Martin Eian

Robustness in Wireless Network Access Protocols

Thesis for the degree of Philosophiae Doctor

Trondheim, September 2012

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics
and Electrical Engineering
Department of Telematics

NTNU – Trondheim
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Telematics

ISBN 978-82-471-3762-8 (printed ver.)
ISBN 978-82-471-3763-5 (electronic ver.)
ISSN 1503-8181

Doctoral theses at NTNU, 2012:230

Printed by NTNU-trykk

Abstract

Wireless network access protocols are used in numerous safety critical applications. Network availability is essential for safety critical applications, since loss of availability can cause personal or material damage. An adversary can disrupt the availability of a wireless network using denial of service (DoS) attacks.

The most widely used wireless protocols are vulnerable to DoS attacks. Researchers have published DoS attacks against IEEE 802.11 local area networks (LANs), IEEE 802.16 wide area networks (WANs) and GSM and UMTS mobile networks.

In this work, we analyze DoS vulnerabilities in wireless network protocols and define four categories of attacks: *jamming* attacks, *flooding* attacks, *semantic* attacks and *implementation specific* attacks. We identify *semantic* attacks as the most severe threat to current and future wireless protocols, and as the category that has received the least attention by researchers.

During the first phase of the research project we discover semantic DoS vulnerabilities in the IEEE 802.11 communication protocols through manual analysis. The 802.11 standard has been subject to manual analysis of DoS vulnerabilities for more than a decade, thus our results indicate that protocol vulnerabilities can elude manual analysis.

We conclude that formal methods are required in order to improve protocol robustness against semantic DoS attacks. We propose a formal method that can be used to automatically discover protocol vulnerabilities. The formal method defines a protocol model, adversary model and cost model. The protocol participants and adversary are modeled as finite state transducers, while the cost is modeled as a function of time. Our primary goal is to construct a formal method that is *practical*, i.e. does not require a vast amount of resources to implement, and *useful*, i.e. able to discover protocol vulnerabilities. We verify and validate our proposed method by modeling the 802.11w amendment to the 802.11 standard using Promela as the modeling language. We then use the SPIN model checker to verify the model properties and experiments to validate the results.

The modeling and experiments result in the discovery and experimental validation of four new deadlock vulnerabilities that had eluded manual analysis. We find one deadlock vulnerability in 802.11i and three deadlock vulnerabilities in 802.11w. A deadlock vulnerability is the most severe form of communication protocol DoS vulnerabilities, and their discovery and removal are an essential part of robust protocol design. Thus, we conclude that our proposed formal method is both practical and useful.

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of philosophiae doctor (PhD) at the Norwegian University of Science and Technology (NTNU). The research was carried out at the Department of Telematics (ITEM) from 2007 to 2011 under the supervision of Professor Stig F. Mjølsnes and Professor Steinar H. Andresen. The research was funded by the Faculty of Information Technology, Mathematics and Electrical Engineering (IME).

I feel privileged to have had the opportunity to study at the Department of Telematics. I had a fantastic time during my four years as a PhD student thanks to the past and present faculty, technical and administrative staff, researchers and students. You all helped make ITEM such a great place to be. Thank you to everyone that supported me during my study, I would not have been able to achieve these results without your help.

I would especially like to thank my main supervisor Professor Stig F. Mjølsnes for his invaluable support, advice, guidance and inspiration. His frequent insights and challenges helped me learn, understand and achieve the results that we aimed for.

Furthermore, I want to thank the evaluation committee members Professor Richard Kemmerer, Dr. Cas Cremers and Professor Øivind Kure for all the time and effort they spent on evaluating this thesis.

I greatly appreciate the hospitality from everyone at the Information Security Institute (ISI) at the Queensland University of Technology (QUT) in Brisbane, Australia. I spent three months there as a visiting researcher in 2010, with Associate Professor Andrew Clark as my local supervisor. Those three months led to a breakthrough in my research, and I am very grateful for all the support from the people at ISI. Special thanks to Professor Colin Boyd, Professor Colin Fidge, Associate Professor Andrew Clark and Dr. Suriadi Suriadi for helping me find the path to the conclusion of my research.

To mamma, pappa, Hilde and Dag, thank you for all your support and encouragement. My extreme curiosity as a child was probably annoying at times, but at the same time a powerful force driving me towards research. You realized that before I did, and encouraged me to go further.

Last, but definitely not least, to my wonderful wife Jing. This thesis is dedicated to you and to our baby. You are always there for me, and you truly understand me. I feel incredibly lucky to have you as my wife. This work concludes an important chapter in our lives, and I look forward to our new adventures together.

Table of Contents

Part I

Thesis Introduction

Chapter 1

Introduction

1.1 Terminology

An electronic communication system consists of entities (e.g. laptop computers, mobile phones and web servers) interconnected by an electronic communication network[1]. The communication network provides a *service*: the interchange of data between the entities. Communication networks are divided into two categories by the physical medium used for the transmission of data: *wired* networks and *wireless* networks. A wired network transmits data through physical cables (e.g. copper or fiber optic cables), while a wireless network transmits data through an existing medium such as air, water, rock or empty space. The most commonly used physical medium for wireless communication networks is radio waves through air, to such an extent that the term "wireless network" now is interchangeable with the term "radio network".

Communication networks are also divided into two categories based on topology: *access* networks and *transport* ("backbone") networks. An access network provides a communication service to endpoints, i.e. the entities that wish to communicate. A transport network aggregates traffic from multiple access networks and acts as a traffic exchange between them.

Communication protocols define the rules for *formatting of data* and *entity behaviour* in electronic communication systems. Data that is to be transmitted through the network is formatted as a *message*[2]. A message contains the data transmitted by the entity and/or data used to manage the communication network. The entity behaviour defines when and how an entity should transmit a message and how to react when it receives a message. The entity behaviour can be modeled as a *state machine*[3], where the action taken by the entity depends on previous sent and received messages. The state machine consists of *states* and *transitions* between the states.

A denial of service (DoS) attack against a communication network prevents

[1]The communication network also consists of entities such as switches and routers, but this level of detail is not necessary for understanding the work presented here.

[2]We use the term "message" as a synonym for a protocol data unit (PDU).

[3]Some communication protocols, e.g. Aloha, are stateless and are not modeled using state machines. We only consider stateful communication protocols in this work.

authorized access to the communication service. A communication protocol may contain vulnerabilities that can be exploited by DoS attacks. We define that a communication protocol is *robust* if it has no such vulnerabilities.

In this work, we investigate the robustness of communication protocols used at the data link layer in wireless access networks. Whenever we use the term "protocol" by itself, we implicitly mean "communication protocol".

1.2 Motivation

Our research effort is motivated by two current trends. The first trend is that wireless networks are increasingly being used in safety critical applications. Wireless networks are used in life critical medical devices, public safety communications, road safety systems, supervisory control and data acquisition (SCADA) systems and alarm and surveillance systems. The second trend is that these wireless networks are commonly constructed using commercial off the shelf (COTS) equipment. COTS equipment offers significant cost reductions due to economies of scale and competition between equipment manufacturers. The communication protocols used by COTS equipment are defined by standards, developed and published by organizations such as IEEE and 3GPP. Standard protocols facilitate the interoperability of equipment from different manufacturers.

Safety critical applications require network availability, since loss of the network communication service could cause or aggravate personal and material damage. Communication protocol reliability and dependability has been studied extensively for several decades. The results of this research have improved network protocol robustness against random or accidental failures that cause loss of availability. Furthermore, the protocol security properties confidentiality, integrity and authenticity are well understood due to extensive research. State of the art protocols are not vulnerable to any known attacks against these properties. Conversely, the research on protocol robustness against intentional failures, or directed attacks against service availability, is not a mature area of research. Such attacks, denial of service (DoS) attacks, disrupt the availability of a network communication service. The research on wireless network protocols during the last two decades has shown that the protocols currently in use are vulnerable to a wide array of DoS attacks. Furthermore, the results indicate that protocol DoS vulnerabilities frequently elude manual analysis during the protocol design process.

Formal methods have been successfully applied to the analysis of the confidentiality, integrity and authenticity properties of protocols. The motivation for the use of formal methods to verify these properties is that experience has shown that manual analysis is not sufficient if the security goals are to be achieved. The same line of reasoning could be applied to the analysis of protocol availability. Thus, the primary goal of our research is to determine if and how formal methods can be applied to the analysis of protocol DoS vulnerabilities, and thus contribute to the construction of more robust protocols.

1.3 Methodology, Research Goals and Results

We divide the research project into three main phases. In the first phase, we perform a literature review. Our primary goals are to identify known attacks against current protocols, to categorize the attacks, and to find related work on the use of formal methods to model protocol availability. Section 1.5 presents the related work on formal methods. We identify four attack categories during the first phase: *jamming* attacks, *flooding* attacks, *semantic* attacks and *implementation specific* attacks. Section 1.4 presents the categories in more detail. The last category, implementation specific attacks, is not related to the protocol design. Thus, this category of attacks is considered out of scope. The results from the first phase indicate that one particular category of DoS attacks, the semantic attacks, is not well understood. Furthermore, the history of published attacks against wireless access network protocols show that semantic attacks are the most severe DoS threat to such networks. Semantic attacks are easy to implement, require no specialized hardware, and can be performed with very little effort. Conversely, a significant research effort has been spent on modeling jamming and flooding attacks. The results of this research are formal methods and models to discover and verify jamming and flooding vulnerabilities. Thus, semantic attacks are selected as the focus of the work in the second and third phases.

In the second phase, we select the medium access control (MAC) protocols of the IEEE 802.11 standard as the protocols to be studied and modeled. We have three main motives for selecting 802.11. First, several protocol DoS vulnerabilities had already been discovered in the 802.11 standard. The standard had been subject to extensive manual analysis, thus it is an appropriate test subject for the use of formal methods. Second, readily available software and hardware for experimentation with 802.11 facilitate the experimental validation of our analytical results. Furthermore, we can experiment without the need for a spectrum license, since 802.11 supports the unlicensed 2.4 GHz industrial, scientific and medical (ISM) frequency band and the 5 GHz Unlicensed National Information Infrastructure (U-NII) frequency band. Third, a new amendment to the standard, 802.11w, was recently developed. Part of the motivation for this amendment is to prevent semantic DoS attacks. We thus focus on 802.11w in particular, to study the effects of the protocol modifications proposed by the amendment. The second phase consists of manual analysis of the protocols and experimental validation of the DoS vulnerabilities found. The results in Paper A in Chapter 2 and Paper B in Chapter 3 show that significant and practical vulnerabilities are discovered by this manual analysis.

In the third phase, we specify, implement and verify a formal protocol model, adversary model and cost model. Furthermore, we experimentally validate the verification results. Our primary goals of this phase are to construct a practical formal method for the analysis of semantic DoS vulnerabilities in protocols, and to determine if this formal method can be used to find new vulnerabilities that had eluded manual analysis. The results in Paper C in Chapter 4 and Paper D in Chapter 5 show that we discover four new deadlock vulnerabilities through the use of the proposed formal method. Furthermore, we demonstrate that a simple model can yield significant and useful results.

1.4 Denial of Service Attack Categories

We divide wireless network DoS attacks into four categories. The categories are based on the published attacks found in the research literature. The four categories are *jamming*, *flooding*, *semantic* and *implementation specific*.

An *implementation specific* attack targets implementation vulnerabilities in software or hardware, and involves the transmission of invalid protocol messages. The adversary may use an invalid message format, an invalid message payload, or both, to trigger a bug in the protocol implementation. A classic example of an implementation specific attack is the buffer overflow, where an adversary can overwrite memory segments to cause a crash in the software processing the messages. Buffer overflows can be avoided by strict bounds checking in software, but are still a significant problem in deployed systems. Implementation specific attacks are not related to the protocol design, since a correct protocol implementation would discard invalid messages. We thus consider implementation specific attacks as out of scope for our investigation.

A *jamming* attack is mounted by emitting noise in the radio frequency band used by the wireless network. The noise reduces the signal to noise ratio, causing a degradation of communication service performance or a DoS. The effectiveness of a jamming attack depends on the adversary's emission power and the bandwidth of the wireless network and emitted noise. A network that uses frequency hopping over a broad frequency band is less susceptible to jamming than a network that uses a narrow frequency band for communications. Furthermore, the physical layer protocols used by the wireless network can amplify the effects of a jamming attack. For example, a jammer can cause DoS against wireless networks that use carrier sense (CS) mechanisms for medium access even if the emitted noise is low power and narrowband. If the adversary can correctly modulate the noise so that the communicating parties interpret it as a valid signal, then the network will not be able to provide a communication service. An adversary typically has to use specialized hardware to mount a jamming attack. An exception to this rule is the case where test modes on standard network equipment can be exploited to turn a standard networking interface card into a jammer. An example of a jamming attack exploiting network interface card test modes is the 802.11 DSSSTEST mode attack described by Wullems et al. [1]. This jamming attack causes a DoS even with low power emissions by the adversary due to the use of a CS mechanism in the 802.11 physical layer.

A *flooding* attack disrupts the network communication service through the exhaustion of resources. The targeted resource could be bandwidth, computational capacity, memory capacity or available energy. An adversary will transmit a large number of valid protocol messages during a flooding attack. If the recipient has to perform expensive computations for each message, then the flood of messages may exhaust all the available computational resources. Similarly, if the recipient has to store state information for each message, then the result may be a total exhaustion of available memory. Transmitting a large number of messages to a battery powered device could exhaust all the energy available to the recipient, thus causing a DoS condition. A classic example of a flooding attack is the Transmission Control Protocol (TCP) Synchronize (SYN) attack [2]. The TCP SYN message is

the first message of the TCP 3-way handshake used for connection establishment. The recipient stores a state for each connection. If the adversary transmits a large number of TCP SYN messages, then the recipient will eventually exhaust the available memory used to store connection states, and thus refuse to accept new connections. Proposed countermeasures against flooding attacks include cryptographic puzzles [3] to offset the recipient's computational cost and client cookies [4] to offset the recipient's memory cost. A flooding attack can be mounted using standard network equipment, since the adversary transmits valid protocol messages.

A *semantic* attack exploits protocol vulnerabilities to desynchronize the protocol state. An adversary mounts a semantic attack by transmitting one or more valid protocol messages. The semantic attack triggers a state transition in the recipient, which can cause a non-synchronized state for the communicating parties. The non-synchronized state may be permanent (deadlock) or temporary. One example of a semantic attack is the deauthentication attack against IEEE 802.11 networks [5], where the adversary can reset the state of a protocol participant by transmitting a single 802.11 Deauthentication Notification message. The protocol participants must exchange several messages before they are able to recover to a synchronized state. The deauthentication attack illustrates an important property: semantic protocol vulnerabilities can function as an amplifier for the adversary. By transmitting a single message, the adversary forces the participants to exchange multiple messages in order to recover. Due to this property, an adversary exploiting semantic vulnerabilities can cause a DoS condition with less transmission time than an adversary using jamming or flooding attacks. A semantic attack can be mounted using standard network equipment, since the adversary only transmits valid protocol messages. The discovery and prevention of protocol vulnerabilities to semantic DoS attacks is the main focus of the work presented in this report.

1.5 Related Work

In 1994, Needham published an article giving an example of a denial of service threat and possible countermeasures [6]. In this article, he points out the lack of research effort on this topic:

> Security threats are often divided into three categories: breach of confidentiality, failure of authenticity, and unauthorized denial of service. The first two have been very extensively studied; confidentiality in particular has been pursued to extraordinary lengths. Indeed, some publications on confidentiality recall medieval disputes about how many angels could stand on the head of a pin. The second has been the subject of inquiry for many years, and is remarkable for the extent to which it is easy to devise wrong protocols. The third has been much less studied, and indeed, the tendency has been to dismiss it as a topic for serious inquiry [...]

Meadows proposed a formal framework for evaluating network denial of service attacks in 1999 [4]. The framework can be used to model authentication and

key agreement protocol vulnerabilities to *flooding* attacks. One of Meadows' key observations is that the adversary models used in cryptographic protocols, such as the Dolev-Yao model [7], are not suitable for modeling denial of service. The Dolev-Yao adversary can delete any network message, and thus always cause denial of service. Meadows introduced adversary and defender *cost* functions as part of the model, since a realistic adversary will have limited resources. The cost functions are not defined in Meadows' framework, so a user of the framework has to quantify cost and define the cost functions. Examples of cost include computational cost, memory cost, energy cost, monetary cost and bandwidth cost. The goal of the framework is to verify that the protocol will stop its execution if the cumulative defender cost exceeds a tolerance bound.

Leiwo et al. published a survey of *flooding* attacks and protocol vulnerabilities in 2000 [8]. Their paper also proposed protocol design principles to reduce protocol vulnerabilites to flooding attacks.

In 2003, Meadows published a review paper on formal methods for protocol analysis [9]. In this paper, denial of service attacks were identified as a growing threat, but the paper concludes that little work had been done on applying formal methods to analyze the denial of service robustness of protocols. Research challenges identified in this paper include developing new adversary models and the development of models, methods and tools to provide assistance to protocol designers.

