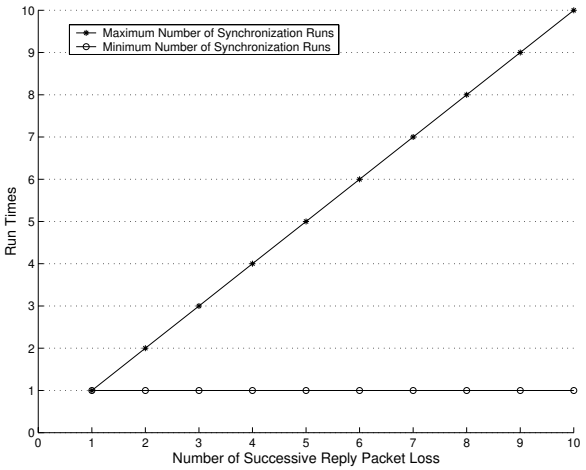


Figure 5.4: *The maximum and minimum number of synchronization runs needed for successive loss of reply packets.*

5.4.2 The Non-blind Protocol

The non-blind protocol is similar to the blind protocol except with regard to the authentication bits, which are transmitted back in the reply packet to A if an authentication failure occurs. This triggers the synchronization scheme in a different way than in case of the blind protocol. Apart from synchronization, the sender and the receiver share the same ASG with corresponding key, and generate the authentication stream in a similar way to the blind protocol. The non-blind protocol is described by the pseudo code of Table 5.2.

```
Algorithm for B

1.  B receives LAC{K_⟨AB⟩, α} from A
2.  If LAC{K_⟨AB⟩, α} == LAC{K_⟨BA⟩, β} then
3.     B's index value = β + k
4.     B → A :  D_B, {s-bit}
5.  end
6.  else if LAC{K_⟨AB⟩, α} ≠ LAC{K_⟨BA⟩, β} then
7.     B → A :  D_B, {f-bit}, LAC{K_⟨BA⟩, β}
8.     B's index value = β + k
9.  end
End Of Algorithm
```

```
Algorithm for A

1.  A receives (D_B, {s-bit}) ∨ (D_B, {f-bit}, LAC{K_⟨BA⟩, β})
from B
2.  if {s-bit} is received then
3.     A's index value = α + k
4.  end
5.  else if {f-bit} is received then
6.     A's index value = index value of the next authentication
block similar to LAC{K_⟨BA⟩, β} (with step size of k) + k
7.  end
End Of Algorithm
```

Table 5.2: *The pseudo code for the non-blind authentication protocol.*

Design Description

In order to further describe the protocol, a communication session with five scenarios (as for the blind protocol) is used to explain the packet transmission between A and B. However, the behavior for the non-blind protocol is systematically similar to that of the blind protocol for the first, the third and the fifth scenario. Therefore, the following design description applies to the second and fourth scenario only.

- In the second scenario, a packet transmitted from node A to node B is received. The authentication bits are found to be incorrect. A failure bit, {f-bit}, and the current k authentication bits LAC$\{K_{\langle BA\rangle},\ \beta\}$ are

piggy-backed in the next packet sent to A. B's bit pointer moves up k bits (the same as for a correct authentication bit scenario) to obtain randomness for the next incoming authentication bits. The complete messages which B and A send are:

$$A \rightarrow B: \quad D_A,\ \mathrm{LAC}\{K_{\langle AB\rangle},\ \alpha\}$$
$$B \rightarrow A: \quad D_B,\ \{\text{f-bit}\},\ \mathrm{LAC}\{K_{\langle BA\rangle},\ \beta\}$$

- In the fourth scenario, a reply packet transmitted from node B to node A is received correctly. If it contains an {s-bit}, node A advances the bit pointer by k bits. However, if the reply packet contains an {f-bit}, the bit pointer is advanced (by a step size of k) to the next authentication block similar to the authentication bits ($\mathrm{LAC}\{K_{\langle BA\rangle},\ \beta\}$) piggy-backed by node B in the reply packet. A's bit pointer is then further increased by k bits. For example, if the authentication stream is 11011010... and $\alpha = 1$, $k = 2$ and $\mathrm{LAC}\{K_{\langle BA\rangle},\ \beta\} = 01$, node A moves its pointer from $\alpha = 1$ to $\alpha = 5$. This is also illustrated by Figure 5.5.

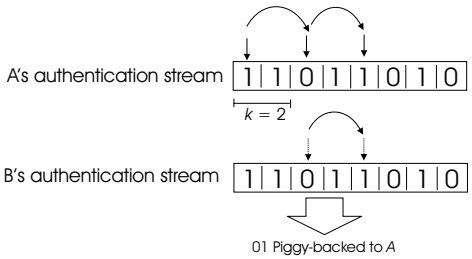


Figure 5.5: *Synchronization between authentication streams with* $k = 2$.

Another option for the non-blind protocol is to always transmit B's current authentication bits, $\mathrm{LAC}\{K_{\langle BA\rangle},\ \beta\}$, in the reply packet. In this case there is no need to add the {s-bit} or the {f-bit}. However, compared to the proposed solution, unnecessary overhead is introduced in the reply packet when the packets are verified correctly.

Analysis

The analysis for the non-blind algorithm is performed in the same way as for the blind algorithm. Lemma 5.1 still holds true, since A does not advance unless it receives the reply packet from B. Lemma 5.2 is also true, in which B and A can have $\alpha \neq \beta$, i.e., being unsynchronized without knowing it. For Lemma 5.3 we make the same claim with a similar proof as for the blind protocol. However, if an {f-bit} is received, node A increases its bit pointer to the next k bits similar to the authentication bits LAC$\{K_{\langle BA\rangle},\ \beta\}$ which node B piggy-backed in the reply packet and $\alpha \leq \beta$.

For Lemma 5.4 the non-blind algorithm will have at least one run of synchronization to obtain synchronized bit pointers. The maximum number of runs for $k = 1$ is ϵ, as shown by Figure 5.4. For $k > 1$ the maximum number of synchronization run times is dependent on k.

5.4.3 Application to Bluetooth

Chapter 3 proposed an authentication protocol as the solution to the problem of access control in IEEE 802.11. This section gives an example of the protocol's interaction with a wireless technology, namely Bluetooth [116]. The following description assumes a basic knowledge of the Bluetooth technology.

Interaction with Bluetooth

In a piconet, the Master will transmit to and receive data from each of the active Slaves. If there is nothing to send, the Master may either transmit a NULL packet or omit the Slave. However, if a Synchronous Connection Oriented (SCO) [116] link is in operation, the Slave communicates regularly according to the SCO repetition rate. Each Slave shares a unique link key with the Master in order to generate the random authentication stream. For each outgoing packet, k bits of the authentication stream are piggy-backed in the Bluetooth packet for authentication of the device. A Slave is forced to reply in the next slot each time it receives a packet for both the Asynchronous Connection Less (ACL) [116] link and the SCO link. This means that one always knows whether the information has been received. It is, therefore,

possible to implement the proposed authentication protocols in the Bluetooth environment. For the blind protocol, the {s-bit} or {f-bit} is piggy-backed in the reply packet according to the scheme, and for the non-blind protocol the current authentication bit or bits for the Slave are also piggy-backed.

Implementation

There are several methods of embedding the LAC$\{K_{\langle AB\rangle},\ \alpha\}$, {s-bit}, and {f-bit}. The optimum solution would be to embed the information in the access code field or the packet header field, since all packets, regardless of the link type (ACL or SCO), have the same packet structure of the specific fields. The access code is used to detect the presence of a packet and to address the packet to a specific device. The Slave detects the presence of a packet by matching the access code to their stored copy of the Master's access code. The packet header contains all the control information associated with the link and the packet. Finally, the packet contains the payload, where the actual message information is stored. The Bluetooth packet structure is illustrated by Figure 5.6.

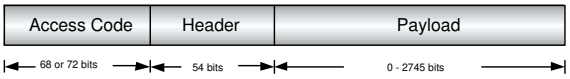


Figure 5.6: *Bluetooth packet structure.*

There are no undefined bits in the access code or header field. There are, however, several undefined bits in the ACL and the SCO payload for all different packet types. Authentication bits could, therefore, be appended to these undefined bits. For instance, the ACL payload header has four undefined bits and the DV packet, which is a combined Data-Voice packet, has a data payload with four undefined bits for multi-slot packets.

Bluetooth ASG

The previously implemented encryption stream, cipher E_0, can be used as the ASG, in which a key stream output is exclusive-OR-ed with the payload bits

and sent to the receiving device. This key stream is based on Linear Feedback Shift Registers (LFSR)[1]. Figure 5.7 depicts the proposed blind protocol for obtaining the random authentication bits, in addition to the Bluetooth encryption features. Figure 5.8 shows a similar figure for the non-blind protocol.

In order to generate the authentication stream with the Bluetooth encryption engine, there are two main operations that need to be performed:

- Authentication Key ($K_{\langle AB\rangle}$ or $K_{\langle BA\rangle}$) generation: typically involving hardware and software elements. Since key generation is not performed frequently, this is not time critical.
- Random number generation: this may be carried out using hardware or software.

According to the Bluetooth specification [116], two associated devices simultaneously derive link keys during the initialization phase (when a user enters an identical Personal Identification Number (PIN)[2] into both devices). The authentication key $K_{\langle AB\rangle}$ or $K_{\langle BA\rangle}$, as depicted in Figure 5.7 and Figure 5.8, is then generated from the current link key. The key size varies from 8 to 128 bits, and is negotiated between the Master and the Slave. The authentication bits are then generated from the authentication key ($K_{\langle AB\rangle}$ or $K_{\langle BA\rangle}$), the Bluetooth address and a slot number, together with the E_0 algorithm.

Each Bluetooth packet contains a one bit sequence number, which provides a sequential numbering scheme to order the data packet stream. For each new transmitted packet the sequence number is inverted (toggle). In the case of retransmission, this bit is not inverted. Therefore, it is not possible to use the sequence number for synchronization of the authentication bit stream.

[1] LFSR's are used in coding (error control coding) theory, cryptography and are common in stream ciphers.

[2] The PIN code can vary between 1 and 16 bytes.

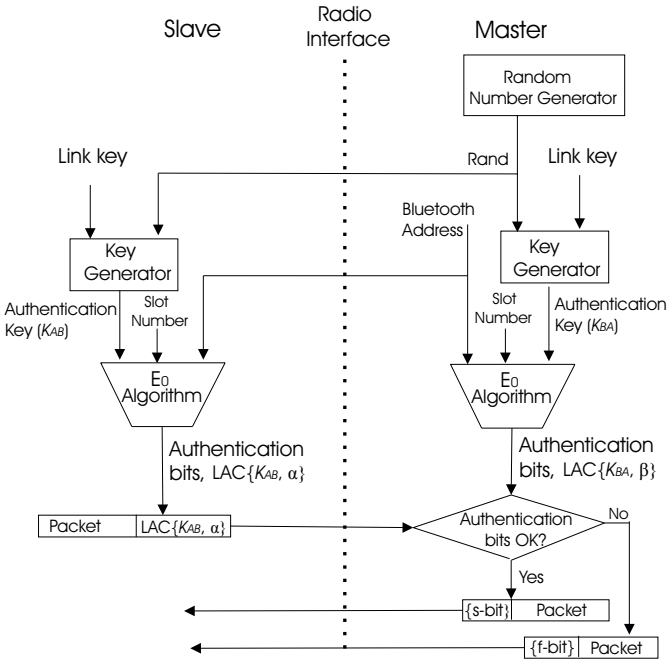


Figure 5.7: *Proposed mechanism for the blind protocol to obtain the random authentication bits.*

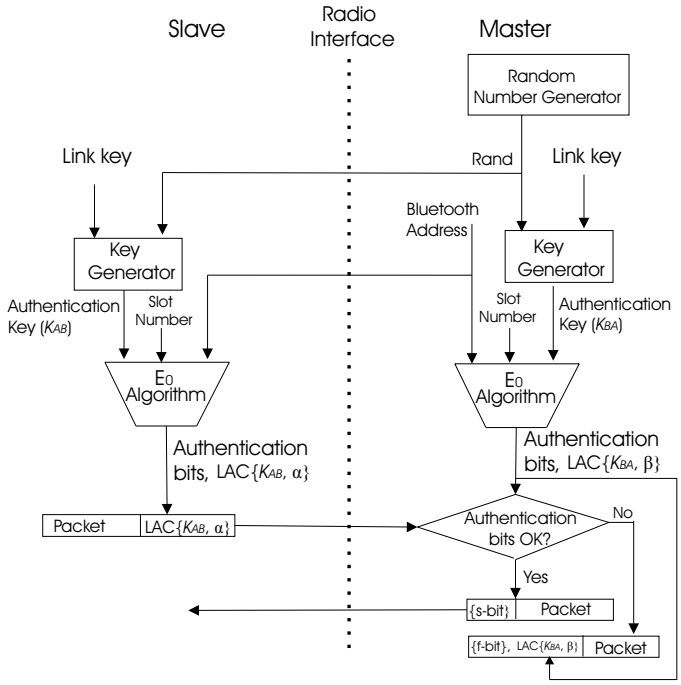


Figure 5.8: *Proposed mechanism for the non-blind protocol to obtain the random authentication bits.*

5.5 IP layer Authentication Protocol

This section describes the novel *Random-Bit Window-based Authentication (RBWA)* protocol. The RBWA protocol further develops the idea of attaching random bits to each packet for the purpose of authentication. The RBWA is deployed in the IP layer and can, therefore, work with various underlying link-layer-specific mechanisms and network topologies. Compared to IPSec, for instance, it reduces the overhead and power consumption by adding only a few authentication bits to each packet. The RBWA protocol was first published in [117] and consists of several windows schemes which handle replay attacks. A replay attack is one in which an adversary either obtains a copy of an authenticated packet and later transmits it to the intended destination, or the adversary guesses the authentication bits of upcoming packets. The receipt of duplicate packets can disrupt service, or may have other undesired consequences. The sequence number field for each packet is designed to thwart replayed packets. However, if the adversary can guess the bits in the authentication block correctly, a DoS attack can be launched by sending a forged packet with a very large sequence number to B in order to cause a number of "good" packets from a valid user to be dropped. Therefore, a robust anti-replay window scheme is required in order to resist the packet reordering which results from either network transmission or malicious intentions.

Considering that IP is a connectionless, unreliable service, the protocol does not guarantee that packets will be delivered in order, or that all packets will be delivered. For example, when a user switches from a wireless to a wired link, packet reordering is common, due to the fact that different links have different propagation delay. Routing problems may also result in packet reordering.

Based on the pre-shared and symmetric key $K_{\langle AB \rangle}$ an identical authentication stream is generated and separated into blocks. Each transmitted packet is then associated with an authentication block. An adversary has to provide the correct bits in the authentication block generated by the ASG in order to pass verification at the receiving principal. The receiver has to know which authentication block is associated with which packet. In the blind and the non-blind protocol, synchronization between the principals is achieved

through the acknowledgement mechanism. A similar implementation of acknowledgement in the IP layer would introduce communication overhead and low throughput performance when waiting for the acknowledgement. In order to achieve the synchronization between A and B in the IP layer, A attaches to each packet a unique sequence number generated by a counter. Thus, the receiver will locate the corresponding and local authentication block by using the incoming sequence number and comparing it to the received authentication bits. If they do not match, the incoming packet will be handled based on a predefined policy, in which the packet could be dropped silently, i.e., without further notification. Otherwise, it will be accepted and forwarded. In order to prevent a replay attack, the RBWA protocol is combined with anti-replay window schemes.

To implement the proposed schemes within the IP header efficiently, the existing IP header fields can be "overloaded" in a manner which does not require transformation on the side of the receiver and has minimal performance impact for end-users. The research performed by [118] demonstrates that less than 0.25% of packets are fragmented and that few Internet hosts are more than 30 hops apart. In order to represent the authentication block and sequence number, the unused bits in the 16-bit identifier field and the 3 bits field in the IP header, starting from the most significant bit in the Time to Live (TTL), can be used. The length of the authentication block and sequence number is negotiated between the principals during the initial phase of communication. If more bits are needed, other fields in the IP header can be used; however, this may require some transformation on the part of the receiver in order to avoid impacting the end-to-end transmission.

The following sections define the notations used. Additionally, they present two different windows schemes which aim to prevent a replay attack from an adversary. The proposed windows schemes are:

- IPSec-like anti-replay sliding window scheme
- Receiving and range window scheme

5.5.1 Notations

Principal A maintains a counter to generate a series of sequence numbers which are separated into segments $\{S_0, S_1, \ldots, S_{m-1}\}$. Assume that the sequence number starts from zero and there are n sequence numbers within each segment, thus $S_i = \{s_{i,0} \| s_{i,1} \| \ldots \| s_{i,n-1}\}$ where $0 \leq i \leq m-1$ and the total number of sequence numbers is $m \cdot n$.

The principals will use the ASG to generate the corresponding authentication stream for each segment. The authentication stream is denoted for S_i by R_i, in which R_i is separated into n blocks with k bits each, $R_i = \{\mathrm{LAC}_{i,0} \| \mathrm{LAC}_{i,1} \| \ldots \| \mathrm{LAC}_{i,n-1}\}$. Then A will attach $\langle s_{i,j}, \mathrm{LAC}_{i,j} \rangle$ to the corresponding packet transmitted to B where $0 \leq j \leq n-1$. A simplified figure of the transmitted packet is depicted by Figure 5.9, in which $s_{i,j}$ and $\mathrm{LAC}_{i,j}$ are inserted for the purposes of synchronization and authentication. Note that each authentication block will be used only once.

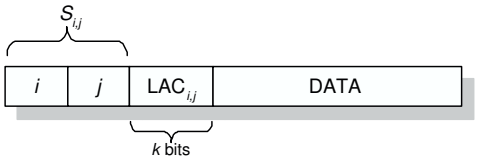


Figure 5.9: *Transmitted packet with corresponding $s_{i,j}$ and $LAC_{i,j}$.*

5.5.2 IPSec-like Anti-Replay Window Scheme

IPSec includes an anti-replay service to control duplicate packets. The objective is to reuse the IPSec mechanisms and obtain an IPSec-like anti-replay scheme, with similar properties. The IPSec-like anti replay scheme is maintained at node B. This scheme includes a window of size w. The left side of the window represents the starting sequence number (ssn). For any packet with a sequence number in the range of ssn to $ssn + w - 1$, which has been properly received and authenticated, the corresponding slot in the window will be marked. When a packet is received with sequence number $s_{i,j}$ and the authentication block $\mathrm{LAC}_{i,j}$, the following actions will be applied:

- Case 1: $s_{i,j} < ssn$
  In this case, B assumes that this packet is already received and discards the packet silently. This is further illustrated by Figure 5.10.

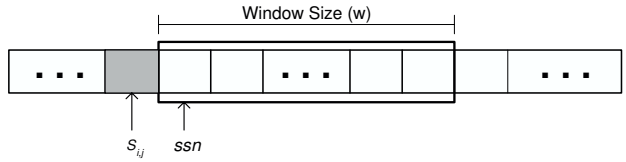


Figure 5.10: *Case 1.*

- Case 2: $ssn \leq s_{i,j} \leq ssn + w - 1$
  If the received packet falls within the window and has been received before, i.e., the slot is already marked, it will be discarded silently. Otherwise, B verifies $\mathrm{LAC}_{i,j}$ in the incoming packet by calculating its own authentication stream $R'_i = \{\mathrm{LAC}'_{i,0} \| \mathrm{LAC}'_{i,1} \| \ldots \| \mathrm{LAC}'_{i,n-1}\}$. Note that $s_{i,j} = i \cdot n + j$ where $0 \leq j \leq n-1$. If $\mathrm{LAC}_{i,j} = \mathrm{LAC'}_{i,j}$, B accepts this packet and marks the corresponding slot in the window. Note that B can cache the authentication blocks for later use. Case 2 is illustrated by Figure 5.11.

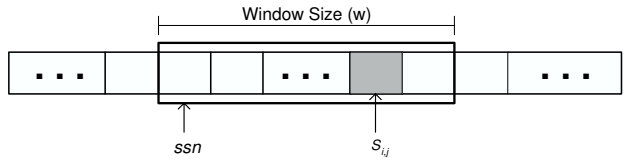


Figure 5.11: *Case 2.*

- Case 3: $ssn + w < s_{i,j}$
  B calculates R'_i using the method above. If $\mathrm{LAC}_{i,j} = \mathrm{LAC'}_{i,j}$, B determines that it has not received this packet before, it slides the window so that $s_{i,j}$ becomes the new right edge of the window and $ssn = s_{i,j} + 1$. The corresponding slot in the window is marked. If $\mathrm{LAC}_{i,j} \neq \mathrm{LAC'}_{i,j}$, B discards the packet silently. Case 3 is illustrated by Figure 5.12.

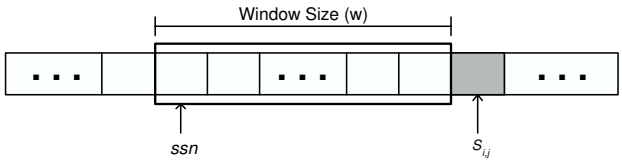


Figure 5.12: *Case 3.*

5.5.3 Receiving and Range Window Schemes

The IPSec-like anti-replay window scheme suffers from the severe packet reordering, where a large sequence number arriving earlier can force the sliding window to shift, so that a lot of "good" (but "late") sequence numbers will be dropped [119, 120]. Packet dropping and reordering deteriorate the end-to-end communication performance.

The proposed receiving window scheme can be formalized as an array of sub-windows, each of which contains information about only one, single sequence number which is either received or not-received; i.e., every sub-window contains only one received sequence number, and the sequence number in two adjacent sub-windows do not have to be consecutive. With the same number of sub-windows, as for the window size in an IPSec anti-replay window, a considerably bigger scope of sequence numbers can be represented. The receiving window can then be organized as a linked list of ascending sequence numbers. The pseudo code in Table 5.3 describes the algorithm of the *receiving window scheme*, where s denotes the incoming sequence number, $MaxW$ denotes the maximum number of sub-windows, and W denotes the current number of sub-windows and is initialized at zero to begin with.