Further results based on Meadows' framework were published by Ramachandran [10], Smith [11] and Tritilanunt [12] in 2002, 2007 and 2009, respectively. Other approaches to modeling *flooding* attacks include game theory [13, 14] and process algebra [15].

The research efforts on modeling flooding attacks have been motivated by the threat of Internet distributed denial of service (DDoS) attacks. In a DDoS attack, the adversary initiates a large number of protocol intances. The modeling of flooding attacks has thus focused on balancing the cost between the protocol initiator and the protocol responder so that a high cost for the responder implies a high cost for the initiator. The most significant difference between the related work on flooding attacks and our work is that the models for flooding attacks are based on the assumption that the adversary initiates the protocol instance. Furthermore, they assume that the adversary cannot directly interfere with established protocol instances. These assumptions are valid for DDoS attacks, but they are not valid for semantic attacks against wireless access networks. An adversary can eavesdrop on all messages and transmit messages at any time when attacking a wireless access network. Our proposed method and models are not based on these assumptions, our adversary is able to transmit messages at any time during a protocol run and to interfere with established protocol instances.

Pelechrinis et al. published a comprehensive survey of *jamming* attacks, detection and prevention in 2010 [16]. The survey includes quantitative jamming efficiency metrics and adversary goals. The adversary goals are defined as maximized jamming gain, targeted jamming and reduced probability of detection.

To the best of our knowledge, the only other research effort to model *semantic* protocol vulnerabilities is presented in a paper by Narayana et al. [17]. They construct a formal model of a subset of the 802.16 MAC layer protocols using the temporal

logic of actions (TLA+), then use the TLA+ model checker (TLC) to discover semantic DoS vulnerabilities. One major difference between the results in [17] and the work presented here is that Narayana et al. did not discover any significant protocol vulnerabilities through the use of their formal model.

1.6 List of Papers, Summary and Contributions

The goal of the work presented in this report is to contribute to the construction of robust wireless access network protocols. In particular, the work focuses on how to construct formal models for the verification of semantic DoS vulnerabilities in protocols. The research results indicate that manual analysis of protocols is not sufficient, and that formal methods should be used in order to detect and prevent protocol vulnerabilities. We propose a method for how to construct and verify protocol, cost and adversary models, and demonstrate how the models can be used to discover severe vulnerabilities in widely used protocols. Our research results are published in four papers:

Paper A, Chapter 2
Martin Eian
"Fragility of the Robust Security Network: 802.11 Denial of Service"
Proceedings of the 7th International Conference on Applied Cryptography and Network Security (ACNS'09)

Paper B, Chapter 3
Martin Eian
"A Practical Cryptographic Denial of Service Attack Against 802.11i TKIP and CCMP"
Proceedings of the Ninth International Conference on Cryptology And Network Security (CANS 2010)

Paper C, Chapter 4
Martin Eian and Stig F. Mjølsnes
"The Modeling and Comparison of Wireless Network Denial of Service Attacks"
Proceedings of the 3rd ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handhelds (MobiHeld '11)

Paper D, Chapter 5
Martin Eian and Stig F. Mjølsnes
"A Formal Analysis of IEEE 802.11w Deadlock Vulnerabilities"
Proceedings of the 31^{st} *Annual IEEE International Conference on Computer Communications (IEEE INFOCOM 2012)*

Paper A presents a manual analysis of DoS vulnerabilities in the the 802.11 medium access control (MAC) layer with the 802.11i and 802.11w amendments. This

paper makes three principal contributions. First, we present and analyze a previously unknown DoS vulnerability in the 802.11 standard. Second, we experimentally validate the new DoS vulnerability together with the deauthentication attack and another vulnerability discovered by J. Epstein in 2007 [18]. Third, we propose a robust solution to the MAC layer DoS vulnerabilities in 802.11.

Paper B presents a manual analysis of a cryptographic DoS vulnerability in the 802.11i Temporal Key Integrity Protocol (TKIP). This paper makes five principal contributions. First, we analyze the 802.11 standard and discover a highly efficient cryptographic DoS attack. Second, we show that the attack also works against clients using counter mode with cipher block chaining message authentication code protocol (CCMP) as the pairwise cipher in networks that support both TKIP and CCMP. Third, we demonstrate that the attack works even if 802.11e quality of service (QoS) support is disabled in the AP. Fourth, we implement the attack and experimentally validate the analytical results. Fifth, we propose a robust solution to the vulnerability and temporary measures to limit the exposure to the vulnerability.

Paper C presents a formal method and model for evaluating wireless network protocol vulnerabilities to semantic DoS attacks. We analyze the adversary goals to find an appropriate quantification of the adversary cost. We then quantify the protocol participant cost, and propose an attack efficiency definition. Finally, we use our model to discover a new deadlock vulnerability in the IEEE 802.11 family of standards, followed by an experimental validation of the vulnerability. The proposed formal method is not protocol specific, it can be used to analyze any wireless protocol.

Paper D presents our application of the formal method proposed in Paper C for the analysis of deadlock vulnerabilities in the IEEE 802.11 MAC layer with the 802.11i and 802.11w amendments. The main contribution in Paper D is a demonstration of *how* formal methods can be used to find deadlock vulnerabilities. In particular, we investigate how to automatically discover vulnerabilities through the construction and verification of a formal protocol model. Our work bridges the gap between theory and practice by giving a detailed description of how to construct and verify a simple and useful protocol model, including the complete model source code. The proposed approach to modeling and verification could help protocol designers discover deadlock vulnerabilities at an early stage of the design process.

Our contributions could help communication protocol designers find and amend semantic DoS vulnerabilities. However, semantic DoS vulnerabilities is still not a mature research area, and there are several open problems. The next section describes open research problems and suggested directions for future research within this field.

1.7 Open Research Problems

A major challenge is how to integrate the different types of models used to evaluate protocol properties. Currently, a protocol designer would have to use one model to evaluate confidentiality and authenticity properties, another model to evaluate vulnerabilities to flooding attacks, and yet another model to evaluate vulnerabilities

to semantic attacks. Ideally, a generic protocol description language and modeling tool that could verify all of these properties should be constructed. This would significantly reduce the work required by the designer, and thus further the use of formal methods in protocol design.

The cost model proposed in this work is simple. A simple cost model facilitates the model construction, but a more realistic cost model may provide a higher degree of validity. The cost model must be based on the adversary constraints, and the model assumptions must be carefully evaluated. For example, we assume that an adversary will limit his transmission time to a minimum. If this assumption does not hold, then the model verification results may not be valid.

Another avenue of research could be to investigate the real time properties of protocols. Our proposed model does not support the verification of real time properties. Thus, timing related semantic DoS attacks might be possible even against a protocol that has been verified as not vulnerable using our model. One would have to use a model checking tool with real time support, such as Uppaal, in order to model real time properties.

In a more long time perspective, protocol design principles should be derived from the model verification results. As common types of attacks and vulnerabilities are identified, techniques to counter them could be developed. Protocol design principles that counter flooding attacks are fairly well understood today, but the same does not hold true for semantic attacks.

Finally, other protocols could be modeled and verified using our proposed method. We have only constructed a model of the 802.11 protocols, but our method could be used on other wireless protocols, such as the protocols used in 802.16, 802.22, GSM, UMTS and LTE.

1.8 Thesis Structure

The rest of the Thesis is structured as follows: Part II contains four papers. All four papers have been peer reviewed before being accepted for publication. The papers have undergone minor editing and reformatting before their inclusion in this thesis.

Paper A, in Chapter 2, presents semantic DoS vulnerabilities in 802.11w and a proposed solution to remove the vulnerabilities. The paper was published and presented at the 7th International Conference on Applied Cryptography and Network Security (ACNS'09) in Paris-Rocquencourt, France [19]. The proceedings were published in the Springer Lecture Notes in Computer Science (LNCS) series.

Paper B, in Chapter 3, presents a new cryptographic DoS vulnerability in 802.11i. The paper was published and presented at the Ninth International Conference on Cryptology And Network Security (CANS 2010) in Kuala Lumpur, Malaysia [20]. The proceedings were published in the Springer Lecture Notes in Computer Science (LNCS) series. The results presented in Paper A and Paper B were obtained by manual analysis in the second phase of the research project.

Paper C, in Chapter 4, proposes a formal method for the modeling of semantic DoS vulnerabilities. Furthermore, it presents a deadlock vulnerability in 802.11i discovered through the use of the proposed method. The paper was published and

presented at the 3rd ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handhelds (MobiHeld 2011) in Cascais, Portugal [21]. Chapter 4 also includes additional material that was removed from Paper C due to space constraints and thus not peer reviewed. The parts that are not peer reviewed are included as subappendices (4.A, 4.B and 4.C). This additional material illustrates in detail how the proposed cost model in Paper C could be used to quantify the cost of semantic DoS attacks against 802.11.

Paper D, in Chapter 5, applies the formal method proposed in Paper C to the formal analysis of 802.11w. Three new deadlock vulnerabilities were found in 802.11w through automatic model checking of the formal model. The paper illustrates the detailed construction of the formal model, and includes the complete model source code. The paper was published and presented at the 31^{st} Annual IEEE International Conference on Computer Communications (IEEE INFOCOM 2012) in Orlando, Florida, USA [22]. The results presented in Paper C and Paper D were obtained during the third phase of the research project.

Part II

Included Papers

Chapter 2

Paper A

Published in:

Martin Eian

"Fragility of the Robust Security Network: 802.11 Denial of Service"

Proceedings of the 7th International Conference on Applied Cryptography and Network Security (ACNS'09)

Lecture Notes in Computer Science (LNCS), vol. 5536
Springer Verlag, 2009
ISBN 978-3-642-01956-2

Abstract

The upcoming 802.11w amendment to the 802.11 standard eliminates the 802.11 deauthentication and disassociation Denial of Service (DoS) vulnerabilities. This paper presents two other DoS vulnerabilities: one vulnerability in draft 802.11w implementations discovered by IEEE 802.11 TGw, and one new vulnerability in 802.11, which is still present in the 802.11w amendment. Attacks exploiting the first vulnerability are significantly more efficient than any known 802.11 DoS attacks, while attacks exploiting the second vulnerability have efficiency and feasability equivalent to a disassociation attack. This paper provides an experimental verification of these attacks, demonstrating their feasability using freely available software and off the shelf hardware. Finally, the root cause of these vulnerabilities is discussed and a backwards compatible solution proposed.

2.1 Introduction

In the original IEEE 802.11 standard [23], ratified in 1997 and accepted as an ISO standard in 1999, the only available security mechanism was Wired Equivalent Privacy (WEP). During the years that followed, WEP was analyzed by the academic community and wireless hackers, and several vulnerabilities were discovered [24] [25] [26]. This motivated the development of a replacement for WEP, IEEE 802.11i. In 2004, the 802.11i amendment was ratified, with two new and improved security mechanisms. The first one, Temporal Key Integrity Protocol (TKIP), was designed as a transitional solution that would support old hardware. The second, counter mode with cipher-block chaining message authentication code protocol (CCMP), was the long term solution to the security vulnerabilities of WEP. The common denominator for WEP, TKIP and CCMP is that they protect 802.11 data frames. No protection is provided for control frames and management frames.

One issue with the lack of management frame protection is that any station on the wireless network can transmit forged management frames. This tactic can be used by an attacker to make a station (STA) deauthenticate or disassociate from the access point (AP). The following association request from the station gives the attacker the service set identifier (SSID) of the wireless network, thus bypassing SSID cloaking. Furthermore, dictionary attacks against TKIP or CCMP using a password derived preshared key (PSK) require that the attacker observes the initial 4-way handshake, and a successful disassociation attack will result in this 4-way handshake between the wireless station and the AP. Last, but not least, transmitting deauthentication or disassociation frames several times per second is a very efficient Denial of Service attack on the wireless network. Aireplay-ng from the aircrack-ng [27] suite is an example of a freely available tool that implements the deauthentication attack. One countermeasure to these attacks is to provide integrity and replay protection for management frames.

Another issue that has surfaced recently is that several of the new amendments to the 802.11 standard extend the use of management action frames, transmitting potentially sensitive information inside management frames. Examples of such

amendments are 802.11k, 802.11r and 802.11v. To avoid the compromise of sensitive information, management frame confidentiality must be provided.

As a response to the above mentioned issues, Task Group w (TGw) was established in 2005 to develop the 802.11w amendment, Protected Management Frames. The original target date for ratification of this amendment was September 2007, but this was later postponed to December 2009. The design goal for 802.11w was to extend the security mechanisms in 802.11i to provide protection for selected 802.11 management frames. 802.11w is currently in draft status. The newest available draft version is 7.0.

The results presented in this paper are based on IEEE 802.11-2007 [28], which includes the 802.11i amendment [29], and 802.11w draft version 3.0 [30] from September 2007. One additional feature from 802.11w draft version 4.0, protection against SA termination attacks, is also discussed. The analysis of potential DoS vulnerabilities in 802.11 with amendments is based on the observations in [4].

The rest of the paper is divided into eight sections. Section 2 presents the contribution. In Section 3, a short description of related work on 802.11 DoS vulnerabilities is presented. Section 4 contains an analysis of relevant topics from the 802.11 standard with amendments. Section 5 presents theoretical DoS vulnerabilities in 802.11, 802.11i and 802.11w and some general observations on network DoS. Section 6 provides a description of the experiments, analysis and results. The results are discussed in Section 7, and a solution proposed in Section 8. Section 9 contains the conclusion and section 10 contains acknowledgements.

2.2 Contribution

This paper analyzes medium access control (MAC) layer DoS vulnerabilities in 802.11 with the 802.11i and 802.11w amendments. One apology for MAC layer DoS vulnerabilities is that an attacker can use physical jamming of the radio frequencies to perform a DoS attack anyway, which is extremely difficult to prevent. The motivation for preventing DoS attacks against the MAC layer is that such attacks are far more efficient than jamming, so the attacker has to spend less effort, and thus will be more difficult to detect and locate. Furthermore, certain attacks against MAC layer vulnerabilities may cause a deadlock such that a station is not able to recover. A jamming attack, on the other hand, will only disrupt network access for as long as the attacker is transmitting.

The configuration used for the experimental analysis is an extended service set (ESS) with a wireless station communicating with an AP. The term station refers to either a non-AP 802.11 device or an AP.

This paper makes three principal contributions. First, a previously unknown DoS vulnerability in 802.11, equivalent to the disassociation vulnerability, and still present in 802.11w, is presented and analyzed. Second, this new vulnerability is tested experimentally together with the deauthentication attack and another vulnerability discovered by J. Epstein in 2007 [18]. All experiments were carried out using freely available tools and off the shelf hardware. Third, a robust solution to the MAC layer DoS vulnerabilities in 802.11 is proposed. It is possible to introduce this solution

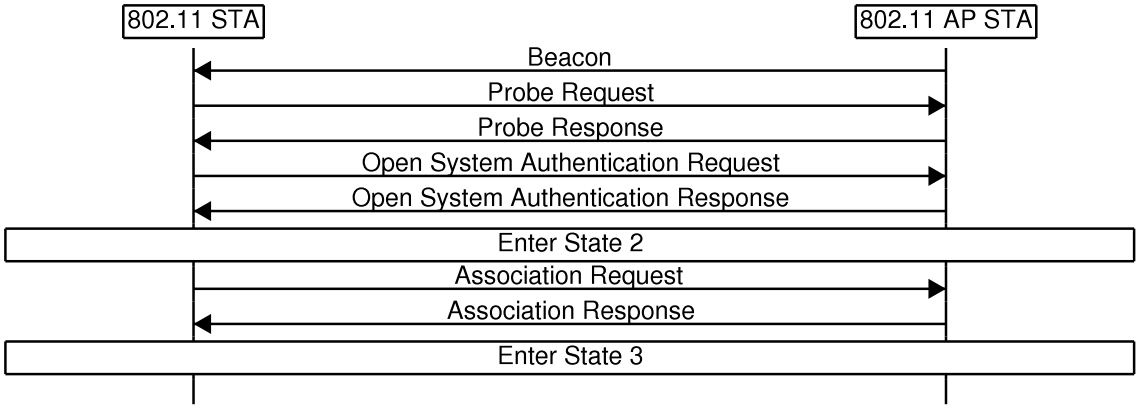

Figure 2.1: 802.11 open system authentication and association.

incrementally, preserving backwards compatibility until all APs and stations are upgraded.

2.3 Related Work

In 2003, Bellardo and Savage demonstrated the feasability and efficiency of the 802.11 deauthentication attack, together with several other DoS attacks against the 802.11 MAC layer [5]. [5] is a useful general reference on DoS attacks against 802.11 networks. In 2007, J. Epstein presented the theoretical SA termination attack [18] and a proposed solution [31] to TGw, which was accepted as part of draft 4.0 of the 802.11w amendment in 2008. The SA termination attack and the proposed solution are analyzed in this paper. The working documents of TGw are available at `https://mentor.ieee.org/802.11/documents`.

2.4 Analysis of the 802.11 Standard

Only the most relevant parts of the 802.11 standard and the 802.11i and 802.11w amendments are presented as background material. The reader is referred to the IEEE standard and draft documents for a comprehensive review.

2.4.1 802.11 Authentication and Association

The original 802.11 standard specifies two types of authentication: shared key and open system. The shared key authentication is optional in WEP, and the open system authentication is a two-message null authentication initiated by the station. After authentication, the station performs an association with the AP. Figure 2.1 shows a successful open system authentication followed by a successful association.

Associations are used to keep track of the stations served by an AP. The 802.11 standard defines two state variables: authentication state and association state. Three of the four possible combinations of these two variables represent the local

Table 2.1: 802.11 States

State 1	Not authenticated	Not associated
State 2	Authenticated	Not associated
State 3	Authenticated	Associated

802.11 station states shown in Table 2.1. Every station maintains a local state for every other station that it communicates with.

802.11 frames are grouped into classes that correspond to the states mentioned above. Frames corresponding to the current state or lower are allowed, thus the allowed frames in State 2 are of Class 1 or 2. If a station receives a Class 2 or 3 frame from a station that is not authenticated, it shall respond with a deauthentication frame. If it receives a Class 3 frame from a station that is authenticated, but not associated, it shall respond with a disassociation frame. Figure 2.2 shows the valid transitions between the local states in 802.11.

Subsection 11.3.1.2 of the 802.11 standard [28] specifies how the destination STA should handle 802.11 authentication requests:

> Upon receipt of an Authentication frame with authentication transaction sequence number equal to 1, the destination STA shall authenticate with the indicated STA using the following procedure:
>
> a) The STA shall execute the authentication mechanism described in 8.2.2.2.
>
> b) If the authentication was successful, the state variable for the indicated STA shall be set to State 2.
>
> c) The STA shall issue an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication.

Note that an open system authentication will always be successful, so an AP that receives an open system authentication request will always enter State 2 (authenticated, but not associated).

2.4.2 802.11i Security Amendments

802.11i introduces a new security framework: The Robust Security Network (RSN). Authentication and key management in an RSN is carried out after the successful completion of 802.11 authentication and association, as illustrated in figure 2.1. However, some of the messages are modified. The beacon frames broadcast by the AP and the probe response contain an RSN information element with the supported security parameters. Cryptographic parameters are negotiated during the association phase by including an RSN information element in the association request from the station. If the security parameters are accepted by the AP, it enters State 3, and authentication is carried out using the Extensible Authentication Protocol (EAP) [32]. EAP encapsulation over Local Area Networks (EAPOL), as specified in

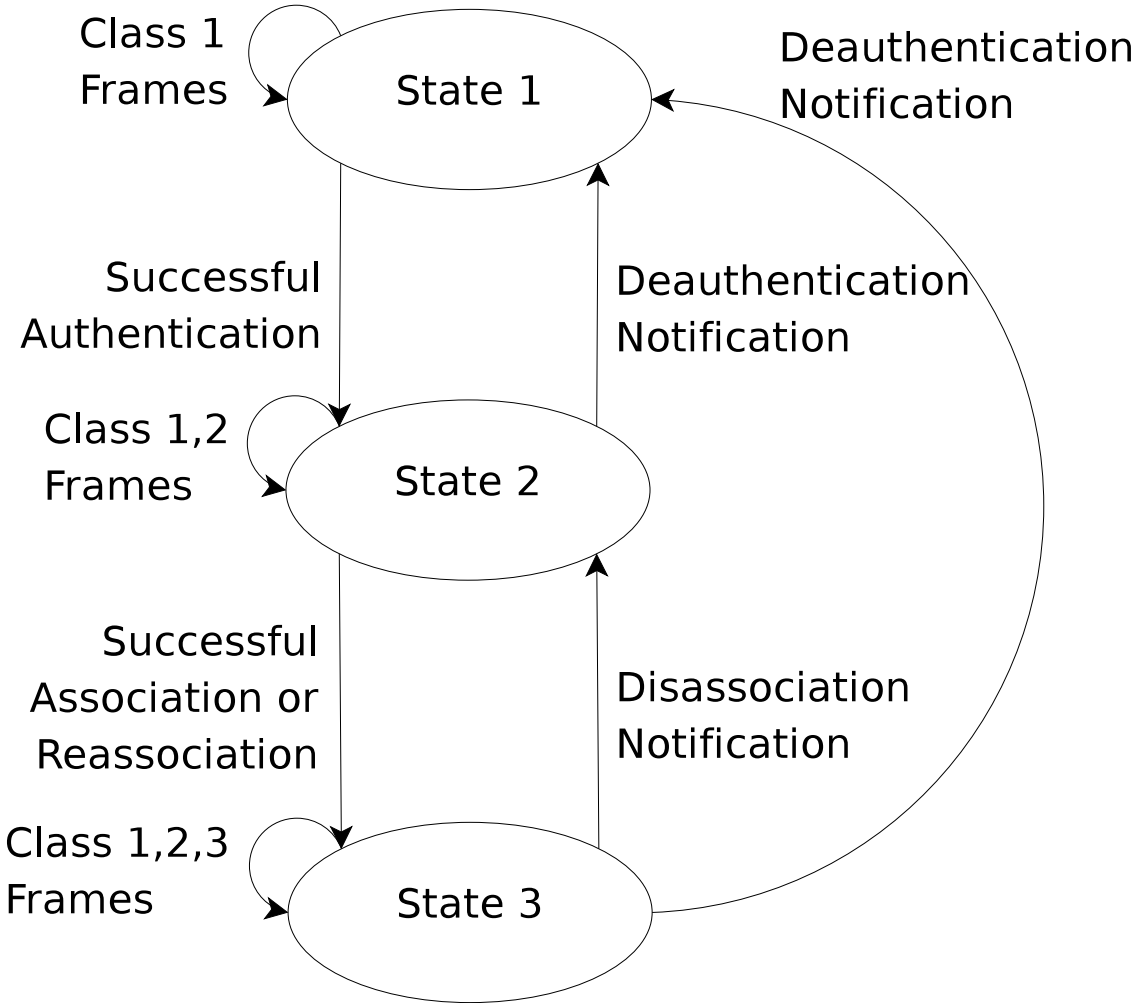

Figure 2.2: 802.11 state transitions. The authentication attack triggers a change from State 3 to State 2 in the AP by transmitting a forged open system authentication request. This transition is not shown in the state diagram, the only transition from State 2 to State 3 is a disassociation notification, but it must be allowed to avoid deadlocks when 802.11w is enabled.

IEEE 802.1X [33], is used to encapsulate the authentication messages in 802.11 data frames. Figure 2.3 shows the authentication and key management in an RSN.

802.11i uses security associations (SAs) to store security policies and cryptographic keys. There are two parts of the SA specifications that are relevant to the vulnerabilities discussed in this paper: SA termination and recovery from lost key state synchronization.

SA termination is triggered when an AP receives or transmits certain management frames. If an AP receives a valid association or reassociation frame from a station, it will delete the pairwise transient key SA (PTKSA), which contains the station's pairwise transient key. The PTKSA is also deleted if the AP sends or receives a deauthentication or disassociation frame.

Loss of key state synchronization can occur if a station reboots and the temporal

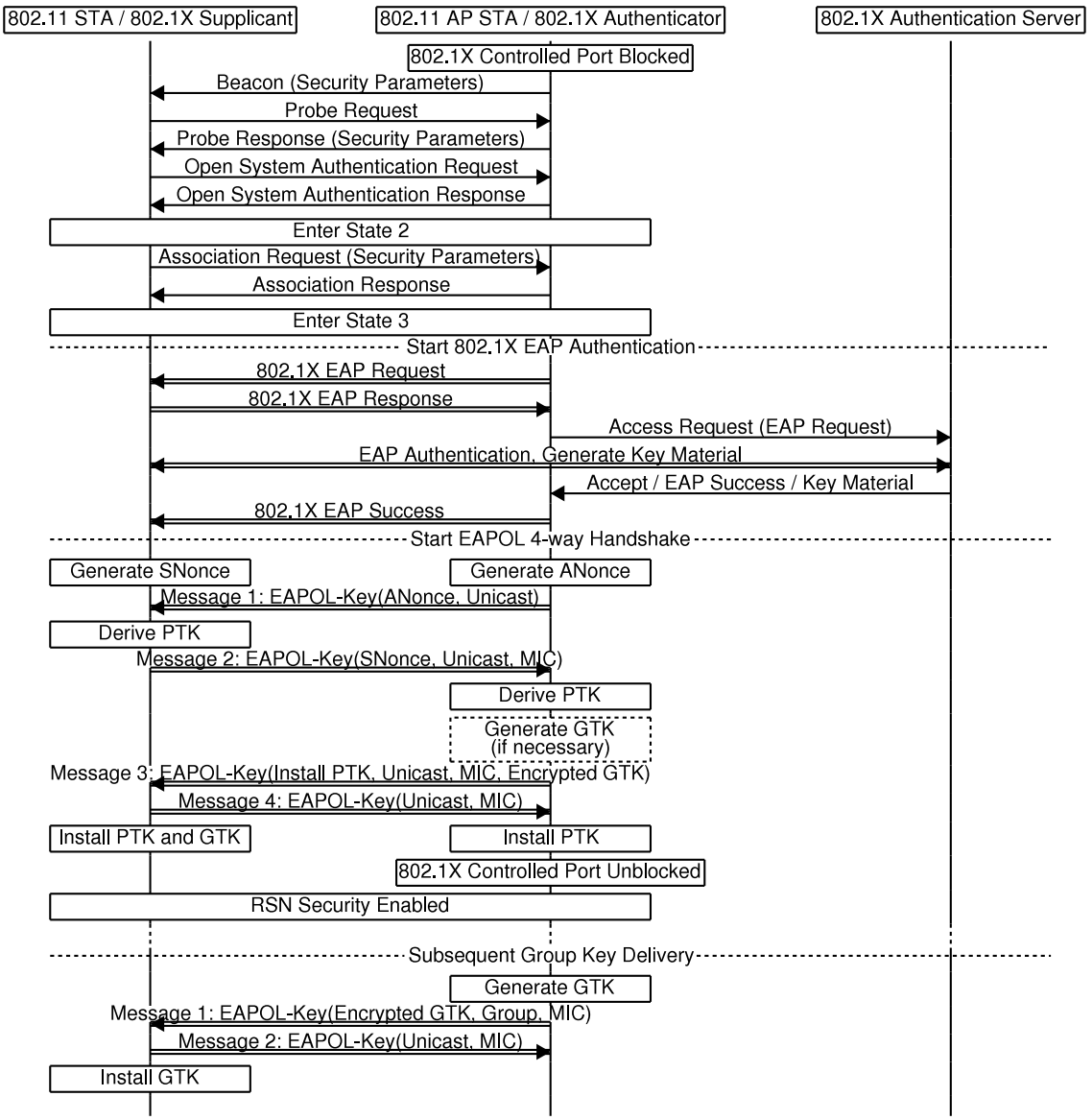

Figure 2.3: 802.11i RSN authentication and key management. Single lines represent management frames, double lines represent data frames. Note that a deauthentication attack will force the station to do the whole procedure over again, starting with the authentication request. If pairwise master key security association (PMKSA) caching is enabled, the 802.1X authentication does not have to be repeated.

keys stored in memory are lost. A station that loses key state synchronization in an ESS shall perform the deauthentication procedure before it sends an authentication request. If the authentication and key management protocol (AKMP) fails between a station and an AP that are associated, both the station and AP shall perform the deauthentication procedure.

2.4.3 802.11w Protected Management Frames

802.11w uses CCMP from 802.11i to provide integrity, confidentiality and sender authenticity for unicast management frames, and Broadcast Integrity Protocol (BIP) to provide integrity for broadcast management frames. In both cases, protection is only provided for management frames of subtype action, deauthentication and disassociation. If protection of management frames is enabled and an unprotected management frame of subtype action, deauthentication or disassociation is received, the frame is silently discarded.

2.5 Vulnerability Analysis

2.5.1 General Observations

Meadows discusses several important principles for protocol design to minimize the vulnerability to DoS attacks [4]. One of the fundamental principles is the following:

> First of all, such a protocol must provide authentication from the very beginning.

802.11 with the 802.11i and 802.11w amendments does not provide this, since the 802.11 authentication and association procedures are carried out, unprotected, before the 802.11i authentication is initiated. All of the messages exchanged prior to the 802.11i authentication can thus be forged by an attacker. Of particular interest are the messages that result in state transitions for the AP: authentication requests, association requests, deauthentication notifications and disassociation notifications. A successful authentication request will make the AP enter State 2. A successful association request will make the AP enter State 3 if it is currently in State 2. Deauthentication and disassociation notifications will make the AP enter State 1 or State 2, respectively. The 802.11w amendment provides integrity protection for deauthentication and disassociation notifications, and in the latest drafts it also provides a mechanism to avoid forged association requests. Authentication requests, however, are not protected. Exploiting unprotected authentication requests to perform a DoS attack against 802.11 with 802.11i and 802.11w is a principal contribution of this paper.

2.5.2 The 802.11 Standard

802.11 deauthentication and disassociation DoS attacks are carried out by forging a deauthentication or disassociation frame. The receiving station will change to

State 1 for a deauthentication or State 2 for a disassociation. The most efficient of these two is the deauthentication attack. If the station is deauthenticated, it has to authenticate and associate to be able to send and receive traffic again. A slightly more efficient approach is to deauthenticate the AP, which resets the AP to State 1. The next data frame from the station will be dropped, the AP will respond with a deauthentication notification, and the station will then authenticate and associate.

2.5.3 802.11i Security Amendments

802.11i significantly "improves" the efficiency of the deauthentication DoS attack. Once a station has been deauthenticated, it must first perform 802.11 authentication and association. Then, if enabled, 802.1X authentication must be carried out. 802.1X authentication is not used with TKIP-PSK and CCMP-PSK, or when PMK caching is enabled and a valid PMKSA exists between the AP and station. Finally, an EAPOL 4-way handshake must be completed to derive the temporal keys. Once the 4-way handshake is completed, the station can send and receive traffic.

The SA termination procedures in 802.11i make an even more efficient DoS attack possible. If an attacker sends a forged association or reassociation frame from the station to the AP, the AP will remain in State 3, but the temporal keys will be deleted. The AP will start the EAPOL 4-way handshake, which will eventually time out, then deauthenticate the station, resulting in the procedure described in the previous paragraph.

2.5.4 802.11w Protected Management Frames

802.11w prevents the deauthentication and disassociation attacks. However, the effect of the SA termination attack is amplified. When the EAPOL 4-way handshake times out, the AP will try to deauthenticate the station. Since the pairwise keys in the AP are deleted, the deauthentication frame will not be protected, and thus discarded by the station. The station will not be able to send or receive any traffic, and is not able to recover, since it discards the deauthentication frames from the AP. An attempt to fix this vulnerability is included in draft version 4.0 and later of 802.11w, where a cryptographically protected SA Query procedure is used to determine whether or not an association or reassociation frame from the station is legitimate. Implementations based on draft 3.0 or earlier, however, are still vulnerable to the SA termination attack.

The SA Query procedure works as follows: if an AP receives an association request from a station with which it has a valid PTKSA, the AP responds that the association request was temporarily rejected. This response tells the station how long it has to wait before it can send another association request. Then, the AP tries to send one or more query messages to the station to check if it has a valid PTKSA. The queries are management action frames protected under the current PTKSA. If a valid response to one of these queries is received, the association request is ignored. If no response is received before the timeout value is reached, the AP will delete the PTKSA. A station that loses key state synchronization will thus have to send an association request, wait until the query procedure times out, then send a

new association request. The number of queries and timeout value are configurable parameters.

Another issue with 802.11w is that the recovery procedure for lost key state synchronization in 802.11i is no longer possible, since a station that loses synchronization will not be able to send a protected deauthentication frame to the AP. To recover, a station has to start 802.11 authentication without first performing a deauthentication, and the AP has to allow this to avoid a deadlock. This can be exploited to enable a new kind of DoS attack against 802.11: The attacker transmits a forged open system authentication frame, which will make the AP enter State 2. The AP still has a valid PTKSA with the station, so once the station transmits a data frame, the AP responds with a protected disassociation frame. The end result is the same as if a disassociation attack had been carried out. This type of attack will from now on be referred to as an "authentication attack".

2.6 Experiments

The goals of the experiments were to verify the feasability of the authentication and SA termination DoS attacks, and to verify that 802.11w protects against the deauthentication attack. To this end, the authentication, SA termination and deauthentication attacks were performed both with 802.11w enabled and disabled, for a total of six experiments. Each attack was performed 100 times to ensure that the results were consistent.