Moreover, assume that the receiving window and the IPSec-like anti replay window have the same number of sub-windows. If sequence number x is dropped due to window shift by the receiving window, then it will be dropped by the IPSec-like anti replay window too.

The memory cost in the receiving window scheme is more than in the IPSec-like anti replay window. On the other hand, communication overhead is saved, which is desirable in a resource constrained environment.

```
Algorithm:  Receiving Window

1.  If s < ssn then
2.      discard packet
3.  end
4.  else if s == sequence number in one of the sub-windows then
5.      discard packet
6.  end
7.  else
8.      a new sub-window containing s is created
9.      W++
10.      if W > MaxW then
11.          remove the first sub-window and update ssn.
12.      end
13.  end
End Of Algorithm
```

Table 5.3: *The pseudo code for the receiving window scheme.*

Example of receiving window scheme: In Figure 5.13 we assume, to begin with, that $ssn = 4$, and $MaxW = 4$. Three packets are then received with different sequence numbers. The first packet in Figure 5.13 has $s_{i,j} = 3$ and is silently dropped since $s_{i,j} < ssn$. The second packet has $s_{i,j} = 5$ and is also silently dropped as it is a replayed packet. The last packet has $s_{i,j} = 10$ and is accepted, the sub-window with sequence number 4 is removed and ssn is updated to 5.

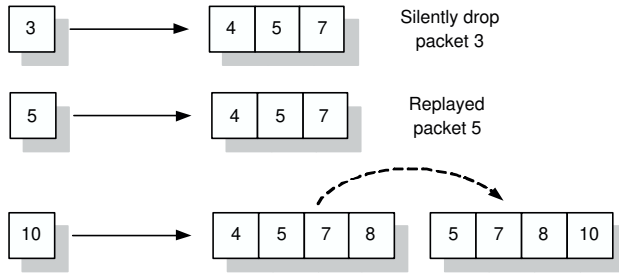


Figure 5.13: *Example of receiving window scheme*

Another window scheme, which utilizes a different approach, is the *range window scheme*. This scheme creates sub-windows that only contain consecutive sequence numbers. These sequence numbers are then aggregated into one sub-window, denoted by [*minseq, maxseq*], where *minseq* is the smallest sequence number and *maxseq* is the largest number in the aggregated sub-window. This is called the *range window scheme* and differs from the *receiving window scheme* only with regard to its "larger" sub-windows.

5.6 Summary

This chapter has presented several lightweight authentication protocols which are relevant for devices operating in a constrained environment. Authentication bits are generated from an ASG, shared between the sender and the receiver. Due to packet loss or attacks, the authentication streams may not be synchronized. Therefore, synchronization schemes have been developed to handle this problem. The first authentication protocol presented by this chapter was the *blind protocol*, in which limited information is distributed between the sender and the receiver in order to synchronize the authentication stream. The second authentication protocol was the *non-blind protocol*, with authentication bits piggy-backed in the reply packet back to the sender for faster synchronization, depending on the k parameter. Finally, the RBWA protocol (placed in the IP layer) was presented. As for the other proposed lightweight authentication protocols, RBWA is well suited for low power devices with limited resources. Instead of a synchronization algorithm, the RBWA protocol uses a sequence number to localize the correct authentication block. Within the RBWA protocol, several anti-replay window schemes were presented to efficiently inhibit replay attacks and packet reordering.

Chapter 6

Evaluation of Link Layer Authentication Protocols

In this chapter the performance of the developed link layer authentication protocols has been evaluated, and security considerations from different perspectives have been addressed. The methodology used for the evaluation is based on simulation, in which a simulator is implemented to imitate the synchronization algorithms of the blind and non-blind authentication protocols described by Chapter 5.

6.1 Simulation

Figure 6.1 schematically describes the simulator, which consists of 12 blocks. These blocks implement various aspects of the blind and non-blind authentication protocols and are described below:

Block 1 generates node A's LAC$\{K_{<AB>}, \alpha\}$ and **block 2** generates node B's LAC$\{K_{<BA>}, \beta\}$. These authentication bits will later be compared in **block 4**. In **block 3** the simulation ends after B has generated a number of N LAC$\{K_{<BA>}, \beta\}$, which is the same as receiving N number of packets

from node A. After comparing A's and B's authentication bits, node B's bit pointer is increased by k steps in **block 5** and **block 6**. Depending on whether A's and B's authentication bits are equal, an {f-bit} or {s-bit} is generated in **block 7** or **block 8**. **Block 9** decides whether or not to drop the reply packet. If the reply packet is dropped, a return to **block 2** is performed. If it is not dropped, **block 11** determines whether an {s-bit} or {f-bit} was generated. If an {s-bit} was generated, the program continues on to **block 12** and increases A's bit pointer by k steps. Otherwise, it continues on to **block 10** and executes a synchronization run according to the developed synchronization algorithms (within the proposed authentication protocols). Finally, the simulation returns to **block 1** and begins again.

6.1.1 Parameters

Performance is evaluated in terms of real and observed failure ratio, batch loss, drop probability for reply packets, equal index values, number of generated authentication bits, and the number of synchronization runs. The parameters are defined as follows:

Observed Failure Ratio: This is the ratio between the total number of failed packets (packets in which the authentication bits are faulty) and the total number of verified packets, which is collected in **block 3**.

Real Failure Ratio: This is the actual failure ratio observed in the simulations due to the number of equal index values between node A and node B. This ratio is only possible to observe in a simulation and not in a real environment. In the simulations, this parameter is calculated from the number of times α and β differed when passing **block 1** and **block 2**.

Batch Loss: This value is the successive number of dropped reply packets used to stress the synchronization algorithms. If a reply packet is dropped, the n subsequent reply packets are also dropped, where n is the batch loss parameter. In the simulations the batch loss varies between 0 and 3.

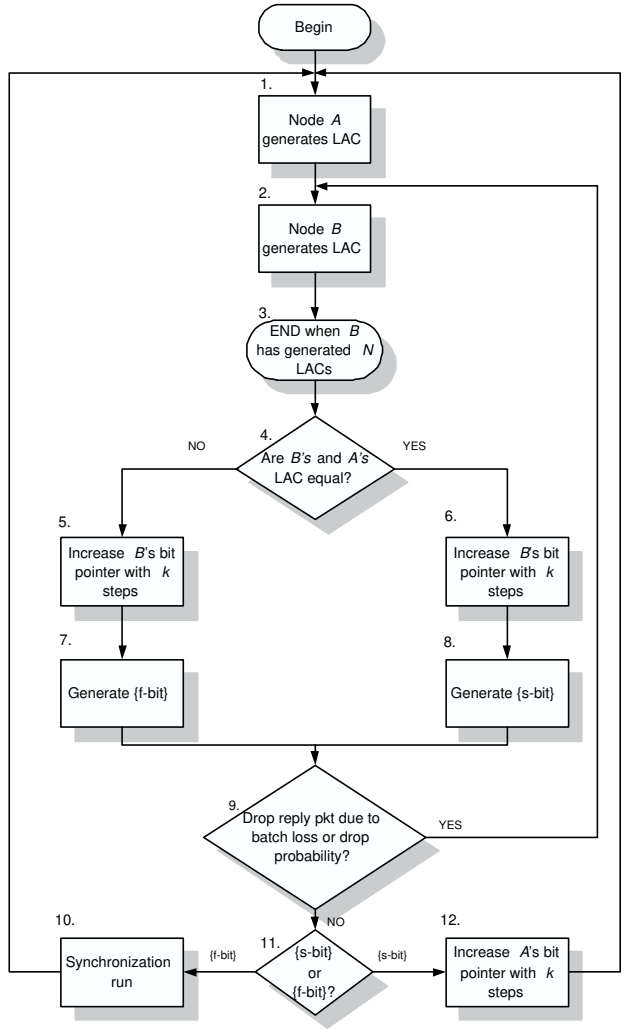


Figure 6.1: *Overview of the implemented simulator.*

The number of generated bits are measured in **block 1** for node A and in **block 2** for node B. In order to collect the number of synchronization runs, a counter is established in **block 10**.

6.1.2 Simulation Results

In this section, several simulation tests are performed for the blind and non-blind authentication protocol, in which the observed and real failure ratios are evaluated against the reply packet drop probability. The number of equal index values during a transmission between node A and node B is measured, as well as the number of synchronization runs and the number of generated authentication bits for node A. The objective is to compare the obtained results for the blind and non-blind protocol in order to determine the protocols' robustness and efficiency.

To achieve a certain statistical quality of the simulation results each simulation test outputs a mean result X_i (obtained from 100000 received packets transmitted from node A to node B). The confidence interval is then given by

$$\hat{X} \pm t_{n-1,1-\alpha/2}\frac{\delta}{\sqrt{n}} \tag{6.1}$$

where $\hat{X}$ (the value presented in Figure 6.2 to 6.11) is the average of $n = 60$ mean results X_i, δ is the standard deviation and $t_{n-1,1-\alpha/2} \simeq 2.00$ for a confidence level of $1 - \alpha = 95$ %. For all the simulation tests performed in this chapter, the half size of the confidence interval $t_{n-1,1-\alpha/2}\frac{\delta}{\sqrt{n}}$ is about 1 % of the mean.

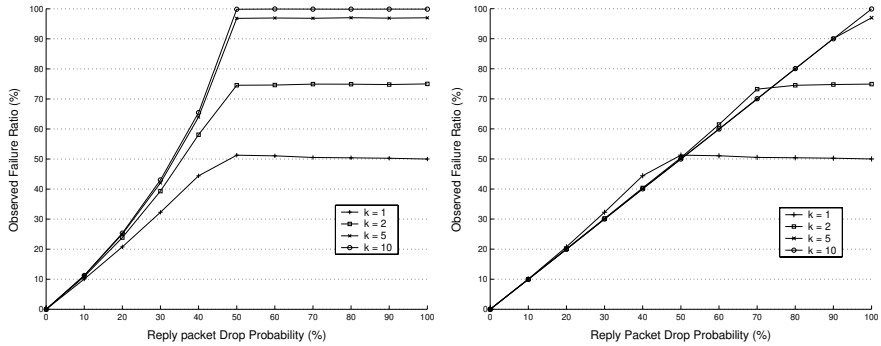
Observed Failure Ratio

The observed failure ratio has been determined for both the blind and the non-blind protocol. Figure 6.2 (a) illustrates the results for the blind protocol, in which the observed failure ratio is measured for different reply packet drop probability. In Figure 6.2 (b), corresponding results for the non-blind protocol are shown. Four different numbers of authentication bits are used,

namely $k = 1, 2, 5,$ and 10. When the reply packet drop probability is between 0 and 20 %, the observed failure ratio is almost equal for the blind and the non-blind protocol. This demonstrates that the algorithms are robust and are able to synchronize A's and B's bit pointers even when reply packets are dropped. However, the non-blind protocol behaves better than the blind protocol for higher drop probabilities. Figure 6.2 further demonstrates that the k parameter hardly affects the observed failure ratio for a low drop probability. We consider $0 - 20$ % to be normal drop probability in wireless access networks.

Given the non-blind protocol, it is natural for the protocol to better synchronize the authentication streams for $k > 1$, compared to the blind algorithm. For $k = 1$ the synchronization algorithms have the same behavior for the blind and non-blind authentication protocols.

A threshold is observed for the blind protocol at a reply packet drop probability above 50 %, in which the observed failure ratios stagnate. The stagnation level can be formalized by $1 - 2^{-k}$, as observed by Figure 6.2 (a). When the stagnation level is reached, node A and node B are always out of sync.



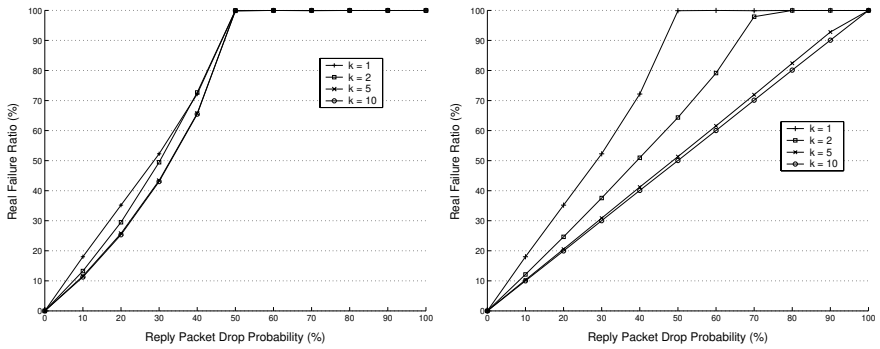
(a) (b)

Figure 6.2: *The observed failure ratio for the blind (a) and the non-blind (b) protocol vs. the drop probability of reply packets for k = 1, 2, 5, and 10.*

Real Failure Ratio

Figure 6.3 (a) depicts the real failure ratio for the blind protocol and Figure 6.3 (b) demonstrates the real failure ratio for the non-blind protocol. The simulation is performed for $k = 1$, 2, 5, and 10. It is evident that the results in Figure 6.3 (a) illustrate a lower real failure ratio for an increasing k value, since the probability of an authentication block of k bits occurring multiple times, decreases if k increases. For the non-blind protocol, the real failure ratio results are even more promising; the k parameter has a higher impact as compared to the blind protocol. This impact is because of the valuable information contained within current authentication bits which are piggy-backed in the reply packet from node B to node A, in which node A synchronizes faster.

In terms of robustness, the non-blind protocol performs better than the blind protocol. When $k = 2$ (for the non-blind protocol) the real failure ratio is lower when compared with the blind protocol. This is true for all $k > 1$. Regardless of the number of authentication bits used in each authentication block, the blind protocol will never be as robust as the non-blind for $k > 1$.



(a) (b)

Figure 6.3: *The real failure ratio for the blind (a) and the non-blind (b) protocol vs. the drop probability of reply packets for k = 1, 2, 5, and 10.*

For the blind protocol, the real failure ratio reaches 100 % when the reply packet drop probability is above or equal to 50 %. Therefore, it would be interesting to investigate the following issue: is it possible to obtain $\alpha = \beta$ after the drop probability temporarily exceeded 50 %? To test this, the following scenario is simulated: assume 10000 dropped packets, i.e., $\alpha = 0$ and $\beta = 10000$. Table 6.1 presents the results indicating the number of packets which are needed for A to synchronize with B after the 10000^{th} dropped packet, with respect to k. Despite this dropping behavior node A is capable of synchronizing with node B.

k	**packets**
1	15224
2	13059
5	11303
10	10007

Table 6.1: *The number of packets needed to synchronize again for the blind protocol after 10000 dropped packets.*

Equal Index Values

Figure 6.4 shows the results from the simulation in which the number of equal index values ($\alpha = \beta$) are observed from node A to node B for 100000 received packets. For each incoming packet to node B, the α value is known (only possible in the simulation environment) and compared with the β value for node B.

The probability for a packet to be mistakenly verified as correct (despite being incorrect) is naturally dependent on the number of authentication bits per packet. As illustrated by Figure 6.4, the number of equal index values are higher for the non-blind protocol as compared to the blind protocol, due to a more efficient synchronization algorithm.

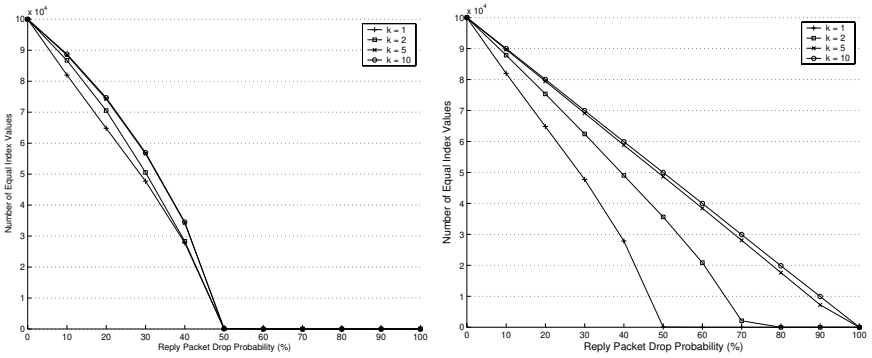


Figure 6.4: *The number of equal index values for the blind (a) and the non-blind (b) protocol vs. the drop probability of reply packets for 100000 received packets, and $k = 1, 2, 5,$ and 10.*

Difference between the Observed and Real Failure Ratio

This section presents the accuracy of the synchronization. The failure ratio difference (ΔFR) in percentage is taken between the real failure ratio (RFR) and the observed failure ratio (OFR), as

$$\Delta\mathrm{FR}_{\mathrm{blind}} = \mathrm{RFR}_{\mathrm{blind}} - \mathrm{OFR}_{\mathrm{blind}} \tag{6.2}$$

for the blind protocol and

$$\Delta\mathrm{FR}_{\text{non-blind}} = \mathrm{RFR}_{\text{non-blind}} - \mathrm{OFR}_{\text{non-blind}} \tag{6.3}$$

for the non-blind protocol. Equation 6.2 is depicted by Figure 6.5 (a) and Equation 6.3 by Figure 6.5 (b). The difference, i.e., the number of false negatives, becomes smaller if k increases and (from Figure 6.5) the non-blind protocol performs better than the blind protocol for $k > 1$.

The difference between the blind and the non-blind protocol is also measured. Figure 6.6 (a) illustrates the difference (ΔOFR) between the observed

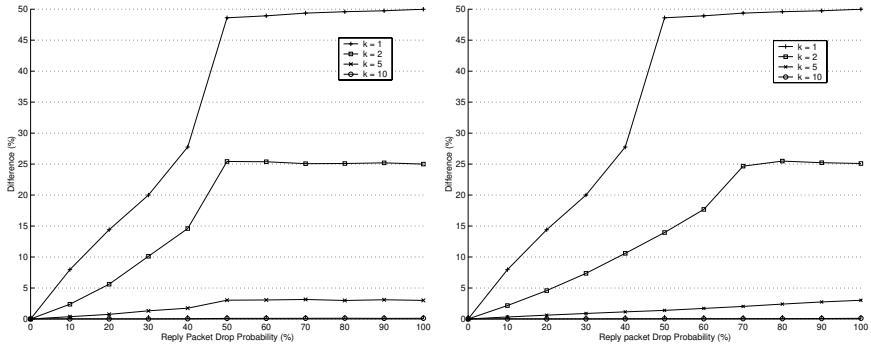
failure ratios for the two protocols as

$$\Delta\text{OFR} = \text{OFR}_{\text{blind}} - \text{OFR}_{\text{non-blind}} \tag{6.4}$$

Moreover, Figure 6.6 (b) illustrates the difference (ΔRFR) between the real failure ratios, where

$$\Delta\text{RFR} = \text{RFR}_{\text{blind}} - \text{RFR}_{\text{non-blind}}. \tag{6.5}$$

It can be seen for both the observed and the real failure ratio that (in the reply packet drop probability region of $0 - 20$ %) the difference is, at most, 5 %. For higher drop probabilities the difference is notably larger.



(a) (b)

Figure 6.5: *The difference (ΔFR) in percentage between the real and the observed failure ratio for the blind (a) and the non-blind (b) protocol vs. the drop probability of reply packets for k = 1, 2, 5, and 10.*

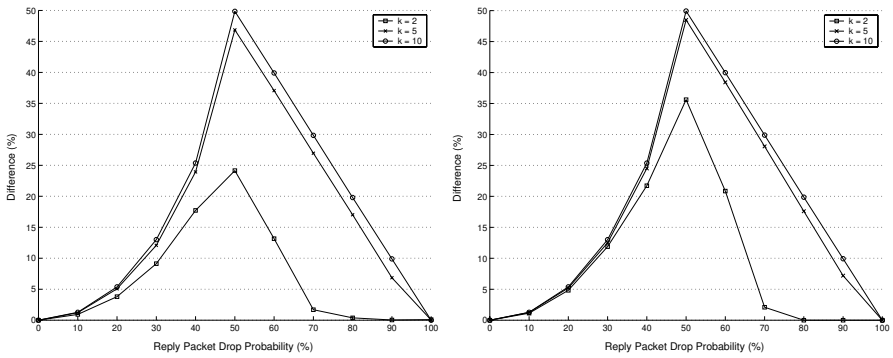


Figure 6.6: *The difference in percentage between the observed failure ratio (ΔOFR) (a) and the real failure ratio (ΔRFR) (b) for the blind and the non-blind protocol vs. the drop probability of reply packets (k = 2, 5, and 10).*

Number of Synchronization Runs

From Figure 6.7 (a) the number of synchronization runs are observed for the blind protocol. The simulation is performed for 100000 received packets from node A to node B. For the blind protocol, the number of synchronization runs increases as k increases. However, the synchronization runs eventually decrease, since the reply packet drop rate increases.

The non-blind protocol requires a smaller amount of synchronization runs compared to the blind protocol (see Figure 6.7 (b)). This relates to the algorithm performance of the non-blind protocol, since it synchronizes faster if k increases. For the blind protocol, A's bit pointer does not normally increase more than one step for each synchronization run if k is sufficiently large. In addition, the idea behind a lightweight protocol is to utilize as little processing and battery power as possible, i.e., few synchronization runs. There is, therefore, a tradeoff between the number of authentication bits used for each packet and the required authentication level for different reply packet drop probabilities. On the other hand (from Figure 6.7) the synchronization runs for different k values are similar for a drop probability of $0 - 20$ %.