2.6.1 Infrastructure Set-Up

The infrastructure under attack consisted of a Cisco 4402 wireless controller (AIR-WLC4402-25-K9) and a Cisco 1030 access point (AIR-AP-1030). Both the wireless controller and access point were running software version 4.2.61.0 with Cisco Management Frame Protection (MFP) based on an earlier 802.11w revision than draft 3.0. 802.11i CCMP-PSK was used for all the experiments. The wireless controller and AP were configured to reject shared key authentication, and CCMP-PSK was required, which means that association requests without an RSN information element were rejected. The station was a laptop computer with a Cisco Aironet 802.11 a/b/g network adapter (AIR-CB21AG-E-K9), running Windows XP SP2. The station was assigned an IPv4 address through DHCP from the wireless controller. Both the AP and station used 802.11g for the experiments. The attacker was a laptop with a wireless network interface card (NIC) with the Atheros AR2413 chipset, running Linux 2.6.22 with the madwifi-ng drivers, and aircrack-ng [27] version 0.9.1 as the attack software. In particular, airmon-ng was used to enable RFMON (monitor) mode, aireplay-ng and airtun-ng were used to inject frames, and airodump-ng to capture traffic. The same wireless network interface was used for frame injection and traffic capture, and the experiments were conducted in a typical office environment, with no shielding from other wireless stations and APs nearby. The only legitimate traffic on the wireless network was an Internet Control Message Protocol (ICMP)

ECHO request from the station to a server on the wired LAN every second, and an ICMP ECHO request from the server to the station every second, along with the ICMP ECHO responses. The ping commands in Windows XP and Linux were used to generate traffic from the station and server, respectively.

2.6.2 Attacks

The attacks were carried out by transmitting a single management frame of subtype authentication request, association request or deauthentication. To ensure that only one frame was transmitted, an attack tool was used to generate the frame, which was captured using airodump-ng. The single frame was then saved to a file and replayed using airtun-ng.

First, the aireplay-ng tool was used to generate an authentication request frame, with the two-byte authentication algorithm field set to "Open System" ($0x0000$). Then, an association request frame containing an RSN information element with CCMP support, which would be accepted by the AP, was obtained by running an authentication attack and recording the subsequent association request transmitted by the station. It is also possible for an attacker to construct a valid association frame from the information contained in the beacon frames broadcast from the AP. The authentication request and association request frames were constructed with the station MAC address as source and the AP MAC address as destination. Last, the aireplay-ng tool was used to construct a deauthentication frame with the AP MAC address as source and the station MAC address as destination.

Once the attack frames were generated, the experiments were performed by transmitting an attack frame once, then waiting for the station to regain connectivity. Once the station was back on-line, a new attack was launched, and this was repeated until a total of 100 attacks of each type had been carried out. All 802.11 frames to and from the AP were recorded for analysis.

2.6.3 Observations

Several significant results were observed while conducting the experiments.

First, as expected, the deauthentication attack did not work when MFP was enabled, but did work as expected when disabled.

Second, the authentication attack worked, both with and without MFP enabled. The station lost its network connection and had to reconnect.

Third, the valid association attack resulted in a permanent DoS when MFP was enabled. After excessive timeouts, the station interface was automatically assigned a link-local IPv4 address (169.254/16 prefix), and manual intervention was needed to get it back on-line.

2.6.4 Results

Once the experiments were completed, Wireshark [34] version 0.99.6 was used to analyze the results.

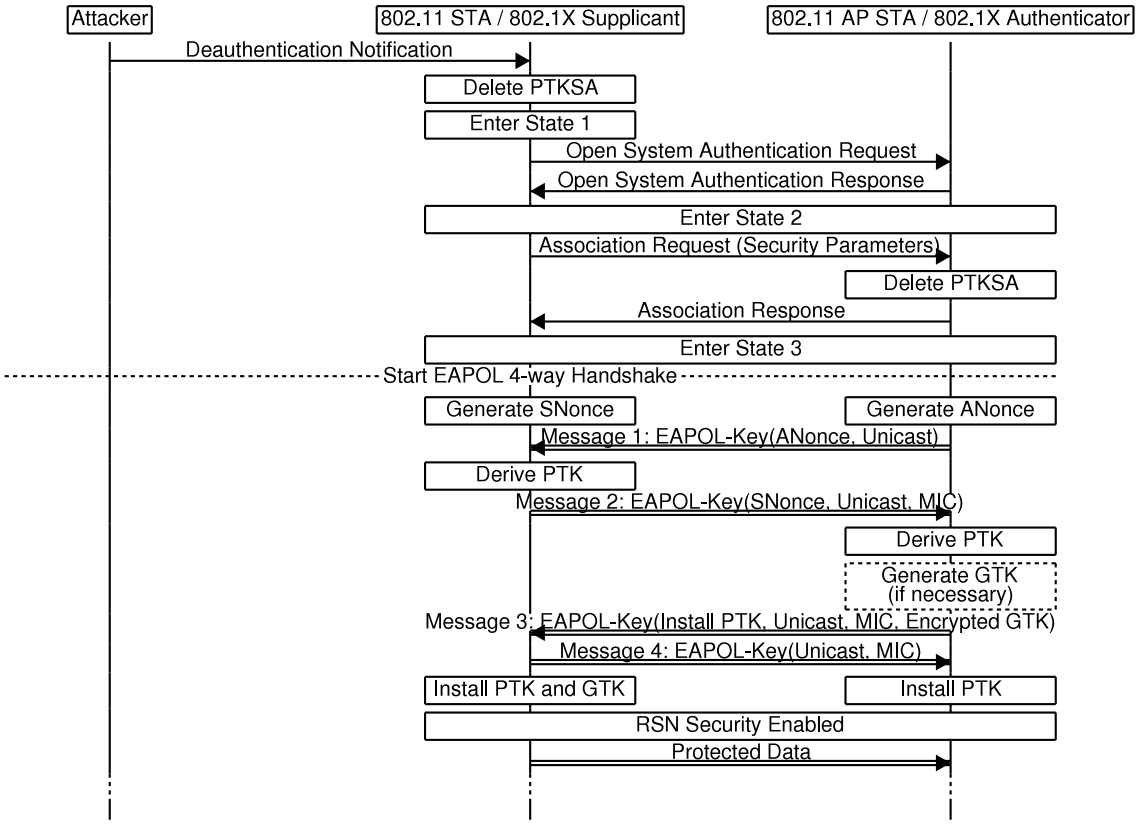

Figure 2.4: Expected and observed results for the deauthentication attack with MFP disabled. The attack had no effect when MFP was enabled.

Deauthentication Attack

The deauthentication attack worked as expected. Figure 2.4 shows the expected and observed results when MFP was disabled. With MFP enabled, the attack had no effect, since the deauthentication notification was ignored by the station.

Authentication Attack

The results of the authentication attack were slightly different from the expected results. Figure 2.5 shows how the attack would work on an implementation that conforms to the standard.

The only difference between the expected and observed results were that the AP responded with a deauthentication notification when it should have used a disassociation notification. Figure 2.6 shows the observed results with MFP enabled. With MFP disabled, the only difference was that the deauthentication notification was not protected. The reason code in the deauthentication notification frame was "Class 3 frame received from nonassociated station (0x0007)", which confirms that the AP was in State 1 or 2 immediately after the attack.

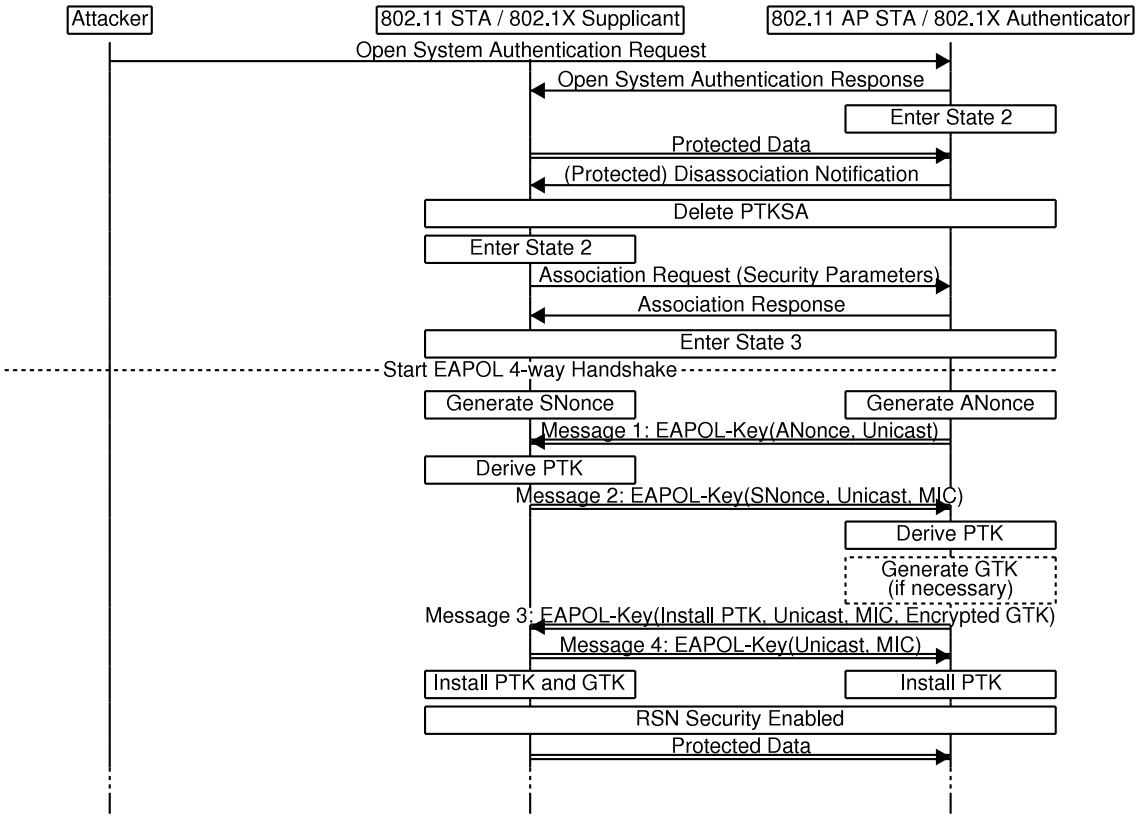

Figure 2.5: Expected results for the authentication attack. The only difference between MFP enabled and disabled was that in the former case the disassociation notification was protected.

SA Termination Attack

The results of the SA termination attacks were also as expected. Figure 2.7 shows the expected and observed results with MFP disabled. One interesting observation is that this attack is more efficient than any other known MAC layer DoS attack against 802.11 when RSN is enabled. In the experiment, the AP sent the first message of the EAPOL 4-way handshake, then waited for one second before retrying. This was repeated three times before the AKMP failed. The SA termination attach thus added three more seconds of downtime compared to the deauthentication attack.

Figure 2.8 shows the expected and observed results with MFP enabled. The station did not accept unprotected deauthentication notifications from the AP. The end result was a deadlock, with manual intervention required to get the station reconnected.

2.7 Discussion

The results from the theoretical analysis and the experiments in the previous sections show that a network using 802.11w is vulnerable to the authentication attack. This attack has the same efficiency and feasability as a disassociation attack. 802.11w thus

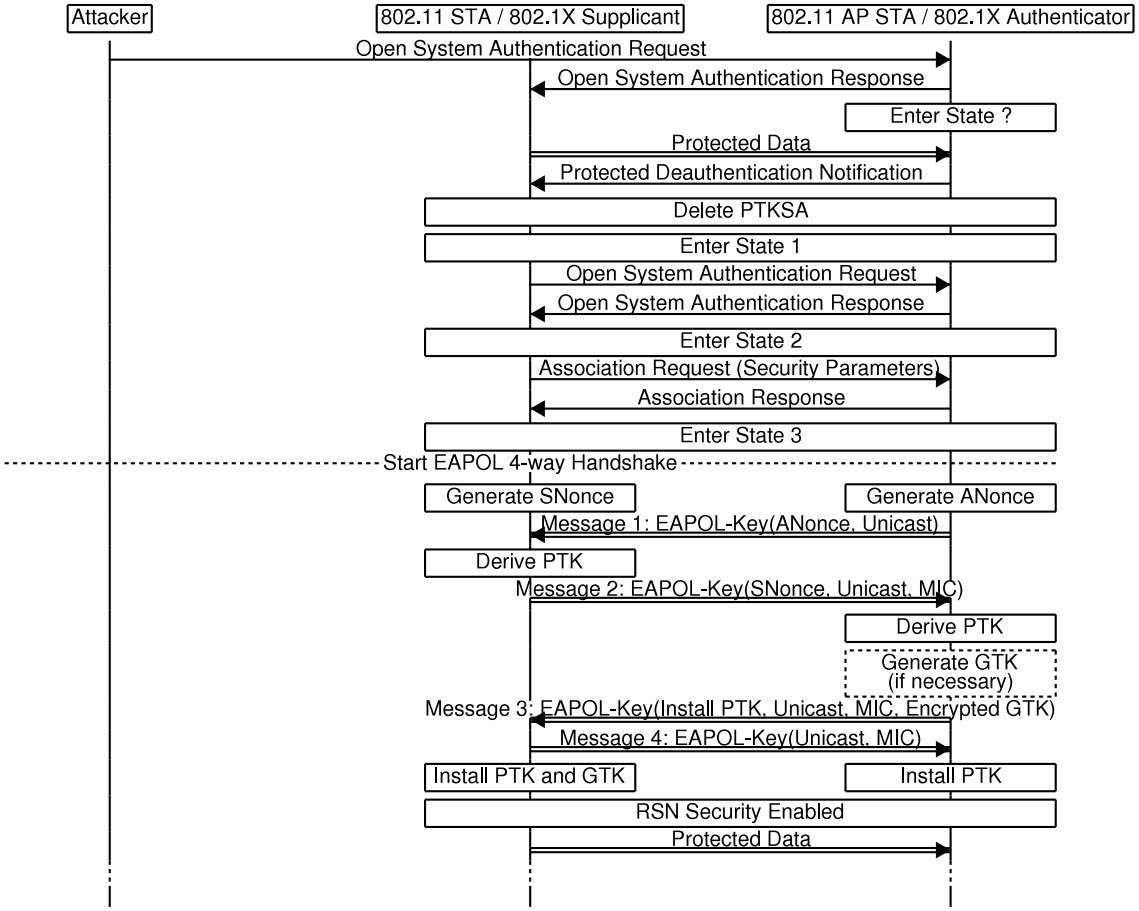

Figure 2.6: Observed results for the authentication attack with MFP enabled. The AP responds with a deauthentication notification when it should have used a disassociation notification.

fails to protect against all DoS attacks that are equivalent to the deauthentication and disassociation attacks.

Introducing protected deauthentication and disassociation frames in 802.11w leads to a deadlock vulnerability. If the PTKSA in the AP is deleted while the station still has a valid PTKSA, then the station is not able to recover. This is the result of the SA termination attack. The proposed solution to this vulnerability by TGw is the SA Query procedure. This procedure has a weakness: an attacker who is able to delete messages or perform radio frequency (RF) jamming attacks will still be able to create a deadlock by sending an association request, then deleting the SA queries or perform RF jamming until the SA Query procedure times out. Message deletion in 802.11 networks is possible in the following way: the attacker listens for messages, then switches on the transmitter while the message is in transit to create a collision. Immediately after the collision, the attacker sends a MAC layer acknowledgment (ACK) to the sender. The sender thus assumes that the message was received, and no retransmission occurs. RF jamming attacks are even easier to perform. Since the association response from the AP contains the timeout value, the attacker knows exactly how long the jamming attack must last to result in a

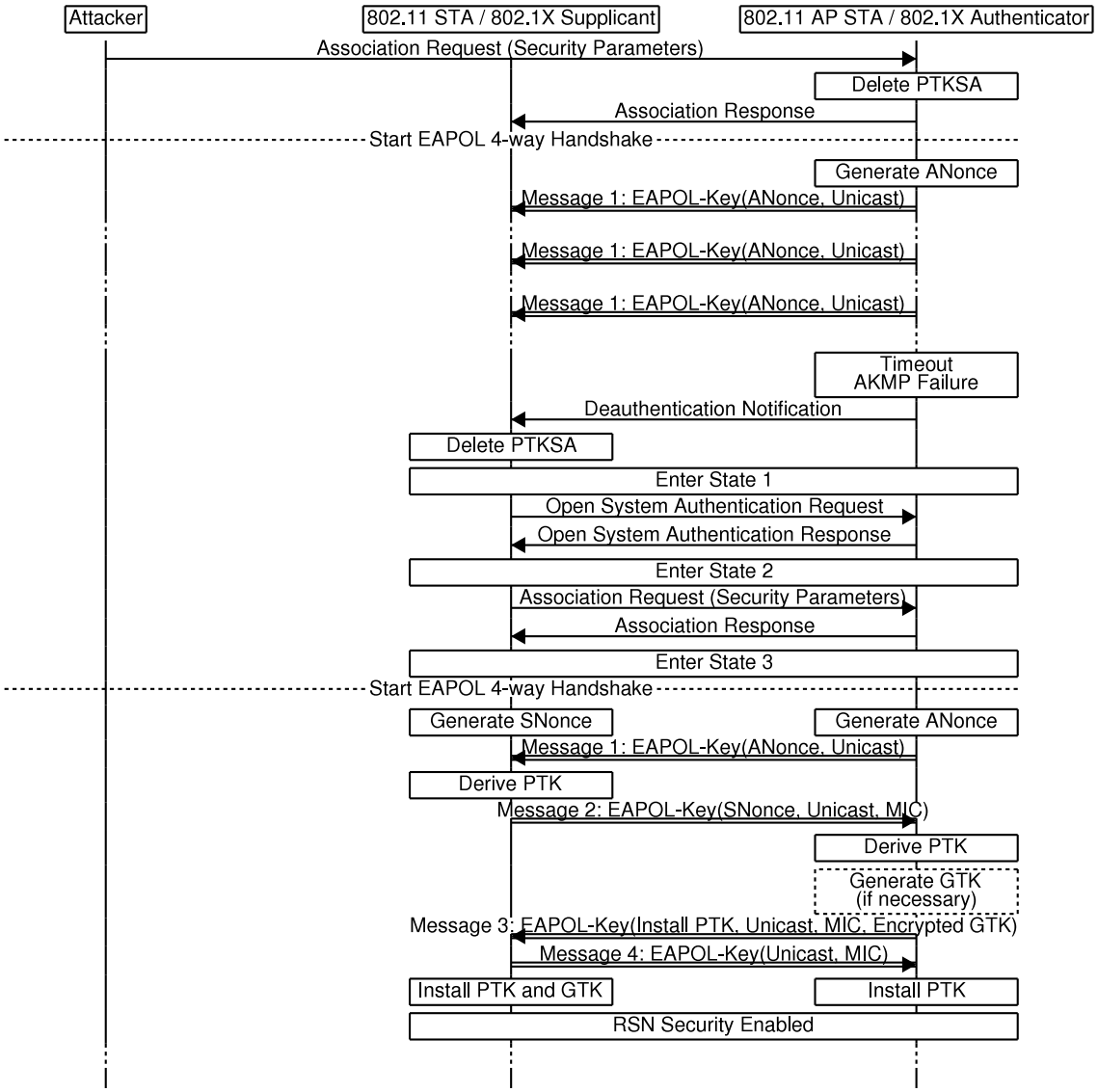

Figure 2.7: Expected and observed results for the SA termination attack with MFP disabled. The failed 4-way handshake adds three seconds of downtime compared to a deauthentication attack.

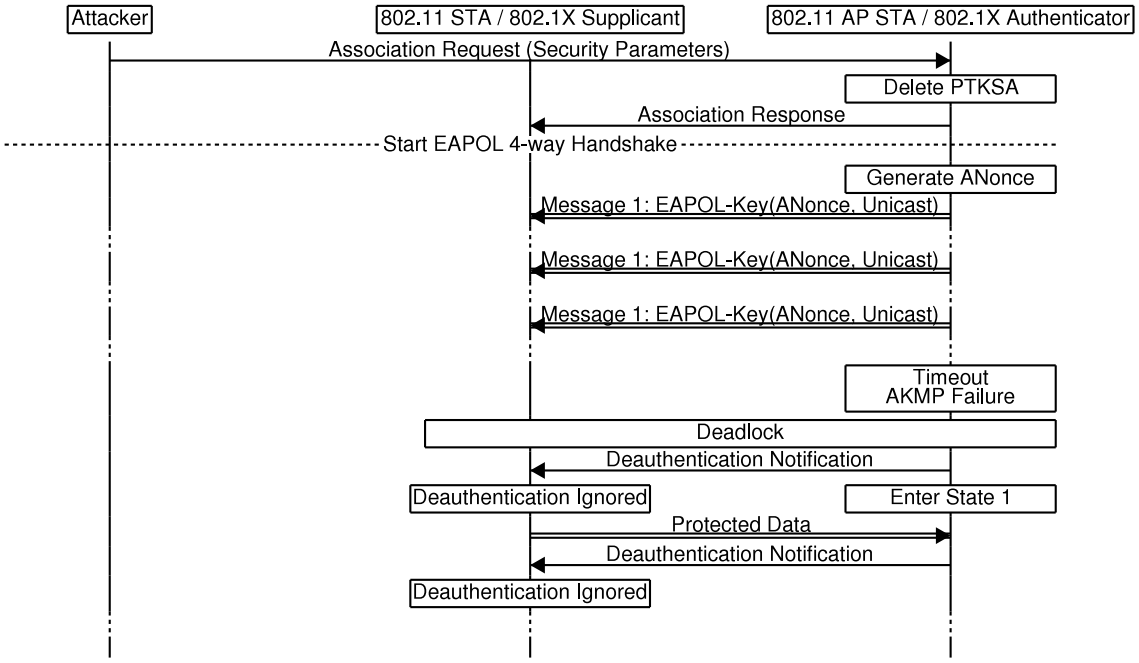

Figure 2.8: Expected and observed results for the SA termination attack with MFP enabled. The result is a deadlock.

deadlock. The 802.11w drafts suggest a timeout value of around one second. An attacker can thus spend one second of RF jamming to permanently disconnect the station.

2.8 Proposal for a Robust Solution

The root cause of the DoS vulnerabilities in 802.11, both the previously known ones and the new vulnerability presented in this paper, is that 802.11 with amendments does not adhere to the first principle from [4]. The proposed solution adheres to this principle: To provide authentication from the very beginning. The challenge is how to do it, given the existing 802.11 standard with amendments. The creators of WEP did one thing right, their shared key authentication was performed as early as possible. This authentication, as well as the 802.11 open system authentication, is carried out using management frames of subtype authentication. Such frames have an important property, they contain an "Authentication Algorithm Number" field with a length of two bytes. Currently, only the values "0" (open system) and "1" (shared key) are used. This means that it is possible to add identifiers for new authentication methods.

Figure 2.9 shows the proposed authentication and key management procedure. The new authentication frame specification is the following: Add a new authentication algorithm number, "2", for RSN authentication. Add an RSN information element (security parameters) to the authentication frame. This enables the station to specify the authentication method and security parameters to be used in the authentication request.

The remaining issue is how to encapsulate the EAPOL messages used for authentication. This is solved by adding a new management frame subtype of Class 1, "authentication and key management". To remove all of the DoS vulnerabilities described in this paper, the 802.11i EAPOL authentication and key exchange messages are encapsulated in the authentication and key management frames, rather than in data frames. 802.11w should then be amended to also provide protection for authentication and key management, association request and association response frames. Note that for backwards compatibility, the use of data frames to transport EAPOL messages must still be supported as defined in 802.11i.

Finally, to avoid deadlocks, the PTKSA should not be terminated after a successful association, disassociation or deauthentication, but rather be replaced with a new PTKSA after a successful 4-way handshake. If the protected association procedure fails, both the station and AP should perform the deauthentication procedure.

The construction outlined above can be backwards compatible with 802.11 with amendments, as noted. However, as long as backwards compatibility is preserved, the network will still be vulnerable to the authentication attack described in subsection 2.5.4. A transitional workaround for this is that the AP maintains a list of stations that have been successfully authenticated using the new authentication method, and that authentication requests for open system or shared key authentication for these stations are ignored. If backwards compatibility is discarded, this is not an issue, since an attacker will not be able to successfully authenticate.

2.9 Conclusions

All of the attacks presented in this paper were carried out using off the shelf hardware and freely available software. No software or hardware modification was necessary, so any person with access to a laptop computer and an Internet connection should be able to replicate these experiments or carry out actual DoS attacks.

Although the SA termination vulnerability from forged association frames has been addressed in recent draft versions of 802.11w, implementations of early drafts are still vulnerable. Until these have been updated, a network with 802.11w enabled is *more vulnerable* to DoS than a network without. Since the only purpose of 802.11w at the moment is to protect against DoS, a sound recommendation would be to disable it until a solution for this vulnerability is provided.

The SA Query procedure proposed as a solution to the SA termination vulnerability does not protect against an attacker who is able to delete messages or perform RF jamming attacks. Due to the severity of this vulnerability, the author strongly recommends that a more robust solution, such as the one proposed in section 2.8, is adopted.

The 802.11w drafts do not, as far as the author is aware of, address the authentication attack of subsection 2.5.4. If protection against all DoS attacks with efficiency and feasability equivalent to the disassociation attack is a goal of TGw, the proposed solution from this paper should be included in the 802.11w amendment.

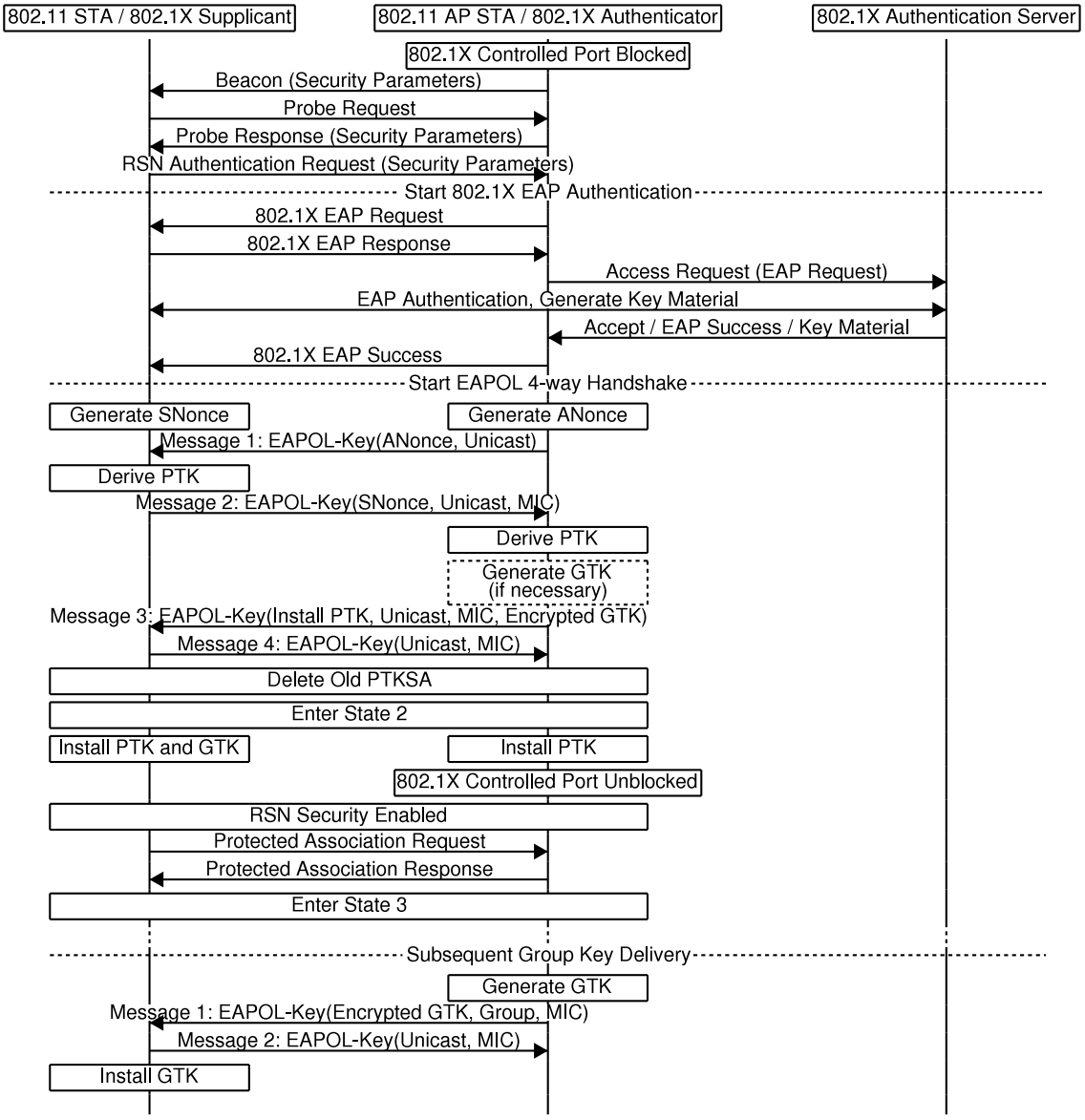

Figure 2.9: The proposed solution for RSN authentication and key management. The authentication procedure is initiated when the station transmits an authentication frame with authentication algorithm number equal to 2 and a valid RSN information element. Management frames of subtype "authentication and key management" are used to encapsulate the 802.1X authentication messages, the 4-way handshake and the group key handshake. Note that the association request and response are protected.

2.10 Acknowledgments

The author would like to thank Stig F. Mjølsnes for valuable help and advice, Jing Xie for suggesting how to design a robust solution, and the anonymous reviewers of this paper for their helpful comments.

Chapter 3

Paper B

Published in:

Martin Eian

"A Practical Cryptographic Denial of Service Attack Against 802.11i TKIP and CCMP"

Proceedings of the Ninth International Conference on Cryptology And Network Security (CANS 2010)

Lecture Notes in Computer Science (LNCS), vol. 6467
Springer Verlag, 2010
ISBN 978-3-642-17618-0

Abstract

This paper proposes a highly efficient cryptographic denial of service attack against 802.11 networks using 802.11i TKIP and CCMP. The attacker captures one frame, then modifies and transmits it twice to disrupt network access for 60 seconds. We analyze, implement and experimentally validate the attack. We also propose a robust solution and recommendations for network administrators.

3.1 Introduction

IEEE 802.11 is a standard for wireless local area networks[1] [28]. The 802.11i amendment to the standard specifies the robust security network (RSN) [29]. An RSN supports two security mechanisms, the temporal key integrity protocol (TKIP) and counter mode with cipher block chaining message authentication code protocol (CCMP).

TKIP was designed to be backward compatible with existing hardware, which put computational constraints on the message integrity code (MIC) algorithm. The TKIP MIC is vulnerable to attacks due to these constraints. Countermeasures were thus introduced to detect and respond to attacks. If two TKIP MIC failures are detected within 60 seconds, all security associations using TKIP are terminated and the negotiation of new security associations using TKIP is disabled for 60 seconds.

The intended long term security mechanism for 802.11 networks, CCMP, has strong confidentiality and integrity protection. CCMP does not use countermeasures to compensate for vulnerabilities. The most common default configuration for 802.11 access points (APs) using 802.11i is to support both TKIP and CCMP. This provides backward compatibility, as well as a stronger security mechanism for clients that support it.

802.11 has been extensively used during the last decade in computers, mobile phones, wireless security cameras and vehicular communication systems. 802.11 networks are thus attractive targets for adversaries that seek to disrupt communications through the use of denial of service (DoS) attacks. An attacker could use physical layer jamming to disrupt a wireless network. In a typical jamming attack the attacker transmits continuously. A distributed intrusion detection system can locate the attacker by measuring the received signal strength on multiple sensors. More sophisticated and efficient attacks target the 802.11 medium access control (MAC) layer. A common MAC layer attack is the deauthentication attack from Bellardo and Savage [5]. Transmitting a deauthentication frame, which takes less than 100 microseconds, disrupts a network using 802.11i for approximately 1 second [35]. The deauthentication attack is far more efficient than physical layer jamming. In general, the higher the efficiency of the attack, the more difficult it is to locate the attacker. Vulnerabilities that can be exploited by highly efficient DoS attacks should be found and amended.

[1] In this paper, "network" is a synonym for a "basic service set" (BSS) in 802.11.

The motivation of this work is to make 802.11 more resilient to DoS attacks by finding and amending the abovementioned vulnerabilities. This paper makes five principal contributions. First, we analyze the 802.11 standard and discover a highly efficient cryptographic DoS attack. Second, we show that the attack also works against clients using CCMP as the pairwise cipher in networks that support both TKIP and CCMP. Third, we demonstrate that the attack works even if 802.11e quality of service (QoS) support is disabled in the AP. Fourth, we implement the attack and experimentally validate the analytical results. Fifth, we propose a robust solution to the vulnerability and temporary measures to limit the exposure to the vulnerability.

A more general lesson from this work is the connection between the cryptographic protocol design and network availability. Information and network security was traditionally categorized as confidentiality, integrity and availability. In the case of TKIP, availability was intentionally put at risk to improve the integrity of the protocol. Later changes to the protocol resulted in a severe DoS vulnerability, and design flaws even put clients using newer security mechanisms at risk. The use of formal methods, models and tools to analyze the confidentiality and integrity properties of cryptographic protocols is a well developed field of research, but this is not the case for availability. The development of formal methods, models and tools for the analysis of availability in cryptographic protocols might help future protocol designers construct more robust protocols.

The rest of this paper is structured as follows. Section 3.2 reviews related work. In Section 3.3, we present relevant parts of the 802.11 standard and a vulnerability analysis. Section 3.4 provides the attack implementation. Section 3.5 presents the experimental setup, and Section 3.6 contains the results. In Section 3.7, we discuss the results, propose a solution, and provide recommendations for wireless network administrators. Section 3.8 concludes the work.

3.2 Related Work

Researchers have discovered several DoS vulnerabilities in the 802.11 standard. An early paper on this topic by Bellardo and Savage demonstrated that DoS attacks were practical [5]. In the years after the publication of this paper, such attacks have become much easier to carry out due to readily available software such as the aircrack-ng tool suite [27] and the driver support for 802.11 monitor mode and frame injection.

One of the most widely implemented DoS attacks against 802.11 is the deauthentication attack [5], which disconnects a client[2] by transmitting one deauthentication frame. Figure 3.4 in Appendix 3.A illustrates the attack. When 802.11i is used, eight frames must be exchanged between the access point (AP) and the client before it is reconnected. Aime et al. performed measurements of the efficiency of the deauthentication attack, and concluded that transmitting one frame per second was sufficient to completely block the wireless channel in a network using 802.11i [35].