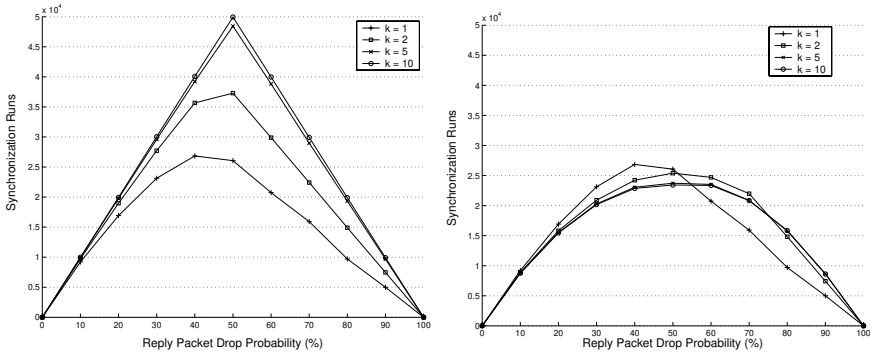
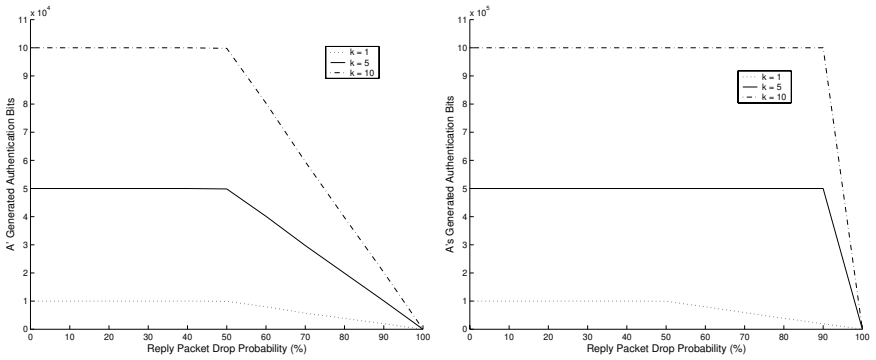


Figure 6.7: *The number of synchronization runs for the blind (a) and the non-blind (b) protocol vs. the drop probability of reply packets for 100000 received packets, and $k = 1, 2, 5,$ and 10.*

Generated Authentication Bits for A

Figure 6.8 (a) illustrates (for the blind protocol) the total number of generated authentication bits for A with 100000 received packets from A to B for $k = 1, 5$ and 10. Figure 6.8 (b) illustrates the same experiment but for the non-blind protocol. The maximum number of generated authentication bits is $100000 \cdot k$. When the reply packet drop probability increases, the number of generated authentication bits eventually decreases. This is due to the fact that A does not advance its bit pointer (in case of reply packet loss) and generates new authentication bits, as illustrated by Figure 6.8. However, another option is to generate the authentication bits in advance, keeping them in memory instead of generated separately for each outgoing packet. Furthermore, the total number of authentication bits generated for node B is constant and independent of the drop probability of reply packets. It is only dependent on k and the total number of received packets, as illustrated by Figure 6.9.



(a) (b)

Figure 6.8: *Node A's number of generated authentication bits for the blind (a) and the non-blind (b) protocol vs. the drop probability of reply packets for 100000 received packets, and $k = 1$, 5, and 10.*

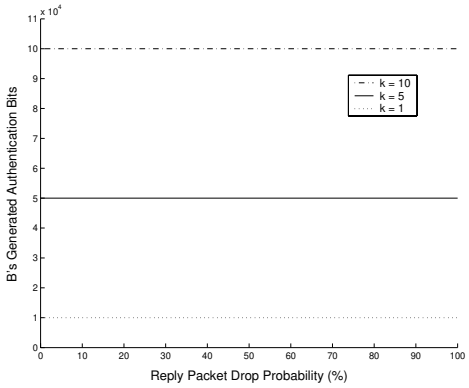


Figure 6.9: *Node B's number of generated authentication bits for both the blind and the non-blind protocol vs. the drop probability of reply packets for 100000 received packets, and $k = 1$, 5, and 10.*

Batch Loss

This section illustrates the synchronization algorithms' responses to successive reply packet drop. The successive packet drop stresses the algorithm by allowing B's bit pointer to move ahead from A's bit pointer.

In Figure 6.10, the observed failure ratio is depicted for a batch loss of 0 to 3 and with $k = 1$, i.e., the same result is obtained for both the blind and the non-blind protocol.

Figure 6.11 demonstrates the number of equal index values for different batch loss values. From the curves, the number of equal index values decreases, while the batch loss parameter increases. However, the influence of the batch loss parameter is of less importance if k increases.

The number of synchronization runs for $k = 1$ is measured with different batch loss values, as illustrated by Figure 6.12. As expected, the number of synchronization runs increases faster for a larger batch loss value.

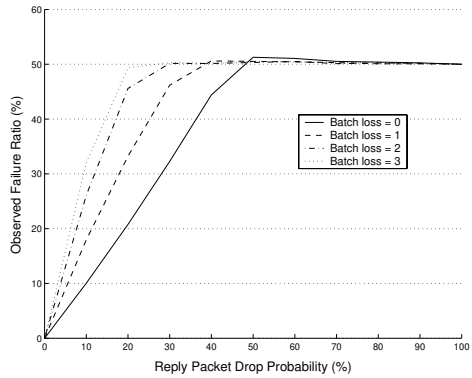
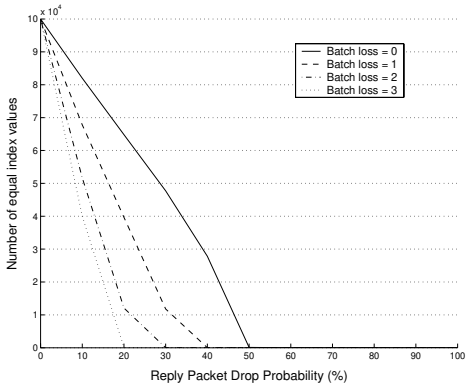
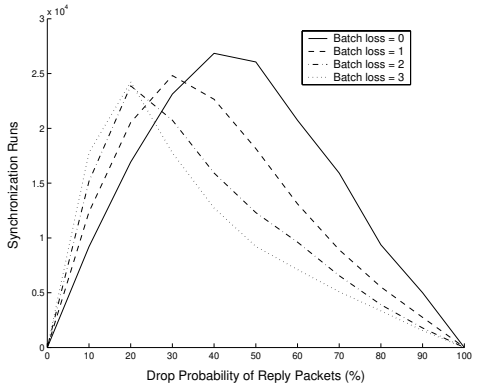


Figure 6.10: *The observed failure ratio for both the blind and the non-blind protocol vs. the drop probability of reply packets for* $k = 1$ *and batch loss = 0, 1, 2, and 3.*



(a)

Figure 6.11: *The total number of equal index values for both the blind and non-blind protocol vs. the drop probability of reply packets for 100000 received packets,* $k = 1$ *and batch loss = 0, 1, 2, and 3.*



(a)

Figure 6.12: *The number of synchronization runs for both the blind and the non-blind protocols vs. the drop probability of reply packets for 100000 received packets,* $k = 1$ *and batch loss = 0, 1, 2, and 3.*

6.1.3 Summary of Results

This section summarizes the simulation results as follows:

- Both the blind and the non-blind authentication protocols are robust and able to synchronize in the presence of normal reply packet drop.
- The non-blind protocol introduces traffic overhead while piggy-backing the expected authentication bits in the reply packet. This traffic overhead is not introduced by the blind protocol. However, the non-blind protocol utilizes the piggy-backed bits to improve the synchronization performance. The non-blind protocol performs better in terms of faster synchronization for $k > 1$, in which a smaller number of synchronization runs are required. When the k value increases, the blind protocol requires more synchronization runs when compared with the non-blind protocol.
- The difference in the number of false negatives between the two proposed protocols is small for a $0 - 20$ % reply packet drop probability.
- When the reply packet drop probability is above 50 %, for the blind protocol, the real failure ratio is 100 %, i.e., the bit pointers are out of sync. However, for the non-blind protocol the real failure ratio is then lower than 100 %, when $k > 1$.
- The capability for the blind protocol to synchronize again after the return from a drop probability above 50 % to a drop probability below 50 % was tested. The results were promising and the protocol succeeded in re-obtaining synchronization (i.e., $\alpha = \beta$). This is a desirable functionality since the two principals, A and B, do not have to renegotiate a new key for the ASG and reset the bit pointers, which is resource demanding and implies a risk factor.
- A threshold value is obtained for $k = 5$ in all the measurements, i.e., if k further increases, the impact on the real failure ratio and the number of synchronization runs is small. Considering the tradeoff between security and performance, the number of authentication bits should increase only

if a high authentication level is required, and not for the purpose of better synchronization performance.

6.2 Performance Considerations

The following performance metrics for the proposed lightweight authentication protocols are considered:

- *Computational Processing Overhead:* The proposed authentication protocols introduce minor computational processing overhead. The processing overhead for generating and verifying the LAC is small (due to the payload-independent authentication bits). The sender and the receiver can compute the authentication stream in advance with the cost of memory space, or compute the bits simultaneously with the packet processing. Depending on the implemented ASG, the processing overhead can vary. The verification of the LAC at the receiver is the most expensive operation, since the receiver first has to generate its own LAC and then start the verification process.

- *Latency:* The developed authentication protocols compute and verify the LAC by using a lightweight ASG. Therefore, the additional latency which the proposed authentication protocols incur is possible to exclude with respect to the overall end-to-end transmission latency of a packet.

- *Traffic Byte Overhead:* The traffic byte overhead is defined as the total number of non-traffic bytes a sender transmits, per time unit. The proposed authentication protocols should, if possible, be implemented in existing standards without any extra overhead. However, if traffic byte overhead is necessary, there are two sources of traffic byte overhead (except for key exchange) in the proposed authentication protocols. First, the sender adds the LAC to every packet it sends. As a result, the overhead is k bits per packet from node A to node B. Second, the verifier adds in the reply packet, depending on the authentication protocol, one ({s-bit} or {f-bit}) or k plus one bits traffic overhead. Therefore, the traffic byte overhead for the blind protocol in both directions is $\lceil \frac{k}{8} + 1 \rceil$,

and $\lceil \frac{k}{8} \rceil + \lceil \frac{k+1}{8} \rceil$ for the non-blind protocol if node A receives an {f-bit}. If an {s-bit} is received the overhead for the non-blind protocol amounts to $\lceil \frac{k}{8} + 1 \rceil$.

6.3 Security Considerations

If an attack against the network access mechanism occurs, it is necessary to have a framework with which to detect it. The major purpose of the proposed protocol is to detect and statistically identify the origin of the packets. This section analyzes different security attacks, such as: spoofing, replay attacks, DoS attacks, and eavesdropping. Further, a discussion is presented which addresses the LAC security level, the CRC, and the countermeasures.

6.3.1 Spoofing

An adversary may communicate under the identity of a valid device if they are able to obtain the authentication bits of that device through guessing or other means. In doing so, the adversary is spoofing that valid user. However, by increasing the k value, a higher security level is obtained for the purposes of authentication. The probability of guessing k bits correctly for one packet is 2^{-k} and for n packets 2^{-kn}.

For instance, as illustrated by Figure 6.2, the observed failure ratio increases if reply packets are dropped. The same result is achieved if an adversary tries to guess the random authentication bits. Anomaly detection of the failure ratio can then be used to determine an ongoing attack.

6.3.2 Replay Attacks

An adversary might be able to eavesdrop packets transmitted from a valid device and store the information in order to replay the same information (the authentication bits) later on. Normally, this is prevented by digital signatures that include time stamps and unique information from the previous transmission (such as the value of a constantly incremented sequence number). With

the proposed authentication protocols, an adversary does not benefit from deploying a replay attack, since the probability of the authentication bits being correct is the same at any time during a communication session. Moreover, the number of replayed packets can be counted and when the number reaches a certain threshold, an alert is launched.

6.3.3 DoS Attacks

Availability is an important aspect of any computer-related system; it means that assets are accessible to authorized parties at appropriate times. For this reason, there is a strong association between availability and its opposite, DoS. The concern for the proposed protocols is that a DoS attack can be 'lightweight', i.e., an adversary only needs to transmit packets with malicious information in order to disturb the synchronization algorithm. The synchronization algorithms simply offer one additional target for an existing style of attack. However, by using anomaly detection of the observed failure ratio, one is able to detect an attack with certain probability. The possibility to detect an attack, even though the security solution is lightweight, is one of the advantages of the proposed protocols.

6.3.4 Eavesdropping

An adversary could gather a significant amount of information about a victim before launching the actual active attack. For instance, eavesdropping is easily achieved in a wireless network, since signals are not carried along a wire; they are broadcasted through the air, making them more accessible to outsiders. For wired LANs, the eavesdropping attack is more difficult.

An adversary could (with regard to the proposed protocols) search the space for possible ASG keys and then predict the authentication stream. This attack could be resisted by increasing the key length, or frequently updating the key.

6.3.5 LAC Security Level

An adversary can try forging a single packet. If the proposed countermeasures are active, every failure will trigger these and log the event, perhaps erase the keys, and, if necessary, put the suspected devices on hold for a predefined period of time.

The developed authentication protocols are memoryless since $B's$ bit pointer always advances to new authentication bits for each received packet. This means that if an adversary intends to repeat an attack by guessing the bits, the probability distribution of the number of additional trials does not depend on how many failures have been observed. For each new trial the adversary guesses on k new authentication bits. Therefore, the geometric distribution [121, 122] is used to model the runs of consecutive successes (or failures), for an adversary, in repeated trials. The geometric distribution is discrete and exists only on the nonnegative integers.

An adversary can perform a series of trials to guess the LAC. Each trial can either succeed or fail, and the trials are repeated until the first success. The parameter p represents the probability of success on a single trial and $q = 1 - p$ represents the probability of failure. For an authentication level in which k bits are used, $p = 2^{-k}$. The random variable X represents the number of trials performed where $X \in \{1, 2, 3, \dots\}$. The density function for X is given by

$$f(x) = (1-p)^{x-1}p = q^{x-1}p. \tag{6.6}$$

The distribution function determined by $f(x)$ is

$$F(x) = \sum_{n \leq x} q^{n-1}p. \tag{6.7}$$

The moment-generating function method can be used to calculate the mean and the variance, where

$$G(t) = \sum_{n=1}^{\infty} e^{tn}q^{n-1}p = p\sum_{n=0}^{\infty}(e^t q)^n. \tag{6.8}$$

Equation 6.8 is simplified as

$$G(t) = \frac{p}{1 - e^t q}. \tag{6.9}$$

The first derivative of $G(t)$ is then determined by

$$G'(t) = \frac{e^t pq}{(1 - e^t q)^2} \tag{6.10}$$

and the mean is given by

$$E[X] = G'(0) = \frac{q}{p} = \frac{1-p}{p} = \frac{1 - 2^{-k}}{2^{-k}}. \tag{6.11}$$

The second derivative for $t = 0$ gives the second moment

$$E[X^2] = G''(0) = \frac{2q^2}{p^2} + \frac{q}{p} \tag{6.12}$$

and, therefore, the variance becomes

$$\sigma^2 = E[X^2] - E[X]^2 = G''(0) - G'(0)^2 = \frac{q}{p^2} = \frac{1 - 2^{-k}}{2^{-2k}}. \tag{6.13}$$

In the simplest case, we make the assumption that a LAC failure depends on a forged LAC. The probability that there are x malicious packets before the first success (false negative) is determined by $f(x)$ (Equation 6.6). Therefore, if a LAC consists of $k = 2$ authentication bits, the adversary can expect

$$E[X] = \frac{1 - 2^{-2}}{2^{-2}} = 3 \tag{6.14}$$

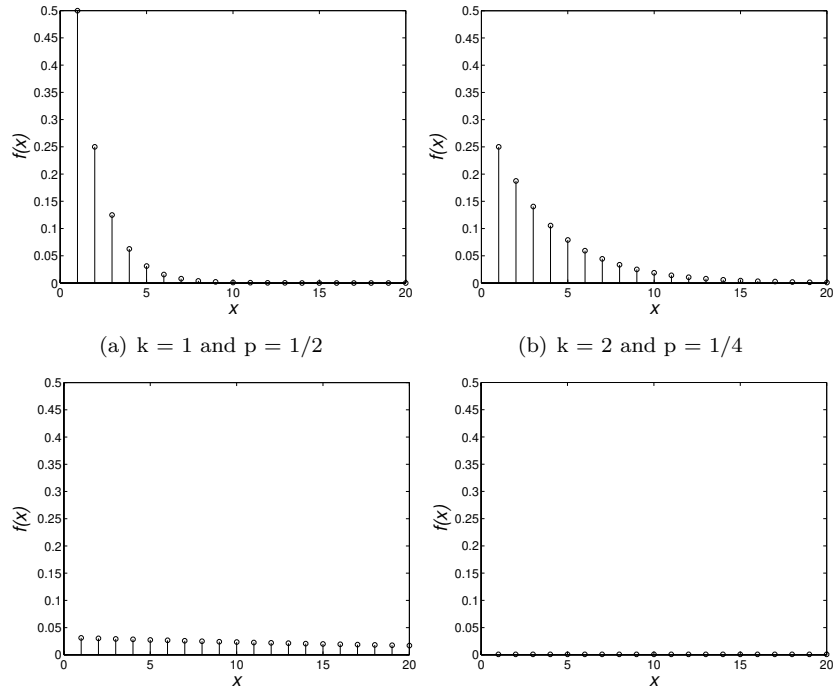
failed attempts before a successful attempt. As a countermeasure for each failed attempt, the suspected device can be put on hold for some time. For example, if the holding time for each failure is 1 minute, the expected Time To the First Forgery (TTFF) is

$$E[\text{TTFF}] = E[X] = 3 \text{ minutes}. \tag{6.15}$$

Even a small increase in the number of authentication bits gives a better performance in our favor. With a LAC of five bits the

$$E[\text{TTFF}] = 31 \text{ minutes}. \tag{6.16}$$

Figure 6.13 illustrates the graph for $f(x)$, in which $k = 1$, 2, 5, and 10 with corresponding $p = 1/2$, $1/4$, $1/32$, and $1/1024$. Notice the quick decrease for $k = 1$ and 2. By increasing k to 5 or 10, the probability for an adversary to succeed is low and does not decrease as fast as for a smaller amount of authentication bits.



(a) k = 1 and p = 1/2

(b) k = 2 and p = 1/4

(c) k = 5 and p = 1/32

(d) k = 10 and p = 1/1024

Figure 6.13: *The probability that x malicious packets are needed to get one success.*

6.3.6 Cyclic Redundancy Check

No accidental failures are expected whilst checking the LAC at the receiver due to the fact that any noise or interference on the medium will first be corrected by an assumed error-correcting code. For instance, if a packet is randomly or deliberately experiencing a bit failure and a 32 bit CRC is used, there is only a 2^{-32} chance of having a correct, i.e., false negative, CRC. Assume that there are 100 packets per second that are perturbed by noise. We can then expect one accidentally modified packet to pass the CRC every $\frac{2^{32}}{100}$ seconds, or less than once a year. A more error-prone medium channel might have 1000 CRC errors per second. If so, we expect only one packet every 50 days on average to pass the CRC code. Note that a network with this behavior most likely has to reduce its transmission speed to obtain a lower error rate.

6.3.7 Countermeasures

In the case that a device has only limited computational resources, it is possible to combine a weak LAC with countermeasures which limit the adversary's actions in other ways. An attack on a LAC involves the adversary sending fraudulent packets with the hope that at least one of them passes the LAC verification. This type of attack is possible to detect with the proposed authentication protocols. The detection rate, coupled with a low rate of accidental LAC failures, allows the receiver to take countermeasures with the objectives of retarding the adversary and ensuring that no key material is compromised.

The simplest way of detecting an active attack is to assume that an active attack is in progress any time a LAC verification failure occurs. However, this might not be the case, since a possible reply packet drop could cause successive LAC to be verified incorrectly, with no active attack launched by an adversary.

If an ongoing attack is assumed, it is possible to take further countermeasures. These countermeasures could achieve the following goals:

- Log the event as a security relevant matter, since a LAC failure could

be an indication of an active attack.

- The current authentication keys used for the ASG must be deleted and not be used again. This prevents the adversary from learning anything about the key and authentication stream generation.
- Depending on the policy, different security levels with regards to authentication can be defined. Therefore, start with lightweight authentication and boost the system to a higher authentication level if one suspects an ongoing attack. This scheme would more efficiently utilize available resources, since attacks do not usually occur. In the meanwhile, a weak access control scheme could be used, which is capable of detecting the attack.
- A realistic scenario would involve setting a threshold for the number of LAC verification failures to accept, incurring countermeasures for repeated LAC failures. The LAC failure ratio must be kept below a predefined threshold, depending on the security policy and available resources. This indicates that suspected devices are put on hold (as described in Section 6.3.5) if the failure ratio exceeds a threshold. The slowdown makes it much harder for the adversary to succeed with a large number of attempts in a short period of time. However, it is a challenge to define a threshold in terms of the failure ratio.

6.4 Summary

In this chapter, an evaluation of the proposed lightweight authentication protocols has been presented which described the robustness of the proposed authentication algorithms, in terms of reply packet loss behavior. The main variables, obtained by simulations, were: the observed failure ratio, the real failure ratio, number of equal index values, the number of synchronization run times, and the number of generated authentication bits for different reply packet drop probabilities. The results indicated that the protocols were robust and well suited to a resource constrained environment. However, the

non-blind protocol was more efficient, as compared to the blind protocol, in terms of synchronizing the bit pointers in the presence of reply packet drops.

Finally, security considerations were made for spoofing, replay attacks, DoS attacks and eavesdropping. Countermeasures were presented that actively attempted to stop or retard an adversary from prevailing in its objectives when an attack was detected. The geometric distribution was used to model the adversary's attempts to guess the authentication bits and determine the expected Time To the First Forgery (TTFF).

The RBWA protocol was not evaluated by this chapter. However, the robustness of the RBWA protocol with regards to severe packet reordering caused by an adversary or the network itself, is an important issue. In [117], an evaluation of the proposed RBWA protocol was performed based on simulation experiments. The experiments used three different packet reordering patterns and the RBWA protocol was also evaluated against the previously proposed double window protocol [119] and the controlled-shift protocol [120].

Chapter 7

Hierarchical Defense Structure for Mitigating DoS Attacks

This chapter provides the contribution of mitigating a DoS attack via a developed hierarchical defense structure with proactive functionality. An important aspect is the tradeoff between performance and security. This novel hierarchical architecture is presented with lightweight authentication protocols acting as a classifier to deny access to harmful traffic. An empirical test of the proposed structure has been performed and results are reported which display the capability of the structure to filter and separate the attack traffic before reaching the target of an IPSec gateway. Thus, the filtering of traffic is performed without being the target itself for new resource exhaustion attacks.