[2]In this paper, "client" is a synonym for a "non-AP station" (STA) in 802.11.

Smith mentioned that there are a number of challenges associated with deliberately invoking the TKIP countermeasures [11, Ch. 6]. He concluded that other DoS attacks against 802.11 were likely easier to mount.

Glass and Muthukkumarasamy published experimental results of a DoS attack in 2007 [36]. The TKIP countermeasures were invoked using a man-in-the-middle technique. They showed that it is possible to mount such an attack in a laboratory environment, but difficult to consistently establish the attacker as a man-in-the-middle between the client and AP. The reason for this difficulty is that the attacker has to compete with the legitimate AP. Since 802.11 uses a wireless broadcast medium, the client will receive messages from both the attacker and the AP, and might choose to connect to the AP rather than the attacker.

Beck and Tews published the first partial key recovery attack against TKIP in 2009 [37]. One of their key observations was that the QoS mechanisms introduced in 802.11e [38] made replay attacks against TKIP possible.

Halvorsen et al. proposed that the attack from Beck and Tews could be used as a cryptographic DoS attack [39] . The attacker has to transmit 129 frames on average to cause the network to shut down for 60 seconds. This attack is less efficient than the deauthentication attack, since the attacker has to transmit more than two frames to cause one second of disruption. The authors assumed that the 802.11e QoS features had to be enabled in the AP for the attack to work.

Könings et al. published two new DoS attacks against 802.11 in 2009 [40]. The attacks exploit the channel switch and channel assessment mechanisms of 802.11h [41]. Their paper also provides an overview and classification of previous DoS attacks against 802.11. With regards to DoS attacks invoking TKIP countermeasures, they only mention the paper by Glass and Muthukkumarasamy [36].

3.3 Vulnerability Analysis

Some background material from 802.11 is required to analyze the TKIP DoS vulnerability. Only the most relevant parts are covered in this paper, see 802.11-2007 [28] for more details.

3.3.1 TKIP and 802.11e

TKIP provides confidentiality and integrity for 802.11 networks by the use of the stream cipher RC4 and the message integrity code (MIC) Michael. The input values to the MIC are the plaintext data, destination address, source address and QoS priority. TKIP generates a new RC4 key for each frame, using a key mixing function. The input values to the key mixing function are the temporal key currently in use, the transmitter's address and the TKIP sequence counter (TSC), a 48-bit monotonically increasing counter. To construct a frame, the MIC is appended to the data, then an integrity check value (ICV) is computed over the data and MIC. TKIP uses a 32-bit cyclic redundancy check (CRC-32) to compute the ICV. Finally, the data, MIC and ICV are encrypted by computing a bitwise XOR with the key stream generated by RC4. Figure 3.5 in Appendix 3.B illustrates the structure of a TKIP frame.

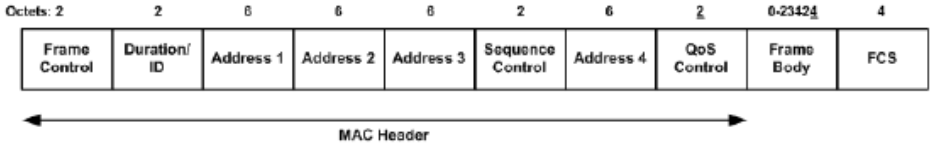

Figure 7-1—MAC frame format

Figure 3.1: The 802.11 MAC frame [28]

The TSC is used to prevent replay attacks. If a frame is received with a TSC value that is equal to or less than the previous value seen, then this frame is discarded. This posed a problem when the QoS mechanisms in 802.11e were introduced. With 802.11e, frames may be transmitted out of order due to different priorities. For example, a frame carrying voice traffic may be transmitted before a frame carrying data from a file transfer, even though the voice frame has a higher TSC. To avoid legitimate frames being dropped, 802.11e introduced a separate TSC for each QoS priority at the receiver. The QoS priority is an integer value stored in the QoS Control field of the medium access control (MAC) header in 802.11, as illustrated in Figure 3.1. 802.11e defines 8 priority classes (0-7).

The wireless multimedia (WMM) specification from the Wi-Fi Alliance, based on 802.11e, defines 4 priority classes (0-3). The QoS Control field is only present in QoS frames. To determine if a frame has a QoS Control field, the receiver inspects the frame type and subtype in the Frame Control field of the MAC header.

The MIC used in TKIP is vulnerable to forgery attacks with a complexity of $O(2^{20})$ [42]. To compensate for this, TKIP uses countermeasures to detect and respond to attacks. All MIC failures at the AP and clients are recorded. If a client experiences a MIC failure, it sends an integrity protected failure report to the AP. If two or more MIC failures or failure reports are observed within 60 seconds, countermeasures are invoked. The countermeasures are to terminate all security associations using TKIP, and to refuse any new security associations using TKIP for 60 seconds. An attacker is thus limited to one MIC forgery attempt per minute.

The designers of TKIP tried to make it difficult to deliberately invoke the countermeasures. The TSC and ICV are checked before the MIC. If either fail, then the frame is discarded and the MIC is not checked. If an attacker changes the TSC, then the encryption key of the RC4 cipher is also changed, so the encrypted ICV would decrypt incorrectly. However, the changes made by 802.11e makes it possible to perform countermeasures based DoS attacks against TKIP. The QoS priority is one of the input values to the MIC, but not to the key mixing function or ICV. If the QoS priority is changed, then the MIC is invalid, but the ICV remains valid. The TSC of the new frame will be checked against the TSC for the new priority. Since a transmitter uses a single, monotonically increasing TSC counter, it is highly probable that the receiver will accept the TSC of a frame that has its QoS priority modified before it is retransmitted. To perform the DoS attack, the attacker captures traffic. When a QoS TKIP frame is observed, the attacker modifies the QoS priority and retransmits the frame twice. The receiver then invokes countermeasures, resulting in

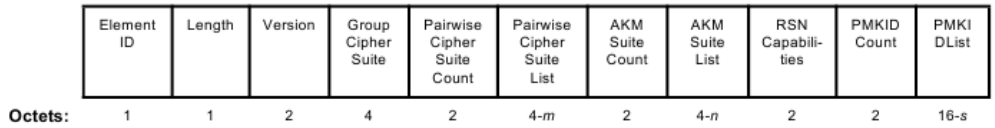

Figure 7-72—RSN information element format

Figure 3.2: The RSN information element [28]

at least 60 seconds of downtime.

3.3.2 TKIP and CCMP

As mentioned in the introduction, the most common AP configuration is to allow both TKIP and CCMP. 802.11i specifies two temporal security associations between an AP and a client. The pairwise transient key security association (PTKSA) protects unicast traffic using a pairwise transient key (PTK). The group transient key security association (GTKSA) protects broadcast and multicast traffic from the AP to clients using a group transient key (GTK). When a client transmits a broadcast or multicast frame, it is protected by the PTK. The frame is decrypted by the AP, encrypted with the GTK, and transmitted to the wireless network. According to Section 9.2.7 of 802.11, clients should discard broadcast and multicast frames that have their own address as the source address [28].

The RSN information element (IE), present in frames of subtype beacon, probe response, association request and reassociation request, contains information about the security mechanisms supported by the transmitter. Figure 3.2 illustrates the RSN IE. The RSN IE supports multiple pairwise cipher suites, but only one group cipher suite. A network that supports both TKIP and CCMP has to use TKIP as the group cipher suite for all clients.

Knowing that CCMP clients use TKIP as the group cipher suite in networks that support both TKIP and CCMP, one might ask what happens if MIC failures occur on broadcast or multicast frames. Section 8.3.2.4 of 802.11 provides the answer [28]:

> The number of MIC failures is accrued independent of the particular key context. Any single MIC failure, whether detected by the Supplicant or the Authenticator and whether resulting from a group MIC key failure or a pairwise MIC key failure, shall be treated as cause for a MIC failure event.
>
> [...]
>
> If less than 60 s have passed since the most recent previous MIC failure, delete the PTKSA and GTKSA. Deauthenticate from the AP and wait for 60 s before (re)establishing a TKIP association with the same AP. A TKIP association is any IEEE 802.11 association that uses TKIP for its pairwise or group cipher suite.

To make the DoS attack work against clients using CCMP, the attacker waits for a broadcast or multicast frame from the AP. The attack is then carried out as described in Subsection 3.3.1. This attack works even if none of the clients use TKIP as the pairwise cipher suite. A side effect of such an attack is that the AP also invokes countermeasures due to the MIC failure reports received from the clients.

One might ask what happens to new security associations using CCMP as the pairwise cipher when the AP invokes countermeasures. Section 8.3.2.4.1 of 802.11 specifies the AP behavior as follows [28]:

> If less than 60 s have passed since the most recent previous MIC failure, the Authenticator shall deauthenticate and delet all PTKSAs for all STAs using TKIP. If the current GTKSA uses TKIP, that GTKSA shall be discarded, and a new GTKSA constructed, but not used for 60 s. The Authenticator shall refuse the construction of new PTKSAs using TKIP as one or more of the ciphers for 60 s. At the end of this period, the MIC failure counter and timer shall be reset, and creation of PTKSAs accepted as usual.

The statement "PTKSAs using TKIP as one or more of the ciphers" contradicts the definition of a PTKSA, which contains only one cipher. If the term "PTKSA" in this context is interpreted to mean "SA", then the statement is consistent with the rest of the standard. Such an interpretation implies that clients using CCMP as the pairwise cipher and TKIP as the group cipher will not be allowed to connect to the AP while countermeasures are in effect. As will be shown in Section 3.6, the experimental results support this interpretation.

3.3.3 Networks Without 802.11e QoS Support

Since the attack relies on 802.11e QoS support, one might ask what happens if QoS support is disabled in the AP. 802.11 is vague on this point, but Section 6.1.1.2 provides a partial answer [28]:

> At QoS STAs associated in a QoS BSS, MSDUs with a priority of Contention are considered equivalent to MSDUs with TID 0, and those with a priority of ContentionFree are delivered using the contention-free delivery if a point coordinator (PC) is present in the AP. If a PC is not present, MSDUs with a priority of ContentionFree shall be delivered using an UP of 0. At STAs associated in a non-QoS BSS, all MSDUs with an integer priority are considered equivalent to MSDUs with a priority of Contention.

The last sentence implies that clients should accept QoS frames even if associated in a network that does not support QoS. The key word in this sentence is "equivalent", which is open to interpretation. If it means that the integer priority is used as input to the TKIP MIC, then the attack works even if QoS is disabled in the AP, as long as the priority is not equal to 0. As will be shown in Section 3.6, the experimental results support this interpretation. To convert a regular data frame to a QoS frame,

an attacker has to flip one bit in the Frame Control field and insert a two byte long QoS Control field.

3.3.4 Analysis Summary

Invoking the TKIP countermeasures is easy due to the modifications in 802.11e. An attacker captures a broadcast or multicast frame from the AP, then modifies the QoS priority and retransmits the frame twice. Figure 3.3 in Appendix 3.A illustrates the attack. The attacker does not need to prevent the reception of the original frame at the clients. The attack is a simple retransmission of a modified frame, and does not use a man-in-the-middle technique. Since the frame is a broadcast frame, all clients except the one that transmitted the frame will invoke countermeasures. The AP will also invoke countermeasures due to the MIC failure reports from the clients. If the frame is not a QoS frame, the attacker flips one bit in the MAC header and inserts the QoS Control field. The attack also works against clients that use CCMP as the pairwise cipher. Furthermore, the attack only relies on QoS support in the clients. Disabling QoS support in the AP does not prevent the DoS attack.

3.4 Implementation

The aircrack-ng [27] tool suite was used as a framework for the vulnerability assessment tool implementation. The implementation depends on a wireless network interface card with driver support for 802.11 monitor mode and frame injection. Network interface cards with the Atheros AR2413 and AR5001X+ chipsets were used as the attacker in the experiments. The driver used was the Linux ath5k driver. A network interface card with the Intel 3945ABG chipset was tested, but not usable as the attacker because it replaced the source MAC address of all transmitted frames with the MAC address of the network interface card. The implementation listens for a TKIP frame using 802.11 monitor mode, then modifies and retransmits the frame to invoke the TKIP countermeasures. The source code for the modification and retransmission of frames is included in Appendix 3.C.

Run-time configuration of several attack parameters is supported to make the vulnerability assessment tool more flexible. The default behavior is to listen for TKIP frames from the AP, and then perform the attack. The run-time options for the vulnerability assessment tool, tkipdos-ng, are included in Appendix 3.D.

To give wireless network administrators the opportunity to test the vulnerability of their networks, tkipdos-ng will be made available as free software licensed under the GNU General Public License version 2.

3.5 Experimental Validation

Based on the analysis in Section 3.3, several vulnerability tests were constructed to validate the theoretical analysis on a wide array of different products. The hypotheses and experimental design are detailed in this section. The hypotheses to be tested were as follows:

1. Converting a non-QoS TKIP frame into a QoS TKIP frame with priority 1, 2 or 3 will cause a MIC failure in clients with QoS support even if QoS is disabled in the AP

2. Clients using CCMP as the pairwise cipher and TKIP as the group cipher will invoke countermeasures if they experience two TKIP MIC failures within a 60 second time period

3. Clients using CCMP as the pairwise cipher and TKIP as the group cipher will not connect to an AP if TKIP countermeasures in the client are currently active for that AP

4. Clients using CCMP as the pairwise cipher and TKIP as the group cipher will not be able to establish a connection to an AP with active countermeasures

To test the hypotheses, two experiments were designed. The first experiment tests hypotheses 1, 2 and 3. An AP is configured with support for both TKIP and CCMP. QoS and TKIP countermeasures are disabled in the AP. Two clients connect to the AP using CCMP as the pairwise cipher suite. The attacker listens for broadcast or multicast TKIP frames from the AP. When such a frame is observed, the attacker converts the frame to a QoS frame with priority 1, 2 or 3 and retransmits the frame twice. The attacker then observes the effect on the client that did not transmit the original broadcast frame. Since TKIP countermeasures are disabled in the AP, this experiment isolates the effect of MIC failures in the client. If hypotheses 1, 2 and 3 are true, then the client invokes countermeasures and will not be able to reconnect to the AP for at least 60 seconds.

The second experiment tests hypothesis 4. An AP is configured with support for both TKIP and CCMP. QoS and TKIP countermeasures are enabled in the AP. A client connects to the AP using TKIP as the pairwise cipher suite. The attacker listens for TKIP frames from the client to the AP. When such a frame is observed, the attacker converts the frame to a QoS frame if it is non-QoS, changes the frame priority to 1, 2 or 3 and retransmits the frame twice. The attacker then observes the effect on the AP. If the AP has invoked countermeasures, a client using CCMP as the pairwise cipher suite tries to connect to the AP. If hypothesis 4 is true, the client using CCMP should not be able to connect to the AP for at least 60 seconds after the countermeasures were invoked.

3.6 Results

The first experiment was performed with different clients to test whether the vulnerabilities were general or implementation specific. A Linksys WRT54GL wireless router with the OpenWrt [43] Kamikaze r19286 firmware was used as the AP. The hostapd [44] implementation in the firmware was modified so that it did not invoke TKIP countermeasures. The clients were configured to use CCMP as the pairwise cipher suite. The clients used for the experiment and the results are listed in Table 3.1. All of the clients that supported 802.11e QoS were vulnerable to the attack. Transmitting two modified broadcast or multicast frames from the attacker invoked

Hardware	Operating System	QoS Support	Vulnerable
Apple iMac 11.1	Mac OSX 10.6.2	Yes	Yes
Apple iPhone	iPhone OS	No	No
Apple iPhone 3G	iPhone OS	No	No
Asus EEE 901	Mandriva Linux 2010.0	Yes	Yes
Compaq 8510p	Windows Vista Ultimate	Yes	Yes
Compaq CQ60	Windows Vista Home Basic	Yes	Yes
Dell Latitude D620	Windows XP	Yes	Yes
Dell Latitude D630	Fedora Linux 11	Yes	Yes
Dell Latitude E4200	Windows 7	Yes	Yes
HTC Hero	Google Android	Yes	Yes
HTC S710	Windows Mobile 6	Yes	Yes
Nokia N810	Maemo Linux 4	Yes	Yes
Nokia N900	Maemo Linux 5	No	No

Table 3.1: Equipment used in the first experiment. All 802.11e QoS supported clients were vulnerable to the DoS attack.

Hardware	Firmware	Vulnerable
Cisco 1242AG	4.1.192.35M	Yes
Linksys WRT54GL	Linksys 4.30.13	Yes
Linksys WRT54GL	OpenWrt 8.09.2 [43]	Yes
Linksys WRT54GL	Tomato 1.27 [45]	Yes
Netgear WNR1000	V1.0.1.5	Yes

Table 3.2: APs used in the second experiment. All of the APs refused to establish new connections using CCMP as the pairwise cipher and TKIP as the group cipher while the attack countermeasures were active.

countermeasures on all the clients, and caused the clients to send MIC failure notifications to the AP. The clients were unable to establish a new connection to the AP for 60 seconds.