7.1 Introduction

Important aspects of network security include service guarantee and availability issues. Traditionally, the security community has focused on the problems associated with confidentiality and authentication but put much less effort

into those associated with availability issues. A DoS attack is a typical availability threat, in which an adversary exploits the connectivity of the network in order to disable the service offered by the target.

In general, there are two types of DoS attacks with the potential to seriously disrupt legitimate traffic at a minimal cost to the adversary: the *logic* attack and the *flooding* attack [123]. The logic attack (i.e., "Ping of Death" [124]) exploits vulnerabilities in existing software and causes a degradation of server performance or the server's eventual crash. However, this kind of attack can sometimes be prevented by updating existing software. The second type of attack, flooding attack, overwhelms the victim's Central Processing Unit (CPU), memory, network resources, or battery power by sending a large number of packets. The attack causes the victim to waste its resources by dealing with the incoming packets. Since there is no simple way to determine if an incoming packet is valid or not, it can be difficult to defend oneself against a flooding attack. In this chapter, we focus only on the instance of a flooding attack, caused by a *protocol-based DoS attack*. These attacks are serious threats to availability. Due to the fact that most of the existing protocols were not designed to be resistant to these types of attacks, the attacks are comparatively easy to implement and are often successful.

The most common access control methods normally perform strong, resource demanding authentication with traffic overhead as a cost. In addition, the DoS attack may use a strong security system's computational requirements against the service itself by utilizing the security mechanism (e.g., cryptographic encryption and authentication) as basis for an attack. In this case, an adversary floods the service with a large number of malicious packets for the target system to handle. Due to the increasing link speeds and the more computation-intensive protocols which must be supported by a security gateway, the gateways tend to become congestion points. Therefore, by applying a range of 'weak' and inexpensive crypto-functions, we may effectively prevent and detect the malicious packets entering the service with different probabilities. The computational requirements for such "weak" crypto-functions (which generate only a few secure random bits) are smaller than those of other methods, for instance, the IEEE 802.11 security framework or for the IPSec authentication method [3].

The implementation of strong and resource demanding security often implies more than a secure system; it may deteriorate the performance of a device with limited resources and pave the way for new threats such as resource exhaustion. In general, strong security is added even if there is no attack. Thus, it is unwise to use strong cryptographic algorithms for devices with limited resources, in the absence of an adversary. It is more efficient to begin with lightweight security, taking further measures when an attack is detected.

There is an increasing need for a finer-grained access control as compared to contemporary solutions without increasing the processing requirements and complexity [113]. Therefore, we propose a practical and useful access control method in the form of a hierarchical defense structure. This hierarchical defense structure provides a convenient first level barrier, based on the proposed lightweight authentication protocols in Chapter 5.

As opposed to other, reactive [125, 126, 127, 128] approaches, the proposed hierarchical approach is proactive. This means that the target is protected in advance by a filtering mechanism in order to block unapproved incoming malicious traffic. We observe that, in practice, it is not absolutely necessary to prevent every invalid packet from entering the security service. By trading off overhead with security (in the sense of access control and not confidentiality or integrity, which should be protected in an end-to-end fashion) it might be practically acceptable to have at most a predefined percentage of bad packets being allowed to access the service without causing a resource exhaustion attack. Furthermore, if an adversary tries to aggressively and maliciously gain service access, the system is able to detect such an intention and take countermeasures based on a predefined security policy.

The remainder of this chapter is organized as follows: Section 7.2 presents the attack model and Section 7.3 gives the system architecture of the defense structure with a high- and low-level design description. The performance evaluation of the hierarchical design is presented in Section 7.4 with empirical results from a prototype implementation. Finally, Section 7.5 provides a conclusion of the chapter.

7.2 Attack Model

The hypothesis for this chapter assumes that an adversary has access to a LAN or WLAN and that the target has bounded resources which can be exhausted by a clever adversary. Further, the adversary has bounded resources which may or may not be greater than or equal to those of the target. The target system of interest is an IPSec gateway for the low-end market, since the number of do-it-yourself VPNs increases [75, 76]. The target site could also contain security services and protocols such as Secure Socket Layer/Transport Layer Security (SSL/TLS) [72, 73], or Secure Shell (SSH) [74].

The objective with the system under study is to allow communication between a confirmed user and the target site. This means that a user's packet must first be authenticated and allowed access to the target site, e.g., a security gateway running an enabled security service.

The attack model described in Chapter 4 by Figure 4.1 is used to reflect a major security concern, namely, a scenario in which a flooding attack against an IPSec gateway is performed in order to exhaust the available resources.

7.3 System Architecture

With the development of new access control methods, it is possible to design a hierarchical defense structure at the edge of the public network with the task to mitigate a DoS attack. Such a structure provides the following benefits:

- Prevention of illegal activities by implementing a multilayer security system;
- Detection of an ongoing attack with a fast and mission-critical response time to counteract the attack;
- Reduction of the risk of downtime by protecting the security service.

In order for a defense system's architecture to be characterized as efficient and useful, it requires a number of desirable properties. In particular, the defense architecture should:

- not be the target itself for new attacks;
- be easy to employ and simple to operate. Therefore, any major extensions to the existing architecture should not be required;
- be able to monitor, record and report an ongoing attack;
- not generate additional traffic, thus increasing the load during attack periods;
- improve the performance of the security service and enhance network security.

The hierarchical authentication architecture falls into two categories, the *lightweight classifier function* that aggressively filters incoming packets, and the *security service*. The filtering function will take the role of a moat defense structure, in which the first line of defense (the moat) is the 'weak' authentication used to protect the 'strong' security mechanism from a DoS attack. The first line of defense is used to deny access to all but authorized users with a certain probability. Once access is gained by either a legitimate or an illegitimate packet, the second line of defense (the strong security service) finally analyzes the packets in an attempt to detect the presence of illegitimate packets.

The lightweight classifier function and the security services can be located at different positions in a WLAN or LAN environment. They can either be physically separated or they can be integrated in the same device. Figure 7.1 depicts the WLAN scenario and illustrates the scenario in which the classifier is integrated in the Access Point (AP). However, it is fully possible to integrate the classifier in the gateway or put it into a stand-alone device before the gateway. It is necessary to consider the increased cost that the extra device results in. Figure 7.2 demonstrates the high-level design for the LAN scenario, in which the classifier could either be integrated or put in front of the gateway (as for the WLAN).

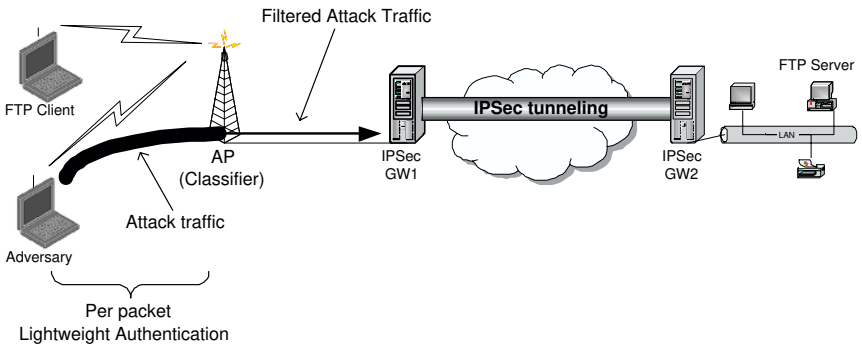


Figure 7.1: *Hierarchical defense structure for WLAN.*

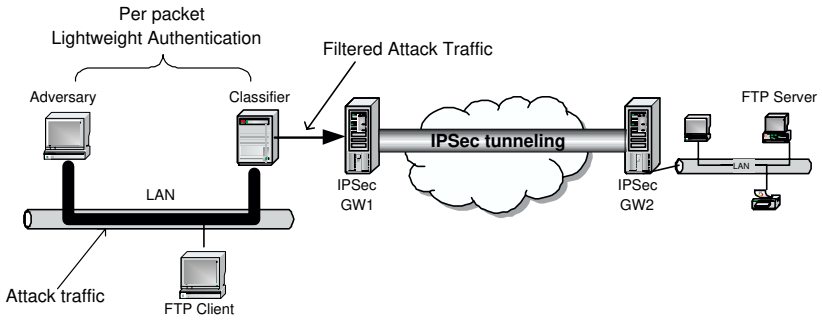


Figure 7.2: *Hierarchical defense structure for LAN.*

7.4 Performance Evaluation

This section conducts a performance evaluation on the hierarchical defense structure. In order to evaluate and validate the proposed structure, two test beds are used and configured (as illustrated by Figure 7.1 and Figure 7.2). The aim of the tests is to provide an awareness of the possibility to launch a DoS attack against a security service like IPSec over a LAN or WLAN environment. In addition, the tests evaluate the efficiency of the developed and novel defense structure with regards to mitigating an aggressive flooding attack. In the WLAN test bed, the classifier is implemented in a separate

device just in front of the IPSec gateway and not integrated in the AP.

7.4.1 Test Bed Configurations

The following section contains a description of the configurations of the test beds. The hardware and software used is described by Table 7.1 for LAN and Table 7.2 for WLAN. The network environment is based on the IPv4 protocol. The LAN environment is a 100 Mbps network while the WLAN consists of the IEEE 802.11g standard based on 3Com equipment (WLAN card and OfficeConnect gateway [129]).

Host or Gateway	**CPU**	**RAM**	**Operating System**
Adversary (Toshiba Sat. A20x)	2.4 GHz	256 MB	Linux/Debian (kernel 2.4.6)
FTP Client (Dell Latitude CPx)	750 MHz	128 MB	Windows XP
FTP Server	2 GHz	512 MB	Windows XP
Classifier	2 GHz	512 MB	Linux/Debian (kernel 2.4.6)
IPSec GW1	2 GHz	512 MB	Windows XP
IPSec GW2	1 GHz	256 MB	Windows XP

Table 7.1: *WLAN test bed configuration.*

Host or Gateway	**CPU**	**RAM**	**Operating System**
Adversary	2 GHz	512 MB	Linux/Debian (kernel 2.4.6)
FTP Client	2 GHz	512 MB	Windows XP
FTP Server	2 GHz	512 MB	Windows XP
Classifier	2 GHz	512 MB	Linux/Debian (kernel 2.4.6)
IPSec GW1	2 GHz	512 MB	Windows XP
IPSec GW2	1 GHz	256 MB	Windows XP

Table 7.2: *LAN test bed configuration.*

7.4.2 Implementation

The main design objective is modularity. Another concern is to ensure that the classifier is lightweight in terms of processing packets, such that it does not become a bottleneck itself in case of an attack. Therefore, the main task of the classifier is to classify every incoming packet as *good* or *bad*. It is the responsibility of the predefined policy to make a decision about the packet, i.e., either to drop it or accept it. The policy of dropping the packet is applied for *bad* packets and an accepting policy is applied for the *good* packets.

The software implementation consists of two different packages, a *classifier package* and a *client package*. Both of these packages are developed on top of Netfilter/libipq [130] and use the same code for the generation of the authentication bit streams.

Protocol Implementation

There are three main phases in the implementation of the authentication protocol.

- *Phase 1:* The classifier authenticates the client. The client generates a random initial authentication bit stream from a secret key. The initial authentication stream is appended in the Type of Service (TOS) field and sent to the classifier. The classifier checks these bits and compares with its own generated bits. If the initial authentication is successful, the server allocates memory resources for that client and the client and server move on to phase 2. This prevents memory exhaustion attacks.

- *Phase 2:* The client and the classifier both generate random authentication bit streams from an ASG. This stream is generated using the same secret key on both sides.

- *Phase 3:* The client and the classifier start the per-packet lightweight authentication. The lightweight authentication protocol implemented as a first line of defense is the network layer RBWA protocol presented by Chapter 5 and [117]. Other lightweight authentication protocols like the *blind* and *non-blind* protocol can, however, also be used. The client

appends the authentication bits from the ASG to the outgoing packet in the 8 bit TOS field in the IP header. The number of random authentication bits appended ($k \in \{1, \ldots, 8\}$) depends on the authentication level. As described in the design of the RBWA protocol there are other implementation alternatives of where to put the authentication bits. Every packet is appended with a sequence number that locates the authentication bits in the stream. This sequence number is incremented with every outgoing packet in the client and every incoming packet in the classifier.

These steps for client implementation are illustrated by the flow chart in Figure 7.3.

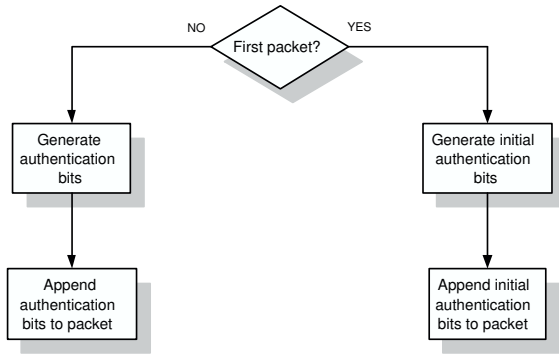


Figure 7.3: *The client's flow chart.*

Software Implementation

The classifier is implemented on the Linux 2.4.6 Kernel with IP forwarding enabled. Because the software is implemented on top of Netfilter/libipq, the packets are available in different queues (INPUT, OUTPUT or FORWARD). The classifier is placed in between the security gateway and the access network, therefore the classifier takes on the role of a routing device with a filtering capability. Therefore, packets going through the FORWARD chain can be queued for future processing. Netfilter forwards this packet to a userspace

application (the implemented classifier software). On the client's side the packets are queued for processing from the OUTPUT queue.

Once the classifier is enabled, it waits for packets from the clients. As soon as the first packet arrives in the classifier, the TOS field is checked. Since there is no ASG assigned with that client, the classifier checks the initial authentication stream. If the client is authenticated, the server allocates one ASG to it and generates a random bit stream for that client. This mapping, which occurs between the client and the ASG, is performed using the client's IP address. In essence, each ASG is assigned to an IPv4 address.

After the initial authentication phase, the client inserts the bits from the ASG into every packet. The client's counter determines the index of the bits to insert. For every incoming packet, the classifier first checks the ASG assigned to this client. It then extracts the IP identification field and determines the bits' position specified by the IP identification field. The classifier's authentication bits are compared to the bits extracted from the TOS field in the incoming packet. If they are equal, the packet is accepted and the *accept flag* is set. Otherwise, its *drop flag* is set and the packet is sent back to the kernel. The flow chart for the classifier implementation is depicted by Figure 7.4.

7.4.3 Test Methodology

The tests are based on realistic usage scenarios. A file is transferred from the FTP client to the FTP server and during the transfer a DoS attack is launched against the IPSec GW1 in order to illustrate the influence of the attack on the FTP session. These scenarios are illustrated by Figure 7.1 and Figure 7.2. The same scenario is performed for both WLAN and LAN. However, the IPSec configurations of the WLAN and LAN test beds differ. The IPSec tunnel is configured with the 3DES-SHA1 algorithms for the WLAN test bed. Due to complete resource exhaustion, it is not possible to perform the evaluation for LAN based on 3DES-SHA1. Instead, MD5 is selected.

The following tests are completed for the performance evaluation:

1. In the first test, IPSec is enabled for tunneling the FTP traffic. However, the classifier is not installed in the test bed to filter incoming packets.

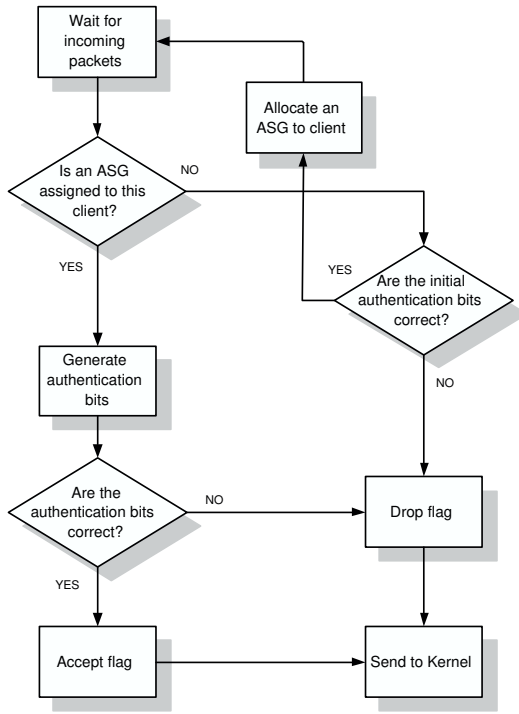


Figure 7.4: *The Classifier's flow chart.*

2. In the second test, both IPSec and the classifier for filtering incoming packets to the IPSec GW1 are enabled.

In a comparison of the first and second test (when the classifier is enabled) the impact of the hierarchical defense structure becomes evident.

The total time for each test is 45 seconds, and during this time FTP traffic is transferred. After about 15 seconds the DoS attack starts and lasts for about 15 seconds. The performance parameters of interests are:

- Throughput;
- Increments of packet sequence numbers;

- CPU utilization.

During the 45 seconds, data is captured for further analysis. The throughput is determined for FTP traffic which is received on the incoming network interface to the IPSec GW1, i.e., the filtered traffic after the classifier. The increments of sequence number is determined based on the captured traffic. In order to capture the traffic, a passive measurement point is placed before the IPSec GW1 which has no impact on the measured traffic performance. The measurement point contained Windows XP Professional, and Ethereal [101] is used to analyse the traffic. The CPU utilization is measured on the IPSec GW1 using Windows XP Professional's performance monitor. The CPU utilization is also measured on the classifier to ensure that the classifier itself did not become the target for a new DoS attack. In this regard, the proposed hierarchical defense structure is beneficial as it pre-empts this problem.

The DoS attack is initiated by flooding the IPSec GW1 with ICMP packets. There are several reasons for choosing ICMP packets: First, it is a simple method that uses the common diagnostic tool "Ping". Second, the UNIX based ping command supports the -f parameter, flood ping, in which the attack can be launched by one command (ping -f IP-address). Third, by sending an ICMP echo request to a destination address behind the IPSec GW1 an ICMP echo reply packet is returned which "hits" GW1 from the IPSec tunnel side as well. By sending one ICMP packet, one obtains a traffic multiplication by a factor of two. Such a multiplication effect is not obtained by sending a UDP packet flood.

7.4.4 Results

The results are focused on evaluating the proposed hierarchical defense structure and the performance behavior of the implemented classifier.

The throughput obtained for the LAN and WLAN is reported by Figure 7.5 and Figure 7.6, while in Figure 7.7 and Figure 7.8, the corresponding received packet sequence numbers are illustrated as functions of time. The throughput is measured based on the numbers of bits collected by Ethereal during each second.

An analysis of the results reveals a number of interesting observations. The first 15 seconds before the attack begins, the FTP experiences no difficulties in transferring the traffic. However, at about 15 seconds, the DoS attack begins with an average throughput of about 35 Mbps for LAN and 9 Mbps for WLAN and exhausts the IPSec GW1. When no classifier is installed as a first line of defense, this results in a severe downgrade of the throughput. This performance deadlock is tangible and unacceptable during the attack, yet when the attack stops the FTP traffic continues as normal. However, when the classifier is installed and the RBWA is configured for different numbers of authentication bits used for each packet ($k = 1$, 4 and 8), the throughput behavior is more promising since the classifier filters malicious packets from reaching the target, e.g., about 50 % in case of $k = 1$. Moreover, the implemented prototype classifier has an impact on the throughput during times when no attack is taking place, since each packet is investigated individually. This investigation entails a performance cost of about 10 % throughput reduction for the LAN environment. However, there is still room to implement the prototype classifier in a more efficient way in order to improve performance. With regards to the WLAN results depicted by Figure 7.6, the throughput behavior is similar to that of the LAN. The WLAN throughput oscillation occurs due to Carrier-Sense-Multiple-Access with Collision-Avoidance (CSMA/CA) behavior. In the WLAN test, the attack stops at about 30 seconds, but (as illustrated by Figure 7.6) the FTP traffic throughput starts to increase after about 33 seconds, i.e., the WLAN test bed needs about 3 seconds to recover from the aggressive flooding attack.

The increments of the packet sequence numbers in Figure 7.7 and Figure 7.8 clearly indicate the attack and further clarify the throughput performance. During the attack period, hardly any FTP packets are received at the IPSec GW1. However, when the classifier is enabled it is possible to see the increment in the packet sequence numbers and how the parameter k influences the results.

We observe that the increments of the packets' sequence number at a specific time can be estimated with

$$\hat{n}_{seq} = \frac{\bar{R}}{\bar{P}}(T_t - T_a \cdot 2^{-k}), \tag{7.1}$$

in which $\bar{R}$ is the average bit rate whilst there is no attack, $\bar{P}$ is the average packet size in bits, and T_t is the total time and also the time of which the estimated packet sequence number is predicted. T_a is the attack time period. For example, Equation 7.1 is used to predict the received packet sequence number for $k = 1$ after $T_t = 45$ seconds, an attack period $T_a = 15$ seconds, and an average bit rate $\bar{R} \simeq 35$ Mbps in the undisturbed case. The average packet size is $1470 \cdot 8$ bits (obtained from the captured traffic). According to Equation 7.1, this gives an estimated packet sequence number of 111607, which is roughly the same as illustrated by Figure 7.7.

The network capacity is the limiting factor for the network throughput before the attack. During the attack the CPU becomes the limiting factor as soon as the flooding traffic reaches the IPSec GW1. The CPU utilization for GW1 is depicted by Figure 7.9 for LAN and by Figure 7.10 for WLAN. The CPU utilization before and after the attack is acceptable for both LAN (MD5) and WLAN (3DES-SHA1). However, during the attack, the CPU reaches 100 % when the classifier is disabled, which is devastating for the throughput performance. When the classifier is enabled the CPU utilization is kept on an acceptable level, and no resource exhaustion problem appears.

With regard to the hierarchical defense scheme, it is of importance not to move the DoS property to the classifier itself. Therefore, during the attack the CPU utilization of the classifier device for the LAN environment is measured and presented by Figure 7.11. As illustrated, the CPU utilization is on a tolerable level and an increase in the number of authentication bits used has only a minor effect on the CPU utilization.

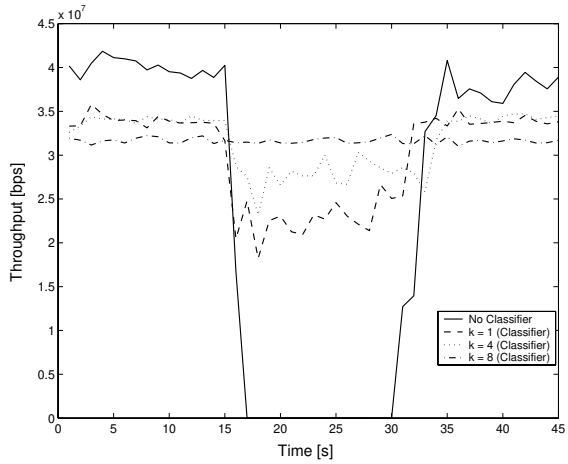


Figure 7.5: *Throughput performance for the FTP traffic over LAN with a launched DoS attack between 15 to 30 seconds.*

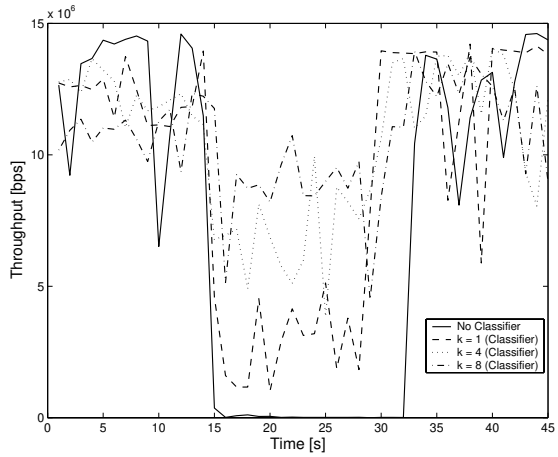


Figure 7.6: *Throughput performance for the FTP traffic over WLAN with a launched DoS attack between 15 to 30 seconds.*

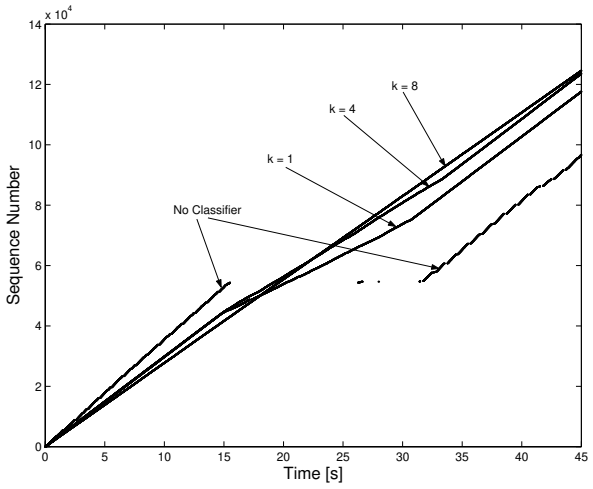


Figure 7.7: *The increment of the packet sequence numbers for the LAN environment with a launched DoS attack between 15 to 30 seconds.*

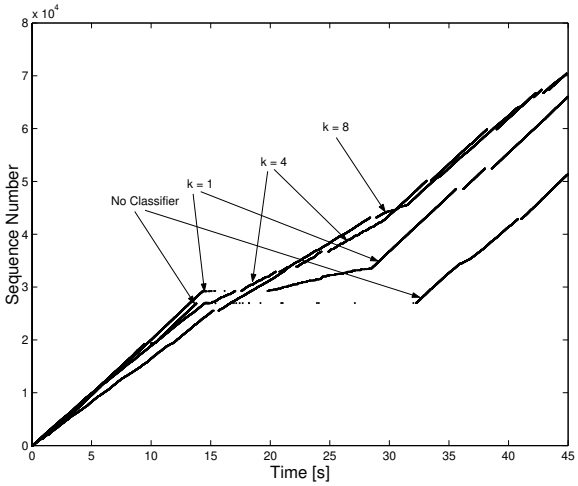


Figure 7.8: *The increment of the packet sequence numbers for the WLAN environment with a launched DoS attack between 15 to 30 seconds.*

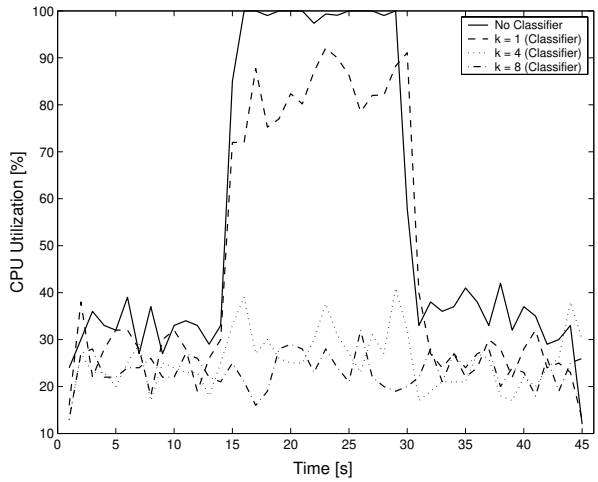


Figure 7.9: *CPU utilization of the IPSec GW1 in the LAN environment with MD5 enabled and a DoS attack between 15 to 30 seconds.*

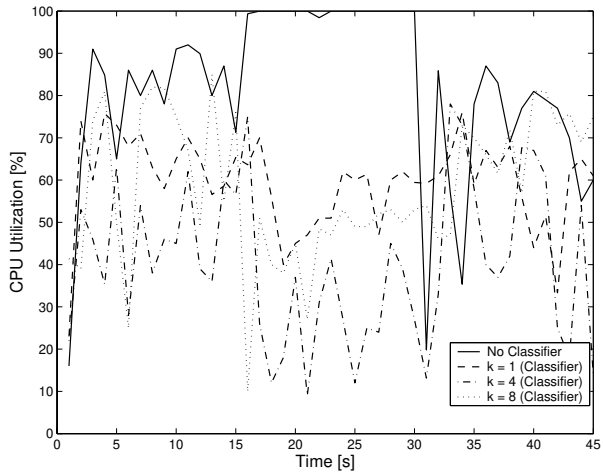


Figure 7.10: *CPU utilization of the IPSec GW1 in the WLAN with 3DES-SHA1 enabled and a DoS attack between 15 to 30 seconds.*

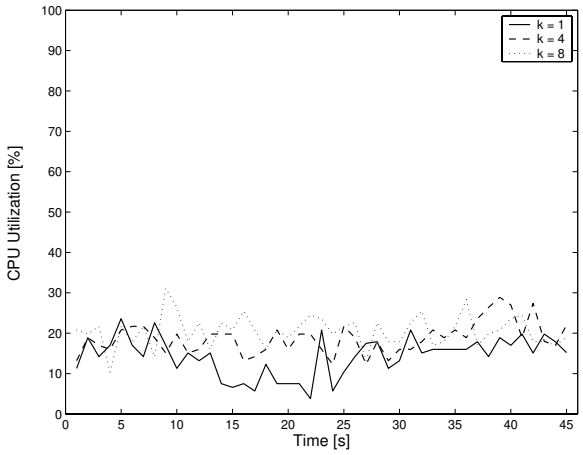


Figure 7.11: *CPU utilization of the classifier in the LAN environment with a launched DoS attack between 15 to 30 seconds.*

Optimal Number of Authentication Bits

In order to determine the optimal number k of authentication bits to use, Equation 7.1 (together with the results from the implemented hierarchical test scenarios and the simulation results from Chapter 6) is used. Equation 7.1 is further illustrated by Figure 7.12, in which the predicted packet sequence number decreases when T_a increases for $k < 5$. However, for $k > 5$ the packet sequence number is not a pronounced function of T_a anymore. The results from Figure 7.5 to Figure 7.11 further indicate that for $k = 4$ the optimal level is about to be reached since a DoS attack will have little impact on the security service performance while $k > 4$. The classifier is able to filter at least 94 % of the malicious packets when $k \geq 4$. It thus efficiently defends the target and defeats the DoS attack. Moreover, if $k < 4$ the incoming malicious packets still have an affect on the performance (throughput and CPU) since at least (for $k = 3$) about 13 % of the malicious packets will pass the classifier. By combining the simulation results (Chapter 6) with the results from Figure 7.5 to 7.12 the optimal number of k is obtained: $k = 5$

appears to be the optimal number of authentication bits, achieving the best compromise between performance and security.

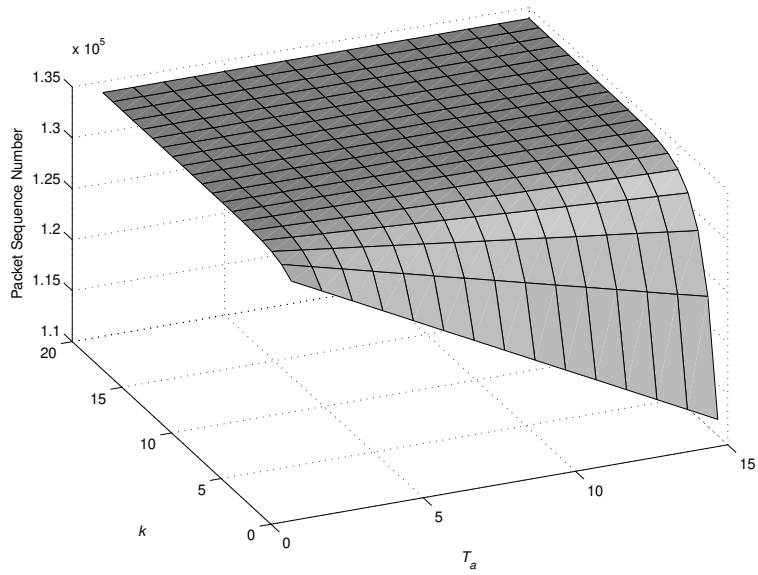


Figure 7.12: *Estimated packet sequence number versus the k and T_a parameter.*

7.5 Summary

In this chapter, a hierarchical defense structure was developed and evaluated aimed at preventing DoS attacks against a strong security service. Performance results, in terms of throughput and increments of packet sequence numbers as well as the CPU utilization of a IPSec gateway, were obtained by real measurements in LAN and WLAN environments. A filtering functionality was implemented in a *classifier*, classifying packets into good and bad ones and dropping the latter. Without using the first line of defense provided by the classifier, the results of the attack proved to be devastating and illustrated the heavy burden put on the IPSec gateway because of

the cryptographic functions used. The hierarchical defense structure, based on the RBWA [117] authentication protocol, proved to be able to decrease the amount of malicious traffic and to remove the DoS threat against the IPSec service itself. This was performed without introducing a new availability problem in the defense structure, which is due to the rather small amount of resources needed to authenticate the users on a per-packet level. From the results, the optimal number of additional authentication bits was determined to be as little as five.

Chapter 8

A Decision Model for Adequate Authentication

This chapter presents a practical decision model developed for finding the adequate authentication level based on desirable security criteria and alternatives. In the model, the Analytic Hierarchy Process (AHP) is used in the process of decision making.

8.1 Introduction

Much work has been put into the proposal and implementation of new and stronger security solutions. However, less effort has been directed towards the actual act of determining a sufficient security level with respect to different criteria. There are concerns when strong security is unnecessarily used, since the load on the processor and the power consumption is increased (even though we believe that the processing overhead is of greater concern than the power consumption). By offering security based on a need determined by a decision model it is possible to optimize security such as to reduce the cost. Performance and efficiency issues are particularly important in environments with constrained capacity.

The complexity of several authentication mechanisms coupled with the need to integrate authentication into environments with limited resources can make the selection of an adequate authentication level difficult. Therefore, it is necessary to develop a decision model which takes into consideration a wide range of factors; including objective and subjective aspects. In this chapter, a decision model is developed to improve the efficiency of security and to determine the most suitable authentication mechanism. The proposed decision model effectively reduces the range of available authentication solutions to the most appropriate alternative. Moreover, the model minimizes the impact of authentication algorithms while maintaining the specifically required level of authentication.

This research is not limited to a single operating system, hardware platform, or to a specific device. Furthermore, the model is not restricted to authentication only; it is also suitable for confidentiality or integrity.

The organization of this chapter is as follows. Section 8.2 describes the problem addressed in this chapter. Section 8.3 presents the idea of Always Best Security (ABS) in addition to a description of the system model in Section 8.4. Section 8.5 explains the fundamental of the AHP. Section 8.6 presents a case study outlining the selection of an adequate authentication level, including a definition of several criteria for two authentication alternatives. Finally, in Section 8.7 a summary concludes the chapter.

8.2 Problem Description

The selection of an adequate authentication level tends to be a complex process and depends upon a whole range of criteria. We define a criteria as *an element on which measurements can be made.* Examples of criteria are residual battery life and threat level. Normally, the criteria are difficult to quantify, meaning that defining and evaluating said criteria can be demanding. The criteria has to be measurable and also ranked in terms of a value for the decision model. With this in mind, the following problem is addressed and further illustrated by Figure 8.1:

Given a set of criteria regarding security attributes, for a set of security level alternatives, which is the best alternative for obtaining the overall goal of adequate authentication?

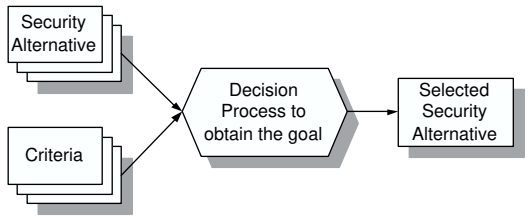


Figure 8.1: *Problem description.*

8.3 Always Best Security

The range of security mechanisms available today differ significantly in terms of complexity of cryptographic algorithms, algorithmic delay and speed, required resources (CPU and battery), key length, patents, etc. Also, security services may differ in terms of authentication, confidentiality, integrity, and nonrepudiation.

A decision making model allows us to collect information from the environment and effectively determine which security alternative best satisfies requirements according to a set of given demands. With the demand on low-power computing it is desirable to efficiently utilize available resources. Thus emerges the concept of Always Best Security (ABS). The term "best" is traditionally associated with strong security (which has a low probability of breaking a system and the use of long keys). It is not necessarily the strongest security mechanism which is the most appropriate for a certain device or network. Instead, the basic idea is to optimally combine various criteria, including: QoS parameters, personal preferences, device capabilities, application requirements, and available network resources.

8.4 System Model

This section describes a system model used to obtain the concept of ABS with different modules as illustrated by Figure 8.2. The model includes three modules. These are responsible for *information collection*, *decision making*, and *adequate authentication establishment*.

The objective is to supply devices in a constrained environment, whilst offering better resource utilization and maintaining adequate levels of authentication. The selection of an adequate authentication method is derived from several parameters collected from device and user preferences. If the collected information content is changed, the decision making process causes the system to select a different authentication method.

As depicted by Figure 8.2, the system model works as follows: initially, the criteria and alternatives need to be defined. Information is monitored and collected from predefined criteria. If any values have changed, the information is sent to the decision module. Otherwise, the model returns to the information collection module. When the decision module receives the values from the collection module, it determines whether a change of authentication level is necessary. If needed, the establishment module is informed and subsequently takes further measures. If the current authentication level is adequate, or a new authentication level is established, the model returns to the information collection module.

8.4.1 Information Collection Module

The information collection module receives valuable information from the criteria and updates the decision making module with new information, if necessary. The information contains a value between $1 - 9$ and quantitatively denotes the status of the defined criteria. The selection of the number scale is explained and motivated in [131]. How the collection and the translation from the measured value to the $1 - 9$ scale is performed depends upon the monitored criteria. This chapter does not, in detail, discuss the translation procedure. Additional work is needed to develop such a procedure, which is beyond the scope of this thesis.

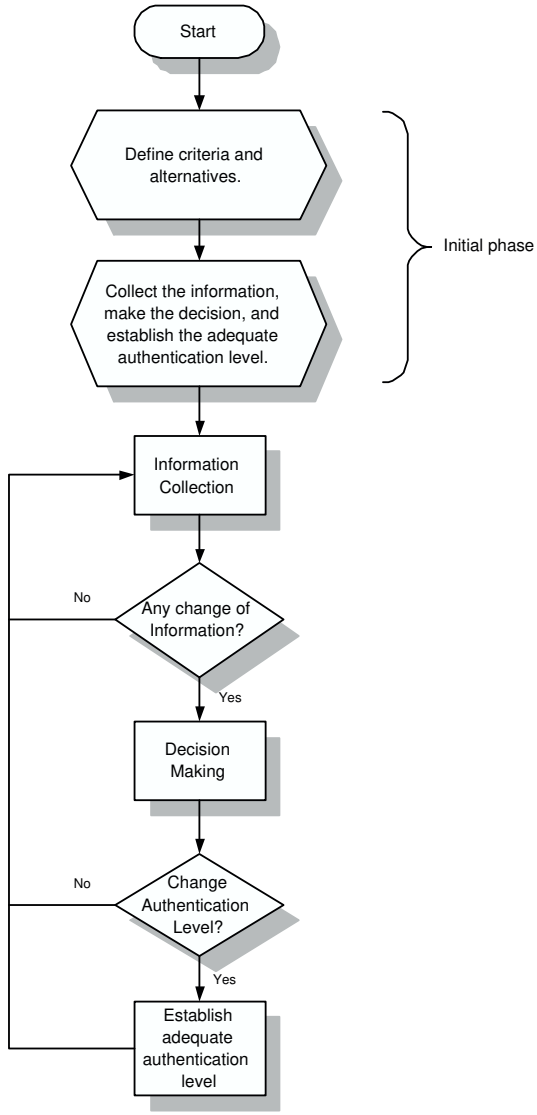


Figure 8.2: *System model.*

The information collection can be made either periodically or on demand. The periodic method may prove resource consuming if the resources of a device are limited, i.e., information might be collected unnecessarily when no change has occurred. Therefore, there is a tradeoff between collecting the information when required, and periodically collecting the information regardless of requirement. If resources are restrained, the periodic intervals should be carefully determined, since acquiring information results in processing delay or reduced battery life.

8.4.2 Decision Making Module

The decision module determines which authentication alternative best balances the tradeoff between security and performance and follows the concept of ABS. This is based on the incoming numeric data from the information collection module. The decision module contains any eligible decision process and outputs the most appropriate authentication method. The result is sent as input to the authentication establishment module. The decision module in this chapter relies on the AHP described in the following section.

8.5 The Analytic Hierarchy Process

The Multi-Criteria Decision Making (MCDM) methods form a well known branch of decision making, in which the AHP [4] is included. It is a general class of models which deal with decision problems in the presence of a number of criteria.

The AHP is a flexible decision process which is capable to formalize the process from quantitative and qualitative considerations of defined criteria. The developed system model uses AHP in the decision module to select the best security alternative with authentication in mind. The AHP was originally proposed by Saaty in [4] to support decision in management science [132]. Further publications of the AHP method include [133, 134, 135, 136, 137].

Using the AHP method in solving decision problems involves five major steps:

- *Step 1:* Construct the hierarchical structure by breaking down and decomposing the decision problem into several decision elements.
- *Step 2:* Create the input values by pair-wise comparisons of decision elements.
- *Step 3:* Estimate the relative weights of the decision elements.
- *Step 4:* Check for consistency.
- *Step 5:* Synthesize the priorities and combine the relative weights to determine the final set of ratings for the different decision alternatives.

Building a hierarchy (step 1) is as much an art as it is a science and the most inventive part of decision making involves the structuring of the decision as a hierarchy. This involves the decomposition of the problem into several elements according to their characteristics. In order to correctly and efficiently model complex decisions the following guidelines are followed:

1. A maximum number of nine elements are included in any set because experiments have shown that it is challenging for human beings to deal with more than nine factors at one time and this may result in less accurate priorities [131]. Moreover, as the number of elements being compared is increased, the measure of inconsistency decreases so slowly that there is insufficient space for improving the assessment as well as consistency.
2. Elements are clustered so that the clusters include comparable elements that do not differ too much from each other. Also, it is of importance that the elements on a lower level are comparable with the elements on the next higher level.

The basic form consists of a hierarchy structure with the goal at the top level. The second level holds the criteria, followed by the alternatives at level three, as illustrated by Figure 8.3.

In the second step, the judgements in the AHP are made in pairs. The scale used is represented by the intensity of importance between the criteria according to the *fundamental scale*, as illustrated by Table 8.1. The fundamental

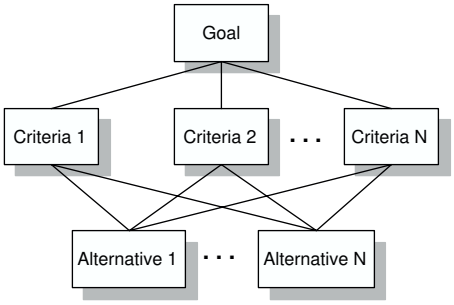


Figure 8.3: *A decomposition of the AHP goal.*

scale is validated according to effectiveness and theoretical justifications [4]. The scale consists of nine levels. To make it even more easy to judge a more restricted scale with five levels can be used; 1 is *equal*, 3 indicates *moderately more*, 5 *strongly more*, 7 *very strongly more* and 9 *extremely more*. A more refined scale is using all nine levels.

First, the criteria are compared pair-wise with respect to the goal. A $n \times n$ matrix, denoted as $\mathbf{A}$, is created using the comparisons with elements a_{ij}, indicating the value of criteria i relative to criteria j, then

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & & & & \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \tag{8.1}$$

The values in a_{ij} are then formed by the following rules: $a_{ii} = 1, a_{ij} = 1/a_{ji}, a_{ij} > 0, \forall\ i$. Therefore, if activity i has one of the above numbers assigned to it when compared with activity j, then j has the reciprocal value when compared with i.