The second experiment was performed with different APs to test whether the vulnerabilities were general or implementation specific. The APs used for the experiment and the results are listed in Table 3.2. Once the AP invoked countermeasures, both clients using CCMP and clients using TKIP as the pairwise cipher suite were unable to establish a new connection for 60 seconds.

3.7 Discussion

The test results confirmed all of the interpretations in Section 3.3. Networks that support both TKIP and CCMP are vulnerable to DoS attacks that invoke the TKIP countermeasures, and such attacks cause all clients to be disconnected for 60 seconds. Disabling QoS support on the AP does not prevent an attack against the clients. Furthermore, as long as at least one associated client supports QoS, the attack will cause it to send MIC failure notifications to the AP. The AP will invoke countermeasures and disconnect all clients, including those that do not support QoS.

Since this cryptographic DoS attack only affects networks using 802.11i, it could be used as a security rollback attack. When the attack is mounted, networks using weaker security mechanisms are functional, while networks using 802.11i are disrupted.

During the experiments, we observed that once clients reconnected after the 60 seconds of downtime, they transmitted several broadcast frames. The client operating systems used the Address Resolution Protocol (ARP) [46] and the Dynamic Host Configuration Protocol (DHCP) [47] to establish Internet Protocol (IP) connectivity. These protocols use broadcast messages that could be captured, modified and retransmitted as a new attack. Once the countermeasures were deactivated, a new attack immediately invoked the countermeasures again.

The vulnerability presented in this paper can be removed by modifying TKIP. If the QoS priority is used as input to the TKIP key mixing function, then a modified priority will result in a different RC4 key stream. A modified frame would then be rejected by the recipient due to a failed ICV check.

A network administrator could split a network that supports TKIP and CCMP into two logical networks to reduce the exposure to the vulnerability. One network would support TKIP only and the other would support CCMP only. This approach guarantees that attacks against TKIP do not affect clients using CCMP.

Another partial solution is to prevent broadcast and multicast traffic from the AP. By default, the Cisco Unified Wireless Network design [48] is configured so that the AP does not transmit any broadcast or multicast frames on the wireless network. As long as none of the clients use TKIP as the pairwise cipher suite, the attack does not work against such networks.

3.8 Conclusions

The DoS attack described in this paper is practical, easy to implement, and can be mounted using off the shelf hardware and readily available software. Transmitting two modified frames disrupts a TKIP/CCMP-based 802.11 network for 60 seconds. It is one of the most efficient known DoS attacks against 802.11. The attack works even if all clients use CCMP as the pairwise cipher and QoS support is disabled in the AP. There are several ways to mitigate or prevent the attack, and recommendations are given in Section 3.7.

3.9 Acknowledgments

Professor Stig F. Mjølsnes provided valuable discussions and feedback about the topics in this paper. Jing Xie provided valuable feedback for revising this paper. The following people and organization contributed their time and equipment for use in the experiments: Hans Almåsbakk, Steinar Andresen, Danilo Gligoroski, Linda Ariani Gunawan, Stig F. Mjølsnes, Pål S. Sæther, Benedikt Westermann, Jing Xie and the Wireless Trondheim project.

3.A Message Sequence Diagrams

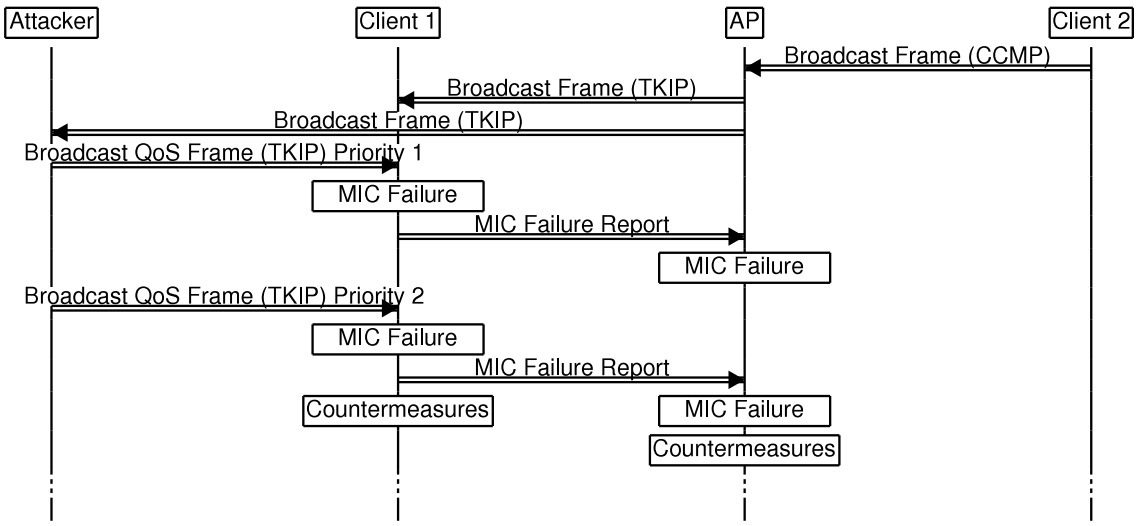

Figure 3.3: The cryptographic DoS attack against clients using CCMP as the pairwise cipher suite. Two transmitted frames from the attacker invokes countermeasures in all clients except the originator of the broadcast frame. Countermeasures are also invoked in the AP due to the MIC failure reports from the clients.

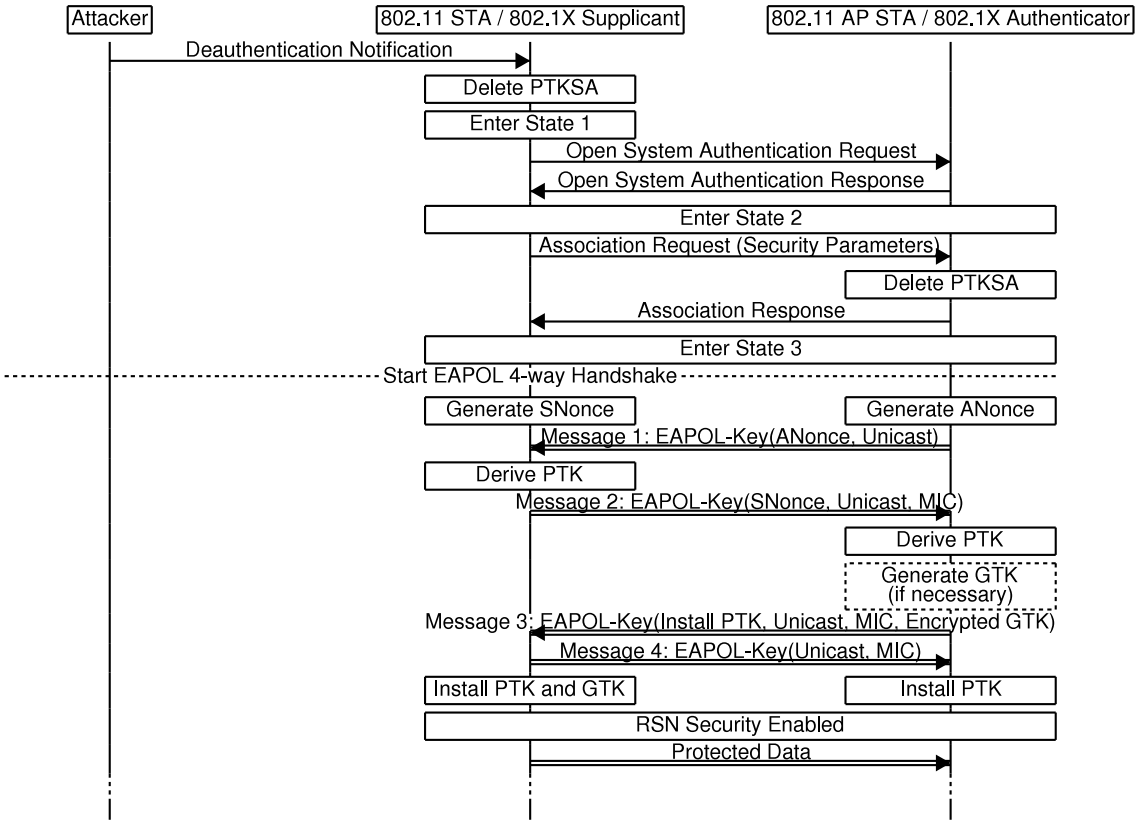

Figure 3.4: The deauthentication attack against a client using 802.11i

3.B TKIP Frame Structure

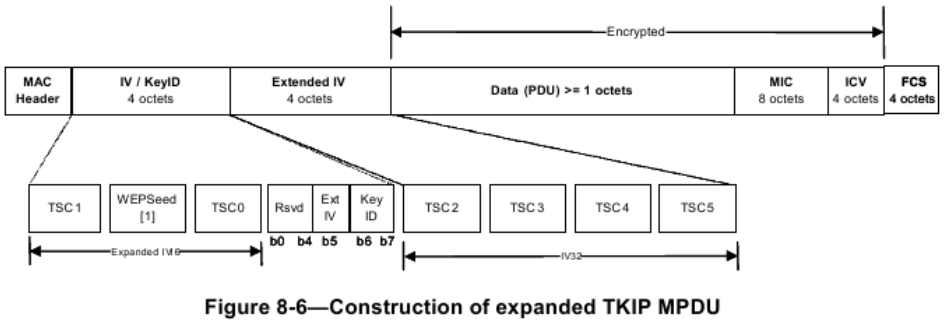

Figure 3.5: An 802.11 TKIP frame [28]

3.C Vulnerability Assessment Tool Source Code

```
qos = 0;
// z = MAC header length
```

```
z = ( ( h80211[1] & 3 ) != 3 ) ? 24 : 30;
if ( ( h80211[0] & 0x80 ) == 0x80 )
{
  qos = 1;
  z += 2;
}
if(qos == 0) // If the frame does not have a QoS Control
             // field then insert one
{
  // QoS data
  h80211[0] |= 0x80;
  // Move frame body 2 bytes to the right
  // to make room for QoS control field
  for(i=caplen+1; i>z+1; i--)
    h80211[i] = h80211[i-2];
  // Add 2 bytes QoS control field
  caplen += 2;
  h80211[24] = 0x00;
  h80211[25] = 0x00;
}

// QoS priority (TID)
tid = h80211[24] & 0x03;

if(tid >= opt.r_npkts)
  tid = opt.r_npkts;

for(i=0; i<=opt.r_npkts; i++)
{
  if(i != tid)
  {
    // Set QoS priority
    h80211[24] = i;
    // Send frame
    send_packet(h80211, caplen);
  }
}
```

3.D Vulnerability Assessment Tool Command Line Parameters

```
usage: tkipdos-ng <options> <replay interface>
Filter options:
    -d dmac   : MAC address, Destination
```

```
    -s smac   : MAC address, Source
    -t tods   : frame control, To DS bit
    -f fromds : frame control, From DS bit
    -D        : disable AP detection
Replay options:
    -a bssid  : set target AP MAC address
    -e essid  : set target AP SSID
    -n npkts  : number of replayed frames per frame captured [1-3]
    -m natks  : number of attacks (keep going forever if not set)

    --help    : Displays this usage screen
```

Chapter 4

Paper C

Published in:

Martin Eian and Stig F. Mjølsnes

"The Modeling and Comparison of Wireless Network Denial of Service Attacks"

Proceedings of the 3rd ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handhelds (MobiHeld '11)[1]

ACM, 2011
ISBN 978-1-4503-0980-6

[1]The material in subappendices 4.A, 4.B and 4.C was removed from Paper C due to space constraints, and has not been peer reviewed or published by MobiHeld '11.

Abstract

Mobile handhelds with wireless access are used in numerous safety critical applications. The wireless network protocols in use are vulnerable to a wide array of denial of service attacks. We propose a formal method for modeling semantic denial of service attacks against wireless network protocols. We then use our proposed model to find a new deadlock vulnerability in IEEE 802.11. The history of published denial of service attacks against wireless protocols indicates that formal methods can contribute to the construction of robust protocols.

4.1 Introduction

The use of mobile handhelds in safety critical applications is increasing. Such devices are used in life critical medical systems, intelligent transportation systems (ITS), emergency communications and alarm systems. Furthermore, the current trend is to use standard commercial off the shelf (COTS) equipment and protocols in safety critical applications. Such safety critical applications require that the wireless networks used for communication by the mobile handhelds are available when needed.

The availability of a wireless network can be disrupted by denial of service (DoS) attacks. An adversary mounting a DoS attack on a wireless network used in safety critical applications could cause injury or death, as well as significant material damage.

We divide wireless network DoS attacks into four categories: *Jamming* attacks, *flooding* attacks, *semantic* attacks and *implementation specific* attacks. *Jamming* attacks are mounted by transmitting noise in the radio frequencies used by the wireless network. *Flooding* attacks exhaust resources by sending a large amount of messages to a protocol participant. *Semantic* attacks exploit protocol weaknesses by transmitting valid protocol messages with forged message fields. One example of a semantic attack is the deauthentication attack against IEEE 802.11 networks [5]. Finally, *implementation specific* attacks target implementation vulnerabilities in specific hardware or software. This category of attacks includes transmitting invalid protocol messages, since a correct implementation should discard all invalid messages. Implementation specific attacks are not related to the protocol design and will not be considered further in this paper.

The link layer protocols used in current wireless networks are vulnerable to semantic DoS attacks. Vulnerable protocols are used in 802.11 wireless local area networks (WLANs) [5, 19, 20, 40], 802.16 broadband networks [49], and GSM and UMTS mobile networks [50]. Due to functionality, performance or cost requirements, certain signaling messages used by these protocols are not integrity protected. One obvious solution to this problem is to integrity protect every message. However, a wireless network needs to exchange signaling messages, e.g. signal strength measurements, before it starts the authentication and key agreement (AKA) procedure. These initial signaling messages cannot be protected using the keys derived from the AKA procedure. Furthermore, small battery powered devices might not have the resources

to verify the integrity of every message. The integrity protection of time critical signaling messages in wireless networks might limit the usable bandwidth. The functionality of the wireless network might even depend on the use of unprotected messages, e.g. a network that has to provide service to clients that do not support any security mechanisms. As a consequence, these link layer protocols only protect a subset of all protocol messages. Any unprotected message could potentially be used by an adversary to disrupt the wireless network. Even the security mechanisms in the long term evolution (LTE) fourth generation (4G) mobile networks only protect a subset of the protocol messages [51].

A significant research effort has been invested in making network protocols more robust against *jamming* and *flooding* attacks. The research results are formal methods, models, tools and design principles to aid protocol designers [4, 10–13, 15, 16]. The research effort on *semantic* DoS attack detection and prevention has been of a more informal nature. A multitude of semantic DoS attacks have been discovered in existing wireless protocols, practical attacks have been demonstrated, and ad hoc countermeasures have been proposed [5, 19, 20, 40]. The fact that new semantic vulnerabilities are routinely being discovered in proposed and operative protocols indicates that it is difficult to avoid protocol vulnerabilities. To improve the robustness of wireless networks used in safety critical applications, we need to verify that an adversary cannot exploit unprotected protocol messages to disrupt the service provided by the protocol.

Narayana et al. proposed a formal model for evaluating *semantic* DoS attacks in 2006 [17]. To the best of our knowledge, this is the *only* proposed model of semantic DoS attacks in the literature prior to our work. Narayana et al. use the temporal logic of actions (TLA+) to model the protocol and adversary, and to specify the model properties. They then apply the TLA+ model checker (TLC) to find DoS vulnerabilities. They define a DoS attack as a situation where the protocol participants cannot reach their final state. Our proposed model has three major contributions compared to their model. First, we propose a cost model. The cost model provides an objective quantification of the severity of the protocol vulnerabilities. Second, our model is able to detect scenarios where participants reach their final state, and are then desynchronized by a new attack. This kind of attack will not be detected in the model proposed by Narayana et. al., since the participants in this case are able to reach their final state. Third, we demonstrate the usefulness of our model through the detection and experimental validation of a previously unknown deadlock vulnerability in the 802.11 standard.

In this paper we propose a formal method and model for evaluating wireless network protocol vulnerabilities to semantic DoS attacks. We analyze the adversary goals to find an appropriate quantification of the adversary cost. We then quantify the protocol participant cost, and propose an attack efficiency definition. Finally, we use our model to discover a new deadlock vulnerability in the IEEE 802.11 family of standards. The proposed formal method is not protocol specific, it can be used to analyze any wireless protocol.

Figure 4.1: Protocol model with initiator and responder. The adversary is modeled as the network. The adversary can read, replay or forge every unprotected protocol message. The adversary cannot delete or intercept (i.e. read and delete) protocol messages.

Our work complements the research efforts to make wireless networks more robust against jamming and flooding attacks. A network must be robust against all categories of DoS attacks. If one category of attacks is not addressed, then an adversary may still be able to disrupt the network.