After constructing the matrix of comparison, the next step is to determine the weights of the criteria, in which w_i is the weight of objective i in the

Intensity of importance	Definition	Explanation
1	Equal importance	Two activities contribute equally to the object
2	Weak	*Between Equal and Moderate*
3	Moderate importance	Experience and judgment slightly favor one activity over another
4	Moderate plus	*Between Moderate and Strong*
5	Strong importance	Experience and judgment strongly favor one activity over another
6	Strong plus	*Between Strong and Very Strong*
7	Very strong or demonstrated importance	An activity is favored very strongly over another; its dominance demonstrated in practice
8	Very, very strong	*Between Very Strong and Extreme*
9	Extreme importance	The evidence favoring one activity over another is of the highest possible order of affirmation

Table 8.1: *The Fundamental Scale for AHP [4].*

weight vector $\mathbf{w} = [w_1, w_1, \cdots, w_n]$ for n criteria. The objective is to obtain $\mathbf{w}$ from matrix $\mathbf{A}$ by finding the solution for some eigenvalue λ, in which

$$\mathbf{A} \cdot \mathbf{w}^T \quad = \quad \lambda \cdot \mathbf{w}^T. \tag{8.2}$$

In order to determine w_i an approximate solution is used that normalizes each column j in $\mathbf{A}$ such that

$$\sum_i a_{ij} = 1. \tag{8.3}$$

We denote the resulting normalized pair-wise matrix by $\mathbf{A}$'. For each row i in $\mathbf{A}$', compute the average value

$$w_i = \frac{1}{n} \sum_j a'_{ij} \tag{8.4}$$

where w_i is the weight of criteria i in the weight vector.

The next step is to check for consistency in order to trust the results. According to [4], three procedures are used, as follows:

1. Compute λ_{max}, which is the largest eigenvalue of matrix **A**
2. Compute the Consistency Index (CI), CI = $\frac{\lambda_{max} - n}{n - 1}$
3. Compute the Consistency Ratio (CR), CR = CI/RI
   - If CI = 0 then **A** is consistent
   - If CI/RI $\leq$ 0.10 then **A** is consistent enough
   - If CI/RI $>$ 0.10 then **A** is not consistent

RI is the average value of CI for randomly chosen entries in **A**, and it is obtained from Table 8.2 [4].

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.52	0.89	1.11	1.25	1.35	1.40	1.45	1.49

Table 8.2: *The RI values.*

The next step is to compare the alternatives to investigate which is more effective in satisfying each criteria on the level above. There are n matrices of judgements since there are n criteria and each matrix contains the weight for each alternative. The matrix is determined in a similar way as described above for w_i in Equation 8.4.

Finally, the last step is to select the alternative that best satisfies the goal by synthesizing the priorities. In the case of n criteria and m alternatives a matrix **B** of size $n \times m$ is created which contains the weight results b_{ij} for the alternative with respect to the criteria. For each j compute the overall weight or score s_i for each alternative by

$$s_i = \sum_i w_i \cdot b_{ij}. \tag{8.5}$$

The alternative with the largest s_{ij} is selected. This way of synthesizing the priorities is known as the distributive mode [4].

8.6 Case Study

To exemplify the proposed system model we consider the following case study, in which two scenarios are presented. In the case study, six criteria and two alternatives are identified, as illustrated by Figure 8.4.

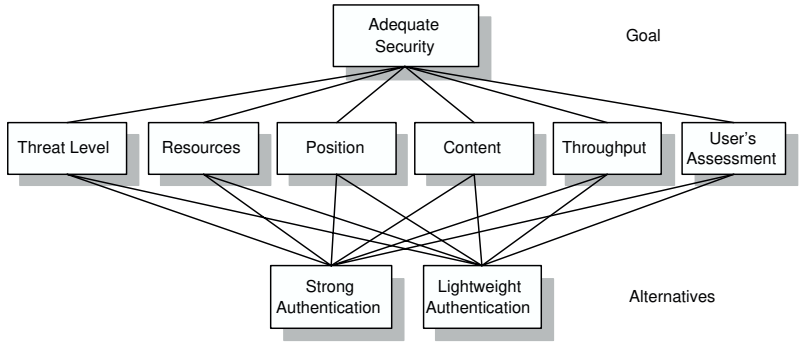


Figure 8.4: *AHP model for the case study.*

The decision process is further divided into the following steps:

- **Define the Overall Goal:** The objective is to find the adequate security level which is most appropriate to the requirements and the resources available. The definition of security is in this case only related to the authentication of a user.
- **Define the criteria:** To determine the preferred authentication level six criteria are considered for evaluation. The criteria are compared

against each other and against each alternative with regards to the overall goal. The criteria utilized by the information collection module could include any desired criteria defined by the user. The criteria used in this case study include *threat level*, *resources*, *data content*, *position*, *throughput*, and *user's assessment*.

Threat level (TL): The threat level element is utilized in order to detect threats and attacks and acts as an Intrusion Detection System (IDS). If the information from the threat level component indicates an ongoing attack, the decision making process will most likely select a stronger authentication method, provided that enough resources are available. A high threat value (within the $1-9$ scale) indicates a high probability of malicious activity.

Resources (RE): The resource element could be a combination of many components. For instance, it can include energy components such as remaining battery capacity or processing resources. If the resources consumed per packet is minimized, then the total resource consumption is also minimized. A high value means that the current resources are good and that the remaining battery time and the available processing capacity is satisfactory.

Position (PO): The location of the device is an important consideration when trust is involved. An unknown area could indicate that the device is in an insecure environment. The position is possible to determine, for example, with the help of the Global Positioning System (GPS) which provides the longitude and latitude coordinates. This information is, of itself, not of any help with respect to security. However, when the coordinates are appended with other information such as, "at home", "in the lab", or "at the airport", the position may prove valuable in terms of authentication and access control. In a WLAN environment, the SSID can be used to indicate the location, if GPS equipment is not available. The SSID information is not as accurate as GPS but could

still be very useful. A position with a high value sent to the decision module indicates that the device is in an insecure area.

Content (CO): Depending on the sensitivity of the content, it is possible to provide different levels of authentication strength. For instance, a video stream might not need a strong authentication method in comparison to a FTP session transferring a classified file. A content that sends a high value to the decision module indicates that the content is sensitive and needs strong authentication.

Throughput (TH): With regard to certain applications, the amounts of data transmitted to and from the device are very important. Therefore, adding extra traffic overhead for authentication purposes might be a burden. On the other hand, if throughput is not an issue, more authentication bits can be added to every outgoing packet. If a high value is presented to the decision module, high throughput is an issue and insufficient throughput might cause performance degradation.

User's Assessment (UA): The user's assessment of a situation could be very valuable when determining whether suspicious activity is going on in the environment. With this criteria the user of a device is able to add an extra, human dimension to the threat level assessment. Even though the IDS system is not able to detect an attack, the user might determine the presence of malicious activity. A passive attack is difficult to detect but with the human perspective involved, the chance to thwart such an attack increases. At the same time, the user should not panic and raise too many unnecessary alarms, which might be the result of such an enabled element. If the user determines that he or she does not want to interfere, the *threat level* value could be used as the default value sent to the decision making module. If the user believes there is an increasing threat in the environment, the value sent to the decision making module is set to a high number.

In addition, there might be other criteria that could be addressed. The proposed model makes it simple to extend and include more criteria and alternatives. Examples of other criteria are as follows:

- *Administration:* How user-friendly is the configuration and operation of the security system?
- *Reliability:* System development (design, implementation, etc.) should attempt to minimize the likelihood of accidental system bugs and malicious code. This may be accomplished by, for instance, reducing the complexity. Therefore, how strong is the reliability of the system?
- *Scalability:* What is the ability of a hardware or software system to adapt to new demands, i.e., how scalable is the system?
- *Availability:* A system should not be sensitive to the loss of, or reduction in, availability when, for instance, a DoS attack is launched. This also includes the terms robustness, fault-tolerance, and the ability of a system to function correctly in the presence of invalid inputs or stressful environmental conditions like packet loss. The potential of a system to detect the presence of an ongoing attack and the attempt to minimize the likelihood of accidental system bugs and malicious code are also important factors.

- **Define the Alternatives:** This step involves the array of possible alternatives which will satisfy the goal. It also forms the lowest level of the hierarchy structure. Two alternatives are defined with respect to the overall goal, namely *strong authentication* and *lightweight authentication*. The definition of the two alternatives are as follows:
  - *Strong authentication:* This is a process controlling the authenticity of the user's identity and the ability of a system to differentiate among people who do or do not have access. The process of generating and verifying the authentication bits is more complex than for the lightweight authentication process, i.e., the authentication bits are normally payload dependent, with the payload used as an input to a cryptographic function.

– *Lightweight authentication:* This process serves the same purpose as the strong authentication process but generates the authentication information in a less complex fashion and uses fewer authentication bits for each packet, as described in Chapter 5 and [2, 114, 117].

- **Perform Pair-wise Comparison of Criteria:** When the hierarchical structure is defined, the next step is to determine the relative importance between the criteria. AHP measures the strength of importance by pairwise comparison. This section explains the translation method of how to transform the values from the collection module so that the AHP can utilize them to create the comparison matrix of Equation 8.1. The scale value presented in Table 8.1 is used to make the comparison of how much more important the i^{th} criteria is compared with the j^{th} criteria. The comparison and translation method is performed by calculating the difference between the collected information values of criteria C_i and C_j as determined by Equation 8.6 and Equation 8.7:

$$C_i - C_j + 1 \quad for \;\; C_i \geq C_j \tag{8.6}$$

$$\frac{1}{|C_i - C_j - 1|} \quad for \;\; C_i < C_j \tag{8.7}$$

Equation 8.6 and Equation 8.7 follow Theorem 1 with corresponding proof:

> **Theorem 1:** In order to obtain a comparison value between two criteria which represent the intensity of importance, addition by 1 (according to Equation 8.6) and subtraction by 1 (according to Equation 8.7) is needed.
>
> **Proof:** The addition and subtraction by 1 is motivated by the use of Table 8.1, which declares that the intensity of importance is 1 when two criteria contribute equally to the object. Then, for $C_i = C_j$ the difference is equal to 0. In Table 8.1,

the equality value of 0 is represented by 1 and, therefore, addition and subtraction by 1 is performed. This also holds for $C_i \neq C_j$.

This is further clarified with an example, in which $C_i = 2$ and $C_j = 1$. According to Table 8.1, C_i has moderate or weak importance compared to C_j. By using Equation 8.6, in which $C_i - C_j + 1 = 2$, the pair-wise comparison value is calculated which is similar to the original definitions of importance for each criteria. Moreover, if the collected value for $C_i = 1$ and the value for $C_j = 7$, this results in a pair-wise comparison of 1/7. Since the difference between the criteria are negative the reciprocal value is used as in Equation 8.7.

After the pair-wise matrix is calculated, each entry in column i is divided by the sum of the entries in column i. This produces a normalized matrix, in which the sum of each column is equal to 1. By computing the average for each row i (according to Equation 8.4), the weights of the criteria and their relative degrees of importance are determined.

In the first scenario, the six criteria are numbered by the following invented values that are supposed to be gathered and determined by the collection module: TL = 7, RE = 7, PO = 3, CO = 2, TH = 7, and UA = 7. The translation of these values is performed and the pair-wise comparison matrix is constructed as presented in Table 8.3. The normalized matrix is presented in Table 8.4 with the sum of each column.

The second scenario is performed in a similar way as in the first scenario. The only difference is the threat level which is decreased from 7 to 2 and users's assessment from 7 to 4. The pair-wise comparison matrix of the criteria is then constructed as shown in Table 8.5 with its normalized matrix in Table 8.6.

	TL	RE	PO	CO	TH	UA
TL	1	1	6	5	1	1
RE	1	1	5	6	1	1
PO	1/6	1/5	1	2	1/5	1/5
CO	1/5	1/6	1/2	1	1/6	1/6
TH	1	1	5	6	1	1
UA	1	1	5	6	1	1
Sum	4.367	4.367	22.500	26.000	4.367	4.367

$\lambda_{max} = 6.049$, CI = 0.010, RI = 1.250, CR = 0.008

Table 8.3: *Pair-wise comparison matrix for the first scenario.*

	TL	RE	PO	CO	TH	UA	Weights
TL	0.229	0.229	0.267	0.192	0.229	0.229	0.229
RE	0.229	0.229	0.222	0.231	0.229	0.229	0.228
PO	0.038	0.046	0.044	0.077	0.046	0.046	0.050
CO	0.046	0.038	0.022	0.039	0.038	0.038	0.037
TH	0.229	0.229	0.222	0.231	0.229	0.229	0.228
UA	0.229	0.229	0.222	0.231	0.229	0.229	0.228
Sum	1	1	1	1	1	1	1

Table 8.4: *Normalized pair-wise comparison matrix with corresponding weights for the first scenario.*

- **Check for Consistency in the Pair-wise Comparison of Criteria:** It is possible that the construction of the pair-wise matrix is inconsistent. This is especially true for high-order matrices [4]. To improve an inconsistent matrix, one may be forced to reconsider pair-wise comparison until the consistency measure proves to be satisfactory [138]. If the CR is less than 0.10, the degree of consistency is satisfactory. However, if it is larger, inconsistencies may exist in the matrix and the results might not be valid. For scenario 1, the degree of consistency is satisfactory because of CR = 0.008, as shown in Table 8.3. The CR of 0.030 (Table 8.5) for scenario 2 is also satisfactory.

	TL	RE	PO	CO	TH	UA
TL	1	1/6	1	1/2	1/6	1/2
RE	6	1	5	6	1	4
PO	1	1/5	1	2	1/5	1/2
CO	2	1/6	1/2	1	1/6	1/3
TH	6	1	5	6	1	4
UA	2	1/4	2	3	1/4	1
Sum	18.0000	2.7833	14.5000	18.5000	2.7833	10.3333

$\lambda_{max} = 6.181$, CI = 0.037, RI = 1.250, CR = 0.030

Table 8.5: *Pair-wise comparison matrix for the second scenario.*

	TL	RE	PO	CO	TH	UA	Weights
TL	0.056	0.060	0.069	0.027	0.060	0.048	0.053
RE	0.333	0.359	0.345	0.324	0.359	0.387	0.351
PO	0.056	0.072	0.069	0.108	0.072	0.048	0.071
CO	0.111	0.060	0.035	0.054	0.060	0.032	0.059
TH	0.333	0.360	0.345	0.324	0.359	0.387	0.351
UA	0.111	0.090	0.138	0.162	0.090	0.097	0.115
Sum	1	1	1	1	1	1	1

Table 8.6: *Normalized pair-wise comparison matrix with corresponding weights for the second scenario.*

- **Perform Pair-wise Comparison of Alternatives:** A pair-wise comparison of the different alternatives with respect to the criteria and their local priorities are performed, in which each alternative needs to be compared with every criteria, forming a pair-wise comparison matrix. Table 8.7 illustrates the pair-wise comparison of the alternatives for the two scenarios. Once decided, these priorities are normally fixed values but can, if necessary, be modified. Table 8.7 is used for both scenarios, i.e., the priorities are not changed.
- **Decide what Alternative to Use:** The final step is to synthesize the

TL	S. Au.	L. Au.	Weights	RE	S. Au.	L. Au.	Weights
S. Au.	1	9	0.900	S. Au.	1	1/7	0.125
L. Au.	1/9	1	0.100	L. Au.	7	1	0.875

PO	S. Au.	L. Au.	Weights	CO	S. Au.	L. Au.	Weights
S. Au.	1	5	0.833	S. Au.	1	4	0.800
L. Au.	1/5	1	0.167	L. Au.	1/4	1	0.200

TH	S. Au.	L. Au.	Weights	UA	S. Au.	L. Au.	Weights
S. Au.	1	1/2	0.333	S. Au.	1	4	0.800
L. Au.	2	1	0.667	L. Au.	1/4	1	0.200

Table 8.7: *Pair-wise comparison matrix of alternatives with Strong Authentication (S. Au.) and Lightweight Authentication (L. Au.). For all the criteria* $\lambda_{max} = 2$*,* $CI = 0$ *and* $CR = 0$*.*

priorities and select the most desirable alternative. This involves creating a matrix which contains the weights of the alternatives with respect to each criteria, multiplying each column of the matrix by the weights of the corresponding criteria and adding across each row, as described in Equation 8.5. As a result, the *strong authentication* alternative, which has the largest priority (0.564), as illustrated by Table 8.8, is the selected and preferred authentication alternative for scenario 1. For scenario 2, the synthesis of the priorities results in Table 8.9 with a selected alternative of *lightweight authentication* (0.593). As noticed, the reduction of the threat level and user's assessment affect the final outcome. Even though the threat level and user's assessment is decreased, the priority result for the alternatives does not change much. The expectation is, of course, that the lightweight authentication alternative will be selected. However, the small difference between the final priority results further motivates the need for a practical decision model since the final outcome is not obvious. Decision making is a complex problem and, therefore, the selection of an adequate authentication level is difficult to make without the help of a decision model.

	TL	RE	PO	CO	TH	UA	
Weights	0.229	0.228	0.050	0.037	0.228	0.228	
S. Au.	0.900	0.125	0.833	0.800	0.333	0.800	0.564
L. Au.	0.100	0.875	0.167	0.200	0.667	0.200	0.436

Table 8.8: *Synthesis for the first scenario.*

	TL	RE	PO	CO	TH	UA	
Weights	0.053	0.351	0.071	0.059	0.351	0.115	
S. Au.	0.900	0.125	0.833	0.800	0.333	0.800	0.407
L. Au.	0.100	0.875	0.167	0.200	0.667	0.200	0.593

Table 8.9: *Synthesis for the second scenario.*

8.7 Summary

This chapter has reported on a system model to determine the adequate authentication level based on the necessary tradeoff between security and performance. The concept of ABS was introduced and a flexible system model was presented, containing three modules that are responsible for *information collection*, *decision making* and *establishment*. The decision module was based on a MCDM method, namely the AHP. From the outcome of the model it is possible to offer necessary security and to help users to choose the most efficient authentication strength based on desirable functionality and given circumstances.

To exemplify the proposed decision model a case study has been reported with two alternatives: *strong authentication* and *lightweight authentication*. The case study further outlined six criteria which were, in turn, weighted against the alternatives and prioritized with respect to the overall goal of adequate authentication.

The described decision making model is normally used for management and business related decisions in which the human aspect is involved to determine the importance between different criteria and alternatives. However, we

have shown that the decision method is useful even for a technical application in the security area to determine the appropriate authentication level between alternatives.

The proposed model is a step towards quantifying security based on qualitative measurements with judgement comparisons. Security has to be treated as a QoS attribute; with the possibility to decide between adequate security levels comes the possibility to trade it against other attributes. Additional work remains to be done in order to arrive to a comprehensive decision model. However, this work is an important first step towards selectable and adjustable security with QoS in mind.

Chapter 9

Concluding Remarks

The use of strong and sophisticated cryptographic protocols is a heavy burden on a device with constrained resources. Due to the fact that, most of the time, a network is not under attack, one may view the application of stronger security as an unnecessary deterioration of performance. Therefore, a desirable practice for forthcoming systems would involve the application of a fundament of lightweight security, which is able to detect an attack and increase the number of authentication bits or change the authentication protocol according to the threat level. Moreover, security is a complex issue that needs to consider packet loss, packet reordering and attacks. Additionally, security services can decrease the performance of an application due to resource limitation.

Another issue closely connected with security is the QoS. The notion of QoS has been expanded to include the level of authentication that can be offered in the presence of resource-constrained settings. Security procedures can be omitted in the interest of performance. However, this solution may not be acceptable where access control is required.

The next section provides a summary of the research conducted and addresses the issue of adjustable and lightweight authentication for network access control.

9.1 Summary of the Thesis

Lightweight Authentication: Novel link and network layer authentication protocols have been designed and evaluated. These protocols further contributed to the understanding of the tradeoff between performance and security. The protocols are resource efficient, able to handle per-packet authentication, robust in terms of handling packet loss, payload independent and adjustable due to a flexible number of authentication bits. Even though the protocols are lightweight, the probability of detecting an ongoing attack is high, which makes it possible to take countermeasures based on a predefined security policy.

Performance Measurement and Redundant Authentication: The motivation for this research was outlined and identified by IPSec performance degradation. The degradation was visually illustrated by throughput statistics, in which a devastating and saturated bottleneck behavior emerged if the traffic exceeded the capacity of the IPSec gateway. The target systems used in the tests were VPN tunnels with IPSec gateways for a low-end market. Further, this thesis discussed and presented the redundant authentication performed over IEEE 802.11, in combination with end-to-end security. The combination of first-hop (802.11i) and end-to-end (IPSec) security was applied due to privacy, access control, or accounting concerns. The disadvantage of a resource-constrained device is that the packets must be authenticated twice over the wireless link. Therefore, the developed authentication protocols have been presented as options for lightweight access control.

Hierarchical Authentication Structure: A hierarchical authentication structure has been developed and implemented in response to the possibility to launch a resource exhaustion attack against a security service with strong and resource-demanding cryptographic mechanisms. The implementation was performed, with the novel authentication protocols, as a classifier function in order to determine the legitimacy of incoming packets to the security service. IPSec was used as the security service and a laboratory test bed environment was built up. The results have clearly indicated the possibility to perform a

DoS attack against an IPSec gateway. The tests have also demonstrated the benefit of the hierarchical structure when it comes to mitigating the attack and relieving the pressure from the incoming and aggressive traffic flow. An important result of these tests was that the proposed hierarchical architecture was not sensitive itself to the flooding traffic generated by the adversary.

Decision Making: Finally, this thesis also presented the concept of Always Best Security (ABS), alongside a practical decision making model based on the Analytic Hierarchy Process (AHP). The decision making model takes into account a range of factors, including objective and subjective considerations, before selecting an adequate authentication level. It is a flexible model which is capable of formalizing quantitative and qualitative considerations of defined criteria with regards to QoS and security.

9.2 Future Work

Much of the research conducted in this thesis could be extended in order to constitute a number of interesting future research projects. Some suggestions include:

- Currently no mechanism has been developed to adaptively change the number of authentication bits. Such a mechanism would be of interest and also of practical importance if the research presented in the thesis is marketable. As a result of this thesis, a product is envisioned which may benefit companies and end users. However, although security is rather a process than a product, there is a demand for efficient and reliable security products which improve security in real time.
- A further step would involve the development of a type of *cognitive security*, not in the psychological sense, but as a notion of cognitive technology [22, 139, 140]. Cognition refers to the act of processing or knowing, including awareness, recognition, judgment, and reasoning. In this specific case, the envisioned system would be able to sense the

environment, learn how to handle threats and then take countermeasures to improve the overall security.

- To define a policy, including a threshold for the observed failure ratio, in order to take countermeasures when suspicious activity is detected. The challenge is to differ between malicious packets and valid packets, in order to lower the amount of false negatives and false positives. For instance, when a LAC is verified as incorrect, how do we know that this was caused by an adversary who guessed the authentication bits or caused by unsynchronized bit pointers?

- Quantitative measures of security constitute an important aspect of QoS for end users, and additional work in this field is needed. The challenge is to define useful methods of determining quantitative values in the security field, which are valuable and serve as an acceptable QoS parameter. Quantitative security is also related to the decision making area and it is of importance when selecting the adequate authentication level.

- Further research is needed to find an interface between the measured data and the Multi-Criteria Decision Making (MCDM) method considered by the decision model. Fuzzy set theory [141] and rough set theory [142] are two alternative tools used to classify a set of data. Is it possible to use fuzzy modeling in the preprocessing and classification of the measured data from the collection module?

- Authenticating and counting packets is critical to guarantee accounting correctness and prevent resource stealing. Usage-based accounting requires that the packets are examined and that the cost in terms of accounting overhead is reduced. Therefore, the presented lightweight authentication protocols are effective for secure usage-based accounting.

- In a WLAN, both reliable user authentication and mobility support are essential issues. However, re-authentication during handoff between two Access Points (APs) causes handoff latency, which affects the QoS for real time applications. Generally, user authentication is performed at

each AP and when a mobile station moves into the coverage area of an adjacent AP. Mobile stations need to be authenticated during and after handoff. During handoff, the mobile station performs a new user authentication procedure and receives a new key to secure the data over the wireless link. A critical issue is that the authentication mechanisms need to be responsive to the handoff time-scale required in micro-mobility environments [143]. It is, therefore, important to apply the proposed lightweight authentication protocols and investigate whether it is possible to minimize the authentication latency and obtain a less complex authentication procedure at the adjacent AP.

Appendix A

Latency Test Results

In Section 4.4, a measurement test is performed as configured in Figure 4.4. The objective is to examine the latency effect of IPSec over a wired link. The response times are presented in Table A.1 and Table A.2 for the different cryptographic algorithms (3DES-SHA1, 3DES-MD5, DES-SHA1, DES-MD5, SHA1 and MD5) with 95 % confidence intervals. The results are further illustrated by Figure 4.5 and Figure 4.6.

No IPSec		
Packet Size [bytes]	**Response Time [ms]**	**95 % Confidence Interval [ms]**
512	0.7865	0.0012
1024	1.1342	0.0004
4096	1.8678	0.0011
8192	2.8591	0.0014
3DES-SHA1		
Packet Size [bytes]	**Response Time [ms]**	**95 % Confidence Interval [ms]**
512	1.3912	0.0006
1024	2.0745	0.0012
4096	4.5746	0.0163
8192	7.6132	0.5538
3DES-MD5		
Packet Size [bytes]	**Response Time [ms]**	**95 % Confidence Interval [ms]**
512	1.3489	0.0005
1024	2.0183	0.0011
4096	4.4199	0.0243
8192	7.3623	0.2528
DES-SHA1		
Packet Size [bytes]	**Response Time [ms]**	**95 % Confidence Interval [ms]**
512	1.1943	0.0004
1024	1.6953	0.0006
4096	3.2714	0.0233
8192	5.2397	0.0035
DES-MD5		
Packet Size [bytes]	**Response Time [ms]**	**95 % Confidence Interval [ms]**
512	1.1566	0.0027
1024	1.6456	0.0341
4096	3.1400	0.0238
8192	4.9845	0.0084
SHA1		
Packet Size [bytes]	**Response Time [ms]**	**95 % Confidence Interval [ms]**
512	1.0159	0.0021
1024	1.4081	0.0203
4096	2.4526	0.0012
8192	3.6888	0.0003
MD5		
Packet Size [bytes]	**Response Time [ms]**	**95 % Confidence Interval [ms]**
512	0.9597	0.0012
1024	1.3504	0.0041
4096	2.3384	0.0122
8192	3.5120	0.0425

Table A.1: *The 95 % confidence intervals for the 2 GHz IPSec gateway.*

No IPSec		
Packet Size [bytes]	**Response Time [ms]**	**95 % Confidence Interval [ms]**
512	0.7249	0.0023
1024	1.0803	0.0003
4096	2.1682	0.0018
8192	3.2166	0.0024
3DES-SHA1		
Packet Size [bytes]	**Response Time [ms]**	**95 % Confidence Interval [ms]**
512	2.2755	0.0011
1024	3.6512	0.0014
4096	10.4356	0.0148
8192	19.0179	0.8568
3DES-MD5		
Packet Size [bytes]	**Response Time [ms]**	**95 % Confidence Interval [ms]**
512	2.1548	0.0011
1024	3.4742	0.0031
4096	9.8009	0.0243
8192	17.8186	0.7568
DES-SHA1		
Packet Size [bytes]	**Response Time [ms]**	**95 % Confidence Interval [ms]**
512	1.6480	0.0051
1024	2.4685	0.0097
4096	6.1377	0.0129
8192	10.6154	0.0000
DES-MD5		
Packet Size [bytes]	**Response Time [ms]**	**95 % Confidence Interval [ms]**
512	1.5332	0.0157
1024	2.2842	0.0881
4096	5.5025	0.0148
8192	9.4317	0.0085
SHA1		
Packet Size [bytes]	**Response Time [ms]**	**95 % Confidence Interval [ms]**
512	1.1891	0.0052
1024	1.7776	0.0180
4096	3.6779	0.0033
8192	5.4238	0.0089
MD5		
Packet Size [bytes]	**Response Time [ms]**	**95 % Confidence Interval [ms]**
512	1.0406	0.0072
1024	1.4601	0.0048
4096	2.8053	0.0292
8192	4.2668	0.0402

Table A.2: *The 95 % confidence intervals for the 300 MHz IPSec gateway.*

Appendix B

Histogram Difference Parameters

In Section 4.4.4, the following parameters characterize the histogram difference plots:

- *peak-to-peak* value = $\max\{\Delta h_i\} + |\min\{\Delta h_i\}| \in [0, 2]$;
- *width* = ΔR ($\max\{i|\Delta h_i \neq 0\}$ - $\min\{i|\Delta h_i \neq 0\}$).

The results in Table B.1 and Table B.2 describe the *peak-to-peak* and *width* values for the measurement performed in section 4.4.4. The algorithms used in the tests are 3DES-SHA1, 3DES-MD5, DES-SHA1, DES-MD5, AES-SHA1, and AES-MD5. Both Windows XP Professional Edition and Linux/Debian kernel 2.4.6 were used in the IPSec gateways. Moreover, the *relative parameter*

$$\sigma = \frac{\delta_{out} - \delta_{in}}{\delta_{in}} \quad (\sigma \geq -1) \tag{B.1}$$

for the measurements is determined from the results in Table 4.1 and Table 4.2 and is illustrated by Figure B.1, Figure B.2 and Figure B.3.

UDP Traffic (R_s^{in}) [Mbps]	Width [Mbps]	Peak-to-peak
No IPSec, Windows XP		
10.25	0	0
20.44	0	0
30.05	0	0
41.01	1	0.13
50.65	0	0
61.37	0	0
70.40	0	0
81.74	0	0
90.22	0	0
98.02	4	1.68
3DES-SHA1, Windows XP		
10.35	0	0
20.35	0	0
30.05	1	0.43
41.05	25	1.03
50.65	50	1.95
60.31	60	1.95
71.42	71	1.93
80.45	80	1.93
90.56	90	1.93
98.13	98	1.95
3DES-MD5, Windows XP		
10.59	0	0
20.63	0	0
30.43	0	0
40.69	2	0.90
50.42	50	1.75
61.02	61	1.85
70.05	72	1.83
81.69	81	1.95
91.24	91	1.95
96.49	98	1.90

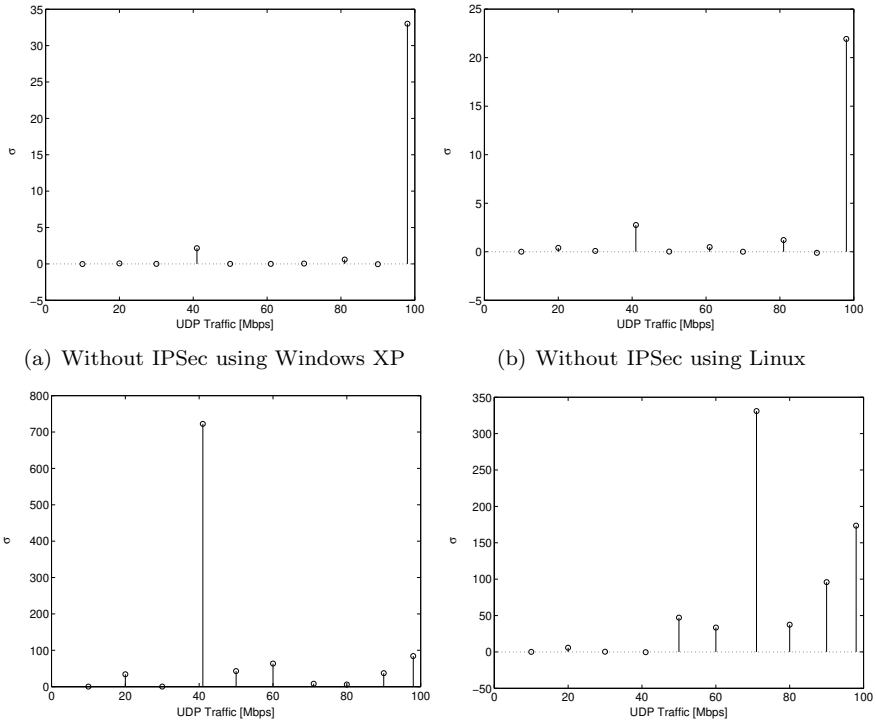
UDP Traffic (R_s^{in}) [Mbps]	Width [Mbps]	Peak-to-peak
No IPSec, Linux		
10.23	0	0
20.43	0	0
30.04	0	0
40.81	0	0
50.35	0	0
61.24	0	0
70.33	0	0
81.03	0	0
90.03	0	0
98.01	4	1.93
3DES-SHA1, Linux		
10.23	0	0
20.22	0	0
30.60	0	0
40.66	1	0.017
50.93	45	1.95
60.72	55	1.93
71.68	64	1.93
80.47	72	1.95
91.46	83	1.92
98.13	90	1.97
3DES-MD5, Linux		
10.10	0	0
20.42	0	0
30.15	0	0
40.36	0	0
51.20	0	0
60.71	9	1.90
70.81	19	1.83
80.94	31	1.78
91.43	41	1.85
98.03	98	1.55

Table B.1: *The width and peak-to-peak values for the measurement performed in section 4.4.4.*

UDP Traffic (R_s^{in}) [Mbps]	Width [Mbps]	Peak-to-peak
DES-SHA1, Windows XP		
10.59	0	0
20.63	0	0
30.43	0	0
40.69	0	0
50.42	1	0.033
61.08	41	0.8
70.41	62	1.5
80.28	80	1.65
91.24	89	1.92
94.80	98	1.82
DES-MD5, Windows XP		
10.60	0	0
20.63	0	0
30.43	0	0
40.71	1	0.033
50.42	2	0.17
61.08	4	0.7
70.91	49	1.17
80.48	71	1.32
91.29	89	1.88
96.50	98	1.84
AES-SHA1, Linux		
10.47	0	0
20.63	0	0
30.43	0	0
41.01	0	0
51.19	0	0
61.10	0	0
71.40	0	0
81.10	1	0.33
91.48	82	1.62
94.86	95	1.27

UDP Traffic (R_s^{in}) [Mbps]	Width [Mbps]	Peak-to-peak
DES-SHA1, Linux		
10.59	0	0
20.63	0	0
30.43	0	0
40.69	0	0
51.19	0	0
61.11	1	0.068
71.44	0	0
81.11	1	0.37
91.40	14	1.45
98.13	69	1.13
DES-MD5, Linux		
10.60	0	0
20.63	0	0
30.43	0	0
40.69	0	0
51.18	0	0
60.33	1	0.033
70.42	2	0.1
81.01	1	0.5
91.44	1	0.1
96.50	81	1.62
AES-MD5, Linux		
10.35	0	0
20.44	0	0
30.53	0	0
40.02	0	0
50.42	0	0
61.08	1	0.033
70.42	0	0
81.14	1	0.37
91.46	2	0.32
96.49	90	1.63

Table B.2: *The width and peak-to-peak values for the measurement performed in section 4.4.4.*



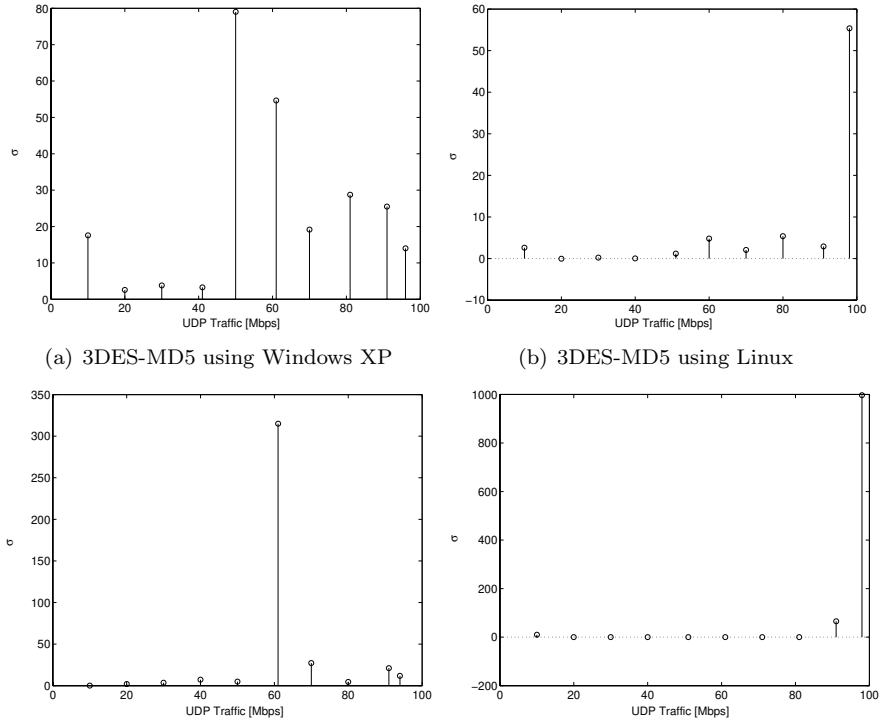
(a) Without IPSec using Windows XP

(b) Without IPSec using Linux

(c) 3DES-SHA1 using Windows XP

(d) 3DES-SHA1 using Linux

Figure B.1: *UDP traffic versus σ for 3DES-SHA1 and without IPSec using both Windows XP professional Edition and Linux/Debian.*



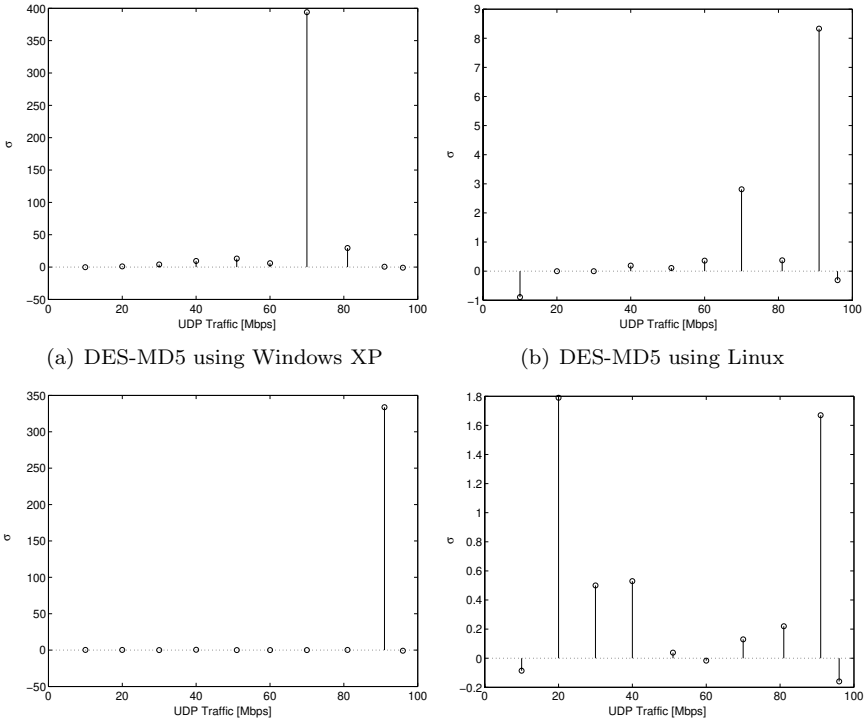
(a) 3DES-MD5 using Windows XP

(b) 3DES-MD5 using Linux

(c) DES-SHA1 using Windows XP

(d) DES-SHA1 using Linux

Figure B.2: *UDP traffic versus σ for 3DES-MD5 and DES-SHA1 using both Windows XP professional Edition and Linux/Debian.*



(a) DES-MD5 using Windows XP

(b) DES-MD5 using Linux

(c) AES-SHA1 using Windows XP

(d) AES-MD5 using Linux

Figure B.3: *UDP traffic versus σ for DES-MD5, AES-SHA1 and AES-MD5 using both Windows XP professional Edition and Linux/Debian.*

Bibliography

[1] H. Kvarnström. *On the Implementation and Protection of Fraud Detection Systems*. PhD dissertation, School of Computer Science and Engineering, Chalmers University of Technology, Göteborg, Sweden, 2004.

[2] S. F. Wu, H. Johnson, and A. Nilsson. SOLA: Lightweight Security for Access Control in IEEE 802.11. *IEEE CS Journal IT Professional*, 6:10–16, May/June 2004.

[3] R. Atkinson. Security Architecture for the Internet protocol. RFC 1851, Internet Engineering Task Force, August 1995. Available from: `www.ietf.org`.

[4] T. L. Saaty. *The Analytic Hierarchy Process*. McGraw-Hill, New York, 1980.

[5] ITU-T Recommendation E.800 (08/94) Terms and definitions related to Quality of Service and network performance including dependability.

[6] Wikipedia–The Free Encyclopedia. Available from: `http://en.wikipedia.org`.

[7] *Random House Webster's College Dictionary*. Random House, New York, 1997.

[8] International Organization for Standardization. Information Technology – Security techniques – Entity authentication mechanisms; Part 1: General Model. ISO/IEC 9798-1, Second Edition, September 1991.

[9] G. Tsudik. Message Authentication with One-Way Hash Functions. In *Proceedings of INFOCOM'92*, New York, USA, May 1992.

[10] O. Kim and D. Montgomery. Behavioral and Performance Characteristics of IPSec/IKE in Large-Scale VPNs. In *Proceedings of the IASTED International, Conference on Communication, Network, and Information Security*, New York, USA, December 2003.

[11] O. S. Elkeelany, M. M. Matalgah, K. P. Sheikh, M. Thaker, G. Chaudhry, D. Medhi, and J. Qaddour. Performance Analysis of IPSec Protocol: Encryption and Authentication. In *Proceedings of the IEEE Communications Conference (ICC 2002)*, pages 1164–1168, New York, USA, 2002.

[12] R. Rivest. The MD5 Message-Digest Algorithm. RFC 1321, Internet Engineering Task Force, April 1992. Available from: `www.ietf.org`.

[13] P. Jones. US Secure Hash Algorithm 1 (SHA1). RFC 3174, Internet Engineering Task Force, September 2001. Available from: `www.ietf.org`.

[14] R. Barbieri, D. Bruschi, and E. Rosti. Voice over IPSec: Analysis and Solutions. In *ACSAC'02: Proceedings of the 18th Annual Computer Security Applications Conference*, page 261, Washington D.C., USA, 2002. IEEE Computer Society.

[15] J. Rosseb, J. Ronan, and S. Davy. An analysis of IPSec deployment performance in high and low power devices. In *Proceedings of the 17th Nordic Teletraffic Seminar, Norway*, August 2004.

[16] J. Ronan, S. Davy, P. Malone, and M. Foghlu. Performance Implications of IPSec Deployment. In *Proceedings of the Interdomain Performance and Simulation (IPS2004)*, Budapest, Hungary, March 2004.

[17] J. Ronan, P. Malone, and M. Foghlu. Overhead Issues for Local Access Points in IPSec enabled VPNs. In *IPS Workshop, Salzburg*, April 2003.

[18] S. Lindskog. *Modeling and Tuning Security from a Quality of Service Perspective*. PhD dissertation, Department of Computer Science and Engineering, Chalmers University of Technology, Göteborg, Sweden, 2005.

[19] P. Schneck and K. Schwan. Authenticast: An Adaptive Protocol for High-Performance, Secure Network Applications. Technical Report GIT-CC-97-22, July 1997.

[20] C. S. Ong, K. Nahrstedt, and W. Yuan. Quality of protection for mobile applications. In *Proceedings of the 2003 IEEE International Conference on Multimedia & Expo (ICME'03)*, Baltimore, Maryland, USA, July 2003.

[21] S. Lindskog and E. Jonsson. Adding Security to Quality of Service Architectures. In *Proceedings of the SS-GRR Conference*, L'Aquila, Italy, August 2002.

[22] C. T. R. Hager. *Context Aware and Adaptive Security for Wireless Networks*. PhD dissertation, Department of Electrical and Engineering, Virginia Polytechnical Institute and State University, Blacksburg, Virginia, November 2004.

[23] CERT. Advisory CA-96.21: TCP SYN flooding and IP spoofing attacks, September 1996.

[24] C. Schuba, I. Krsul, M. Kuhn, G. Spaffod, A. Sunderam, and D. Zamboni. Analysis of a denial of service attack on TCP. In *Proceedings of the 1997 IEEE Symposium of Security and Privacy*, pages 208–223, Oakland, CA, May 1997.

[25] L. Ricciulli, P. Lincoln, and P. Kakkar. TCP SYN Flooding Defense. In *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference*, 1999.

[26] N. Ferguson and D. Senie. Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2267, IETF, January 1998. Available from: `www.ietf.org`.

[27] SANS Institute. Egress filtering V0.2, February 2000. Available from: `www.sans.org`.

[28] CERT. Advisory CA-1998-01: Smurf IP Denial-of-Service Attacks, 1998.

[29] D. Senie. Changing the Default for Directed Broadcasts in Routers. RFC 2644, Internet Engineering Task Force, August 1999.

[30] Berkeley Software Design Inc. BSDI releases defense for Internet Denial-of-Service attacks, October 1996.

[31] R. Bona. TCP syn attacks – a simple solution, October 1996.

[32] E. Shenk. Another way thought on dealing with SYN flooding, September 1996.

[33] P. Karn and W. Simpson. Photuris: Session-Key Management Protocol. RFC 2522, Internet Engineering Task Force, March 1999. Available from: `www.ietf.org`.

[34] P. Karn and W. Simpson. Photuris: Design criteria. In *Proceedings of the International Computer Congress ICC'99, LNCS, Springer Verlag*, Hong Kong, 1999.

[35] A. Zúquete. Improving The Functionality of SYN Cookies, 2002. Available from: `citeseer.ist.psu.edu/zuquete02improving.html`.

[36] C. Dwork and M. Naor. Pricing via Processing or Combatting Junk Mail. In *Proceedings of Advances in Cryptology CRYPTO'92, Lecture Notes in Computer Science, LNCS 740*, pages 139–147. Springer Verlag, August 1992.

[37] A. Juels and J. Brainard. Client puzzles: A cryptographic countermeasure against connection depletion attacks. In *Proceedings of the Network and Distributed System Security Symposium NDSS99*, San Diego, California, 1999.

[38] T. Aura, P. Nikander, and J. Leiwo. DoS-resistant authentication with client puzzles. In *Proceedings of the 8th International Workshop on Security Protocols*, Cambridge, UK, 2000.

[39] C. Meadows. A formal framework and evaluation method for network denial of service. In *Proceedings of 12th IEEE Computer Security Foundations Workshop*, pages 4–13, Mordano, Italy, August 1999.

[40] C. Meadows. A cost-based framework for analysis of denial of service in networks. *Journal of Computer Security*, 9(1–2):143–164, 2001.

[41] L. Gong and P. Syverson. Fail-stop protocols: An approach to designing secure protocols. In *Proceedings of the 5th International Working Conference on Dependable Computing for Critical Applications (DCCA-5)*, pages 44–55, University of Illinois at Urbana-Champaign, 1995.

[42] S. D. Pohekar and M. Ramachandran. Application of multicriteria decision making to sustainable energy planning – a review. *Renewable and Sustainable Energy Reviews*, 8(4):365–381, December 2004.

[43] K. K. Bachu. Risk Assessment in the Acquisition Process Using Analytic Hierarchy Process (AHP) Model Approach. In *Proceedings of the 38th Annual Air Traffic Control Association Convention*, Nashville, TN, 1993.

[44] L. W. Barnthouse, D. L. DeAngelis, R. H. Gardner, R. B. ONeill, and C. D. Powers. Methodology for Environmental Risk Analysis. Technical Report ORNL/TM-8167, Oak Ridge National Lab, 1982.

[45] M. Gerrits, M. A. Ridgley, and F. R. Rijsberman. Risk prioritization and site selection for sanitation in a Rhine estuary. In *Proceedings of the Third International Symposium on the Analytic Hierarchy Process*, Washington D.C., USA, July 1994.

[46] W. Walsh, I. Susel, and A. Ronayne. Setting risk based priorities– a method for ranking site for response. In *Proceedings of the National Research and Development Conference on the Control of Hazardous Materials*, Anaheim, California, 1991.

[47] M. Toshtzar. Multi-Criteria Decision making Approach to Computer Software Evaluation: Application of the Analytic Hierarchy Process. *Mathematical and Computer Modeling*, 11:276–281, 1988.

[48] G. Fatemeh and B. A. Calway. Risk analysis of the end user computing. In *Proceedings of the Fourth International Symposium on the Analytic Hierarchy Process*, pages 541–546, Simon Frasier University, Burnaby B.C., Canada, July 1996.

[49] M. Kitamura. Knowledge engineering approach to risk management and decision making problems. *Reliability Engineering and System Safety*, 38(12):67–70, 1992.

[50] J. L. Riggs, S. B. Brown, and R. P. Troublood. Integration of technical, cost and schedule risks in project management. *Computers and Operations Research*, 21(5):521–533, 1994.

[51] S. A. Butler. Improving Security Technology Selections with Decision Theory. In *Proceedings of the Third Workshop on Economics-Driven Software Engineering Research (EDSER-3)*, Toronto, Canada, 2001.

[52] M. Svahnberg, C. Wohlin, L. Lundberg, and M. Mattsson. A method for understanding quality attributes in software architecture structures. In *SEKE'02: Proceedings of the 14th international conference on Software engineering and knowledge engineering*, pages 819–826, New York, NY, USA, 2002. ACM Press.

[53] J. Karlsson and K. Ryan. A Cost-Value Approach for Prioritizing Requirements. *IEEE Software*, 14(5):67–74, September/October 1997.

[54] J. Karlsson, C. Wohlin, and B. Regnell. An Evaluation of Methods for Prioritizing Software Requirements. *Journal of Information and Software Technology*, 39(14–15):939–947, 1998.

[55] M. Svahnberg. An industrial study on building consensus around software architectures and quality attributes. *Information & Software Technology*, 46(12):805–818, 2004.

[56] K. Sallhammar and S. J. Knapskog. Using game theory in stochastic models for quantifying security. In *Proceedings of the 9th Nordic Workshop on Secure IT Systems (Nordsec 2004)*, Espoo, Finland, November 2004.

[57] K. B. B. Madan, K. V. Goseva-Popstojanova, and K. S. Trivedi. A method for modeling and quantifying the security attributes of intrusion tolerant systems. *Performance Evaluation*, 56:167 – 186, 2004.

[58] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright, J. Dobson, J. McDermid, and D. Gollmann. Towards operational measures of computer security. *Journal of Computer Security*, 2:211–229, October 1993.

[59] K. Lye and J. M. Wing. Game strategies in network security. In *Proceedings of the 15th IEEE Computer Security Foundations Workshop*, Cape Breton, Nova Scotia, Canada, 2002.

[60] K. B. B. Madan, K. V. Goseva-Popstojanova, and K. S. Trivedi. Modeling and quantification of security attributes of software systems. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN'02)*, Washington D.C., 2002.

[61] R. Ortalo and Y. Deswarte. Experimenting with quantitative evaluation tools for monitoring operational security. *IEEE Transactions on Software Engineering*, 25(5):633–650, Sept/Oct 1999.

[62] IEEE Std. 802.11-1999. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification, reference number ISO/IEC 8802-11:1999(E)*, 1999.

[63] IEEE Std. 802.11a. *Supplement to Part 11: Wireless LAN medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-speed Physical Layer in the 5 GHz Band*, 1999.

[64] IEEE Std. 802.11b. *Supplement to Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-speed Physical Layer Extension in the 2.4 GHz Band*, 1999.

[65] IEEE Std. 802.11g/D1.1-2001. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band*, 2001.

[66] S. R. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the Key Scheduling Algorithm of RC4. In *Proceedings of the Eighth Annual Workshop on Selected Areas in Cryptography (SAC'01)*, Toronto, Canada, 2001.

[67] N. Borisov, I. Goldberg, and D. Wagner. Unsafe at any key size: an analysis of the WEP encapsulation. In *Proceedings of the International Conference on Mobile Computing and Networking*, March 2000.

[68] J. Walker. Intercepting mobile communications: The insecurity of 802.11. Technical report, 03628E, IEEE 802.11 committee, July 2001.

[69] Federal Information Processing Standards Publication (FIPS) 197, The Advanced Encryption Standard (AES), November, 2001. Available from: `http://csrc.nist.gov/CryptoToolkit/aes/`.

[70] M. Bellare, J. Kilian, and P. Rogaway. The security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences*, 61(3):362–399, 2000.

[71] R. Thomas. ISP Security BOF III. The North American Network Operators' Group (NANOG) meeting, June 2003.

[72] A. Frier, P. Karlton, and P. Kocher. The SSL Protocol Version 3.0. Technical report, Internet Engineering Task Force, November 1996. draft-freier-ssl-version3-02.txt.

[73] T. Dierks and C. Allen. The TLS Protocol Version 1.0. RFC 2246, Internet Engineering Task Force, January 1999. Available from: `www.ietf.org`.

[74] T. Ylonen. The SSH (Secure Shell) Remote Login Protocol. Technical report, Internet Engineering Task Force, November 1995. draft-ylonen-ssh-protocol-00.txt.

[75] B. Bellman. Do-it-yourself VPNs. *Business Communications Review*, 32:28–32, 2002.

[76] T. Yager. Low-end VPNs secure networks affordably. *InfoWorld*, 19:61–69, 2001.

[77] National Institute of Standards and Technology. *NISTIR 90-4250: Secure Data Network Systems (SDNS) Network, Transport and Message Security Protocols*, February 1990.

[78] *ISO-IEC DIS 11577 – Information technology – Telecommunications and Information Exchange Between Systems – Network Layer Security Protocol*.

[79] D. P. Andersson et al. A Protocol for Secure Communication in Large Distributed Systems. Technical Report UCB/UCSD 87/342, University of California, Berkeley, 1987.

[80] J. B. Postel. Internet Protocol. RFC 791, Internet Engineering Task Force, September 1981. Available from: `www.ietf.org`.

[81] J. Ioannidis and M. Blaze. The Architecture and Implementation of Network-Layer Security Under Unix. In *Proceedings of the Fourth Usenix Security Symposium*, San Diego, CA, October 1993.

[82] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 1883, Internet Engineering Task Force, January 1996. Available from: `www.ietf.org`.

[83] W. Simpson. IP in IP Tunneling. RFC 1853, Internet Engineering Task Force, October 1995. Available from: `www.ietf.org`.

[84] P. Metzger, P. Karn, and P. Simpson. The ESP Triple DES-CBC Transform. RFC 1825, Internet Engineering Task Force, October 1995. Available from: `www.ietf.org`.

[85] P. Metzger, P. Karn, and P. Simpson. The ESP DES-CBC Transform. RFC 1829, Internet Engineering Task Force, August 1995. Available from: `www.ietf.org`.

[86] P. Metzger and W. Simpson. IP Authentication using Keyed SHA. RFC 1852, Internet Engineering Task Force, October 1995. Available from: `www.ietf.org`.

[87] P. Metzger and W. Simpson. IP Authentication using Keyed MD5. RFC 1828, Internet Engineering Task Force, August 1995. Available from: `www.ietf.org`.

[88] R. Atkinson. IP Authentication Header. RFC 2402, Internet Engineering Task Force, November 1998. Available from: `www.ietf.org`.

[89] R. Atkinson. IP Encapsulating Security Payload. RFC 2406, Internet Engineering Task Force, November 1998. Available from: `www.ietf.org`.

[90] R. Pereira and R. Adams. The ESP CBC-Mode Cipher Algorithms. RFC 2451, Internet Engineering Task Force, November 1998. Available from: `www.ietf.org`.

[91] H. Orman. The OAKLEY Key Determination Protocol. RFC 2412, Internet Engineering Task Force, November 1998. Available from: `www.ietf.org`.

[92] D. Maughan, M. Schneier, and M. Schertler. Internet Security Association and Key Management Protocol. RFC 2408, Internet Engineering Task Force, November 1998. Available from: `www.ietf.org`.

[93] H. Krawczyk. SKEME: A Versitile Secure Key Exchange Mechanism for Internet. In *IEEE Proceedings of the 1996 Symposium on Network and Distributed Systems Security*, San Diego, CA, 1996.

[94] S. Franker, S. Kelly, and R. Glenn. The AES Cipher Algorithm and its Use with IPSec. RFC 3602, Internet Engineering Task Force, September 2003. Available from: `www.ietf.org`.

[95] Openswan software package. Available from: `www.openswan.org`.

[96] M. Bustosand T. Van Herck and M. Kaeo. Terminology for Benchmarking IPSec Devices. Technical report, Internet Engineering Task Force, March 2004. draft-ietf-bmwg-ipsec-term-03.

[97] Netperf, A Network Performance Benchmark. Available from: {`ftp://ftp.cup.hp.com/dist/networking/benchmarks/netperf/`}.

[98] D. Mills. Network Time Protocol (version 3) Specification, Implementation, and Analysis. RFC 1305, Internet Engineering Task Force, March 1992. Available from: `www.ietf.org`.

[99] T. Lindh. A new approach to performance monitoring in IP networks - combining active and passive methods. In *Proceedings of the Passive and Active Measurements Workshop 2002*, Forth Collins, USA, 24-26 March 2002.

[100] M. Fiedler and K. Tutschku. Application of the Stochastic Fluid Flow model for Bottleneck Identification and Classification. In *Proceedings of Design, Analysis and Simulation of Distributed Systems 2003 (DASD '03)*, Orlando, FL, April 2003.

[101] Ethereal network protocol analyser. Available from: `www.ethereal.com`.

[102] P. Arlos. *On the Quality of Computer Network Measurements*. PhD dissertation, Blekinge Institute of Technology, Sweden, October 2005.

[103] M. Fiedler, K. Tutschku, P. Carlsson, and A. Nilsson. Identification of Performance Degradation in IP Networks Using Throughput Statistics. In *Proceedings of the 18th International Teletraffic Congress – ITC-18*, Berlin, September 2003.

[104] C. Ahlberg and B. Shneiderman. Visual Information seeking: Tight coupling of dynamic query filters with starfield displays. In *Proceedings of the CHI'94: Human Factors in Computing Systems*, pages 313–317, Vienna, Austria, 1994.

[105] W. Yurcik, K. Lakkaraju, J. Barlow, and J. Rosendale. A prototype tool for visual data mining of network traffic for intrusion detection. In *Proceedings of the ICDM Workshop on Data Mining for Computer Security (DMSEC'03)*, Melbourne, FL, 2003.

[106] L. Girardin. An eye on network intruder-administrator shootouts. In *Proceedings of the Workshop in Intrusion Detection and Network Monitoring (ID'99), Berkeley, CA. USENIX Assoc*, 1999.

[107] K. Muniandy. Case study: Visualizing time related events for intrusion detection. In *Proceedings of the IEEE Symposium on Information Visualization*, Salt Lake City, Utah, 2000.

[108] T. Takada and H. Koike. Tudumi: Information visualization system for monitoring and auditing computer logs. In *Proceedings of the 6th International Conference on Information Visualization*, London, UK, 2002.

[109] M. Ankerst, M. Ester, and H. P. Kriegel. Towards an effective cooperation of the user and computer for classification. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining (KDD'00)*, Boston, Massachusetts, 2000.

[110] S. T. Teoh, K. L. Ma, S. F. Wu, and X. Zhao. Case study: Interactive visualization for Internet security. In *Proceedings of the IEEE Conference on Visualization 2002*, pages 505–508, Boston, Massachusetts, 2002.

[111] S. T. Teoh, K. L. Ma, and S. F. Wu. Visual exploration process for the analysis of Internet routing data. In *Proceedings of the IEEE Conference on Visualization 2003*, pages 523–530, Seattle, Washington, 2003.

[112] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen. A survey of energy efficient network protocols for wireless networks. *Wireless Networks*, 7(4):343–358, 2001.

[113] F. Guo and T. Chiueh. Sequence Number-Based MAC Address Spoof Detection. In *Proceedings of the International Symposium on Recent Ad-*

vances in Intrusion Detection (RAID), Seattle, Washington, September 2005. Springer-Verlag.

[114] H. Johnson, A. Nilsson, J. Fu, S. F. Wu, A. Chen, and H. Huang. SOLA: A one-bit identity authentication protocol for access control. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'02), Taipei, Taiwan*, pages 768–772, November 2002.

[115] Random number generation and testing. Available from: `http://csrc.nist.gov/rng`.

[116] Specification of the Bluetooth System, V.1.1, 2001.

[117] F. Zhao, Y. Shin, S. F. Wu, H. Johnson, and A. Nilsson. RBWA: An Efficient Random-Bit Window-based Authentication Protocol. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'03), San Francisco, CA*, pages 1379–1383, December 2003.

[118] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical Network Support for IP Traceback. In *Proceedings of the ACM SIGCOMM'02*, Pittsburgh, PA, August 2002.

[119] M. G. Gouda, C. T. Huang, and E. Li. Anti-replay window protocols for secure ip. In *Proceedings of the 9th IEEE International Conference on Computer Communications and Networks*, Las Vegas, October 2000.

[120] C. T. Huang and M. G. Gouda. An anti-replay window protocol with controlled shift. In *Proceedings of the 10th IEEE International Conference on Computer Communications and Networks*, Scottsdale, October 2001.

[121] M. R. Spiegel. *Theory and Problems of Probability and Statistics*. New York: McGraw-Hill, 1992.

[122] R. C. Tripathi et al. Some Generalizations Of The Geometric Distribution. *Sankhya – the Indian Journal of Statistics Series B*, 49:218–223, 1987.

[123] D. Moore, M. V. Geoffrey, and S. Savage. Inferring Internet Denial-of-Service Activity. In *Usenix Security Symposium*, Washington D.C., August 2001.

[124] Fyodor. Ping of death. 1996. Available from: `http://www.insecure.org/sploits/ping-o-death.html`.

[125] D. Dean, M. Franklin, and A. Stubblefield. An Algebraic Approach to IP traceback. In *Proceedings of the Network and Distributed System Security Symposium*, pages 3–12, San Diego, California, February 2001.

[126] S. Savage, A. Karlin D. Wetherall, and T. Andersson. Network Support for IP Traceback. In *ACM/IEEE Transactions on Networking, vol. 9, no 3*, pages 226–237, June 2001.

[127] M. T. Goodrich. Efficient Packet Marking for Large-Scale IP Traceback. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS)*, pages 117–126, Washington D.C., November 2002.

[128] S. Ioannidis and S. M. Bellovin. Implementing Pushback: Router-Based Defense Against DDoS Attacks. In *Proceedings of the Network and Distributed System Security Symposium*, San Diego, California, February 2002.

[129] 3Com Corporation. Available from: `www.3com.com`.

[130] The Netfilter project. Available from: `www.netfilter.org`.

[131] T. L. Saaty and M. S. Ozdemir. Why the magic number seven plus or minus two. *Mathematical and Computer Modeling*, 38:233–244, 2003.

[132] D. R. Anderson, D. J. Sweeney, and T. A. Williams. *Quantitative Methods for Business.* Thompson / South-Western, Mason, OH, 9. edition, 2004.

[133] T. L. Saaty. How to make a decision: The analytic hierarchy process. *European Journal of Operational Research*, 48(1):9–26, 1990.

[134] T. L. Saaty. Homogeneity and clustering in AHP ensures the validity of the scale. *European Journal of Operational Research*, 72:598–601, 1994.

[135] T. L. Saaty and L. G. Vargas. *The Logic of Priorities*. Kluwer Nijhoff Publishing, Boston, 1982.

[136] T. L. Saaty and M. S. Ozdemir. Negative priorities in the analytic hierarchy process. *Mathematical and Computer Modeling*, 37:1063–1075, 2003.

[137] T. L. Saaty and L. G. Vargas. *Models, Methods, Concepts and Applications of the Analytic Hierarchy Process*. Kluwer Academic Publishing, Dordrecht the Netherlands, 2001.

[138] P. Harker. An analytic hierarchy approach for the determination of interregional migration patterns, presented at the Association of American Geographers annual meeting, Washington D.C., April, 1984.

[139] T. W. Rondeau, C. J. Rieser, B. Le, and C. W. Bostian. Cognitive Radios with Genetic Algorithms: Intelligent Control of Software Defined Radios. In *Proceedings of SDR04*, pages 3–8, Phoenix, August 2004.

[140] J. Mitola and G. Q. Maquire. Cognetive Radio: Making Software Radios More Personal. *IEEE Personal Communications*, 6(4):13–18, 1999.

[141] L. A. Zadeh. Fuzzy sets. In *Proceedings of Inform. Control 8*, 1965.

[142] Z. Pawlak. Why rough sets? In *Proceedings of 1996 IEEE International Conference on Fuzzy Systems*, pages 738–743, Piscataway, NJ, 1996.

[143] A. T. Campbell and J. Gomez-Castellanos. IP micro-mobility protocols. *SIGMOBILE Mob. Comput. Commun. Rev.*, 4(4):45–53, 2000.

ABSTRACT

The increasing use of Internet access networks raises the demand for secure and reliable communication for both users and businesses. Traditionally, the aim has been to provide the strongest possible security. However, with the demand for low-power computing it has become desirable to develop security mechanisms which efficiently utilize available resources. The tradeoff between performance and security plays an important role. In general, strong security is added even if there is no attack. The implementation of strong and resource demanding security often implies more than a secure system; it may deteriorate the performance of a device with limited resources and pave the way for new threats such as resource exhaustion. It is, therefore, unwise to use strong cryptographic algorithms for devices with limited resources in the absence of an adversary. It is more efficient to begin with lightweight security, taking further measures when an attack is detected.

The overall focus of this thesis is on adjustable and lightweight authentication protocols for network access control. The thesis studies the performance degradation of strong security using empirical tests on IP security (IPSec) with a visual bottleneck indicator based on the time-discrete fluid flow model and throughput histogram differences. The results emphasize the possibility of a Denial of Service (DoS) attack against IPSec itself.

The redundant authentication performed in a Wireless Local Area Network (WLAN) also motivates the development and evaluation of novel lightweight authentication protocols for the link and network layer. The developed authentication protocols are resource efficient, per-packet based, and robust in terms of handling packet loss. The protocols are further used as part of a hierarchical defense structure, which has been implemented and evaluated, in order to mitigate protocol based DoS attacks.

Finally, this thesis presents the concept of Always Best Security (ABS) and a practical decision making model based on the Analytic Hierarchy Process. The model takes a number of factors into consideration, including subjective and objective aspects of security in order to select an adequate authentication level. It is a flexible model which formalizes quantitative and qualitative considerations of a defined set of criteria, keeping Quality of Service in mind.

2005:09

ISSN 1653-2090
ISBN 91-7295-077-3