The rest of this paper is structured as follows: Section 4.2 presents the formal protocol model and adversary model. Section 4.3 presents the cost model. Section 4.4 presents our model of IEEE 802.11 and Section 4.5 presents the experimental results. Finally, Section 4.6 gives the conclusions.

4.2 Protocol and Adversary Model

We model a two party protocol $\boldsymbol{P}$ as illustrated in Figure 4.1. The protocol participants are the initiator $\boldsymbol{I}$ and responder $\boldsymbol{R}$.

Each participant has a set of local protocol states. We model $\boldsymbol{I}$ and $\boldsymbol{R}$ as *deterministic* finite state transducers defined by the 7-tuples

$$\boldsymbol{I} = (\Sigma, \mathrm{S}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{I}_0}, \delta_{\boldsymbol{I}}, \omega_{\boldsymbol{I}}, \gamma_{\boldsymbol{P}}, \mathrm{F}_{\boldsymbol{I}})$$

$$\boldsymbol{R} = (\Sigma, \mathrm{S}_{\boldsymbol{R}}, \mathrm{s}_{\boldsymbol{R}_0}, \delta_{\boldsymbol{R}}, \omega_{\boldsymbol{R}}, \gamma_{\boldsymbol{R}}, \mathrm{F}_{\boldsymbol{R}})$$

where:

- ϵ is the empty string
- $\Sigma = \{\sigma | \sigma \text{ is a protocol message}\} \cup \{\epsilon\}$
- $\mathrm{S}_{\boldsymbol{I}} = \{\mathrm{s}_{\boldsymbol{I}} | \mathrm{s}_{\boldsymbol{I}} \text{ is a protocol initiator state}\}$
- $\mathrm{S}_{\boldsymbol{R}} = \{\mathrm{s}_{\boldsymbol{R}} | \mathrm{s}_{\boldsymbol{R}} \text{ is a protocol responder state}\}$
- $\mathrm{s}_{\boldsymbol{I}_0} \in \mathrm{S}_{\boldsymbol{I}}$ is the protocol initiator initial state
- $\mathrm{s}_{\boldsymbol{R}_0} \in \mathrm{S}_{\boldsymbol{R}}$ is the protocol responder initial state
- $\delta_{\boldsymbol{I}} : \mathrm{S}_{\boldsymbol{I}} \times \Sigma \rightarrow \mathrm{S}_{\boldsymbol{I}}$ is the initiator state transition function
- $\delta_{\boldsymbol{R}} : \mathrm{S}_{\boldsymbol{R}} \times \Sigma \rightarrow \mathrm{S}_{\boldsymbol{R}}$ is the responder state transition function
- $\omega_{\boldsymbol{I}} : \mathrm{S}_{\boldsymbol{I}} \times \Sigma \rightarrow \Sigma$ is the initiator output function

- $\omega_{\boldsymbol{R}} : \mathrm{S}_{\boldsymbol{R}} \times \Sigma \rightarrow \Sigma$ is the responder output function
- $\gamma_{\boldsymbol{I}} : \mathrm{S}_{\boldsymbol{I}} \times \Sigma \rightarrow \mathbb{R}^{+}$is the initiator cost function
- $\gamma_{\boldsymbol{R}} : \mathrm{S}_{\boldsymbol{R}} \times \Sigma \rightarrow \mathbb{R}^{+}$is the responder cost function
- $\mathrm{F}_{\boldsymbol{I}} \subseteq \mathrm{S}_{\boldsymbol{I}} = \{\mathrm{s}_{\boldsymbol{I}} | \mathrm{s}_{\boldsymbol{I}} \in \mathrm{S}_{\boldsymbol{I}}$ and $\boldsymbol{P}$ provides service$\}$
- $\mathrm{F}_{\boldsymbol{R}} \subseteq \mathrm{S}_{\boldsymbol{R}} = \{\mathrm{s}_{\boldsymbol{R}} | \mathrm{s}_{\boldsymbol{R}} \in \mathrm{S}_{\boldsymbol{R}}$ and $\boldsymbol{P}$ provides service$\}$

The service provided by a protocol depends on the protocol's purpose. One common service is the transport of higher layer user data. The protocol $\boldsymbol{P}$ is in a state $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}})$ where it provides service when $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}}) \in (\mathrm{F}_{\boldsymbol{I}} \times \mathrm{F}_{\boldsymbol{R}})$.

We model an adversary $\boldsymbol{A}$ who can read, replay or forge every unprotected protocol message $\sigma_{\boldsymbol{A}} \in \Sigma_{\boldsymbol{A}}$. The adversary cannot delete or intercept (i.e. read and delete) messages. These capabilities correspond to a real world adversary who utilizes commercial off the shelf hardware and software. An example of such an adversary is someone with an 802.11 network interface card with driver support for monitor mode and frame injection. When not attacking, the adversary simply forwards all messages between $\boldsymbol{I}$ and $\boldsymbol{R}$. We model the attack behavior of the adversary as a *nondeterministic* finite state transducer defined by the 7-tuple

$$\boldsymbol{A} = (\Sigma_{\boldsymbol{A}}, \mathrm{S}_{\boldsymbol{A}}, \mathrm{s}_{\boldsymbol{A}_0}, \delta_{\boldsymbol{A}}, \omega_{\boldsymbol{A}}, \gamma_{\boldsymbol{A}}, \mathrm{F}_{\boldsymbol{A}})$$

where:

- $\Sigma_{\boldsymbol{A}} \subseteq \Sigma = \{\sigma_{\boldsymbol{A}} | \sigma_{\boldsymbol{A}} \in \Sigma$ and $\boldsymbol{A}$ can forge $\sigma_{\boldsymbol{A}}\} \cup \{\epsilon\}$
- $\mathrm{S}_{\boldsymbol{A}} = \{\mathrm{s}_{\boldsymbol{A}} | \mathrm{s}_{\boldsymbol{A}}$ is an adversary state$\}$
- $\mathrm{s}_{\boldsymbol{A}_0} \in \mathrm{S}_{\boldsymbol{A}}$ is the adversary initial state
- $\delta_{\boldsymbol{A}} : \mathrm{S}_{\boldsymbol{A}} \rightarrow \mathcal{P}(\mathrm{S}_{\boldsymbol{A}})$ is the adversary state transition function
- $\omega_{\boldsymbol{A}} : \mathrm{S}_{\boldsymbol{A}} \rightarrow \Sigma_{\boldsymbol{A}} \times \Sigma_{\boldsymbol{A}}$ is the adversary output function
- $\gamma_{\boldsymbol{A}} : \mathrm{S}_{\boldsymbol{A}} \rightarrow \mathbb{R}^{+}$is the adversary cost function
- $\mathrm{F}_{\boldsymbol{A}} \subseteq \mathrm{S}_{\boldsymbol{A}} = \{\mathrm{s}_{\boldsymbol{A}} | \mathrm{s}_{\boldsymbol{A}} \in \mathrm{S}_{\boldsymbol{A}}$ and $\boldsymbol{A}$ has finished attack$\}$

The function $\omega_{\boldsymbol{A}}$ outputs a pair of messages. The first message is transmitted to $\boldsymbol{I}$, and the second message to $\boldsymbol{R}$. If the output is (ϵ, ϵ), then no messages are transmitted. The adversary mounts an attack by transmitting one or more messages $\sigma_{\boldsymbol{A}}$. A successful attack triggers a transition from a protocol state $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}}) \in (\mathrm{F}_{\boldsymbol{I}} \times \mathrm{F}_{\boldsymbol{R}})$ to a protocol state $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}}) \notin (\mathrm{F}_{\boldsymbol{I}} \times \mathrm{F}_{\boldsymbol{R}})$. We limit the number of messages that the adversary can transmit. Once this limit is reached, the adversary transitions to a state $\mathrm{s}_{\boldsymbol{A}} \in \mathrm{F}_{\boldsymbol{A}}$.

4.3 Cost Model

The model presented in Section 4.2 can be used without a cost model to find deadlock vulnerabilities in a protocol. However, we have to define a cost model to find other semantic DoS vulnerabilities that do not cause a deadlock, since a protocol might be vulnerable to highly efficient attacks that cause temporary disruption. First, we define four additional functions. The function $\tau_{\mathrm{m}} : \Sigma \rightarrow \mathbb{R}^{+}$, with $\tau_{\mathrm{m}}(\epsilon) = 0$, represents the transmission time of a protocol message σ. The function $\tau_{\mathrm{o}} : \Sigma \rightarrow \mathbb{R}^{+}$, with $\tau_{\mathrm{o}}(\epsilon) = 0$, represents the protocol overhead time of a message σ, e.g. waiting for a time slot before message transmission. Finally, we define the functions:

$$\tau(\sigma) = \tau_{\mathrm{m}}(\sigma) + \tau_{\mathrm{o}}(\sigma)$$

$$\tau_{\mathrm{m_{sum}}}(\sigma_1, \sigma_2) = \tau_{\mathrm{m}}(\sigma_1) + \tau_{\mathrm{m}}(\sigma_2)$$

We assume that the adversary seeks to maximize network disruption and minimize the probability of being located. The adversary has to be in physical proximity of a wireless network in order to mount an attack. Location determination methods such as triangulation or trilateration can be used by the network operator to determine the physical location of the adversary. If an adversary is located, then an ongoing attack can be stopped and the adversary can be apprehended. The precision of location determination depends on the number of measurements. The longer an adversary transmits a wireless signal, the higher the probability of being located. An adversary would thus limit his transmission time to a minimum to avoid being located.

The time spent transmitting a signal is also strongly correlated with energy consumption. If an adversary can mount an attack by very infrequent transmissions, then he could use a battery powered device to cause long term disruption of the network. Such long lived, low power devices are referred to as "cyber mines" [16]. The ability to use "cyber mines" reduces the risk to the adversary, since he no longer has to be physically present when the attack is mounted.

The adversary constraints are thus location determination time and energy usage. We propose that the transmission time of the messages used for the attack, measured in seconds, is the most appropriate quantification of the adversary cost $\Gamma_{\boldsymbol{A}}$. Formally, we first define $\gamma_{\boldsymbol{A}}$ as follows:

$$\gamma_{\boldsymbol{A}}(\mathrm{s}_{\boldsymbol{A}}) = \tau_{\mathrm{m_{sum}}}(\omega_{\boldsymbol{A}}(\mathrm{s}_{\boldsymbol{A}}))$$

We define the adversary cost $\Gamma_{\boldsymbol{A}}$ as the cumulative output of $\gamma_{\boldsymbol{A}}$, with $\Gamma_{\boldsymbol{A}} = 0$ in the initial state $\mathrm{s}_{\boldsymbol{A}_0}$. A more sophisticated adversary model could include the computational cost of constructing a message or breaking certain cryptographic primitives as part of the cost function $\gamma_{\boldsymbol{A}}$. We do not include these costs in our model, since we model an adversary that only transmits unprotected messages. The energy costs of computation are thus insignificant compared to the energy costs of transmission.

A wireless network provides one or more services. A DoS attack causes a time period where service is not provided. In our model, this is represented by the time

spent in protocol states $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}}) \notin (\mathrm{F}_{\boldsymbol{I}} \times \mathrm{F}_{\boldsymbol{R}})$. We propose to use this time period, measured in seconds, to quantify the protocol cost $\Gamma_{\boldsymbol{P}}$. Formally, we first define $\gamma_{\boldsymbol{I}}$ and $\gamma_{\boldsymbol{R}}$ as follows:

$$\gamma_{\boldsymbol{I}}(\mathrm{s}_{\boldsymbol{I}}, \sigma) = \begin{cases} 0 & \text{if } \mathrm{s}_{\boldsymbol{I}} \in \mathrm{F}_{\boldsymbol{I}} \text{ and} \\ & \delta(\mathrm{s}_{\boldsymbol{I}}, \sigma) \in \mathrm{F}_{\boldsymbol{I}} \\ \tau(\omega(\mathrm{s}_{\boldsymbol{I}}, \sigma)) & \text{if } \mathrm{s}_{\boldsymbol{I}} \notin \mathrm{F}_{\boldsymbol{I}} \\ \tau(\omega(\mathrm{s}_{\boldsymbol{I}}, \sigma)) & \text{if } \mathrm{s}_{\boldsymbol{I}} \in \mathrm{F}_{\boldsymbol{I}} \text{ and} \\ & \delta(\mathrm{s}_{\boldsymbol{I}}, \sigma) \notin \mathrm{F}_{\boldsymbol{I}} \end{cases}$$

$$\gamma_{\boldsymbol{R}}(\mathrm{s}_{\boldsymbol{R}}, \sigma) = \begin{cases} 0 & \text{if } \mathrm{s}_{\boldsymbol{R}} \in \mathrm{F}_{\boldsymbol{R}} \text{ and} \\ & \delta(\mathrm{s}_{\boldsymbol{R}}, \sigma) \in \mathrm{F}_{\boldsymbol{R}} \\ \tau(\omega(\mathrm{s}_{\boldsymbol{R}}, \sigma)) & \text{if } \mathrm{s}_{\boldsymbol{R}} \notin \mathrm{F}_{\boldsymbol{R}} \\ \tau(\omega(\mathrm{s}_{\boldsymbol{R}}, \sigma)) & \text{if } \mathrm{s}_{\boldsymbol{R}} \in \mathrm{F}_{\boldsymbol{R}} \text{ and} \\ & \delta(\mathrm{s}_{\boldsymbol{R}}, \sigma) \notin \mathrm{F}_{\boldsymbol{R}} \end{cases}$$

We then define $\Gamma_{\boldsymbol{I}}$ as the cumulative output of $\gamma_{\boldsymbol{I}}$, with $\Gamma_{\boldsymbol{I}} = 0$ in the initial state $\mathrm{s}_{\boldsymbol{I}_0}$. Next, we define $\Gamma_{\boldsymbol{R}}$ as the cumulative output of $\gamma_{\boldsymbol{R}}$, with $\Gamma_{\boldsymbol{R}} = 0$ in the initial state $\mathrm{s}_{\boldsymbol{R}_0}$. Finally, we define $\Gamma_{\boldsymbol{P}} = \Gamma_{\boldsymbol{I}} + \Gamma_{\boldsymbol{R}}$.

We compare the cumulative cost functions $\Gamma_{\boldsymbol{A}}$ and $\Gamma_{\boldsymbol{P}}$ to a physical layer jamming attack as an illustration. In a constant jamming attack, the adversary transmits noise in the radio frequencies used by the wireless network. As long as the adversary is transmitting, the protocol does not provide service. Once the adversary stops transmitting, the protocol immediately provides service again. In our model, $\Gamma_{\boldsymbol{A}}$ represents the time spent transmitting by the adversary and $\Gamma_{\boldsymbol{P}}$ represents the time period where the protocol does not provide service. Thus, an equivalent semantic attack would have $\Gamma_{\boldsymbol{A}} = \Gamma_{\boldsymbol{P}}$.

To achieve his goals, an adversary would try to find attacks where $\Gamma_{\boldsymbol{P}} \gg \Gamma_{\boldsymbol{A}}$. Such attacks could be considered as an amplifier for the adversary. An attack where $\Gamma_{\boldsymbol{P}} = 2\Gamma_{\boldsymbol{A}}$ is twice as efficient as a constant jamming attack, i.e. it has an amplification of 2. We define this amplification as the *attack efficiency* E:

$$E = \frac{\Gamma_{\boldsymbol{P}}}{\Gamma_{\boldsymbol{A}}}$$

The adversary's goal is thus to find an attack with the highest possible attack efficiency E. A semantic attack with $E > 1$ could be considered a protocol weakness. However, it is up to the protocol designer to decide the acceptable attack efficiency threshold.

Note that the cost model can be replaced by modifying the cost functions of the protocol and adversary models. Our proposed cost model is simple, which makes it easy to implement and model check. A more sophisticated cost model might give more realistic results. The tradeoff between realism and practicality in the cost model is one possible avenue of future work.

4.4 Modeling IEEE 802.11

We validate the usefulness of our proposed model by model checking IEEE 802.11 [28]. IEEE 802.11 is a family of standards for wireless local area networks (WLANs). The standard specifies the WLAN medium access control (MAC) and physical (PHY) layers. The 802.11 specification is unclear on several points, which require interpretation. We use the open source hostapd [44] and wpa_supplicant [52] implementations of 802.11 as a guideline for how to interpret ambiguities. We model a subset of the IEEE 802.11 MAC layer with the 802.11i [29] and 802.11h [41] amendments. The cost parameters are based on the 802.11g [53] PHY layer with a 54 Mbit/s transfer speed. We use the Promela language to implement the model and the SPIN model checker to find protocol vulnerabilities [54]. Our model consists of three entities: An access point (AP), a station (STA) and an adversary. The AP acts as the responder $\boldsymbol{R}$ and the STA acts as the initiator $\boldsymbol{I}$. The entities are modeled as Promela processes. The cumulative cost functions $\Gamma_{\boldsymbol{A}}$ and $\Gamma_{\boldsymbol{P}}$ are global variables that are incremented during state transitions. The network is modeled as two asynchronous message channels in Promela, one in each direction. The STA initiates a connection setup whenever it is in state $\mathrm{s}_{\boldsymbol{I}_0}$.

Once the AP and STA have performed the 802.11 authentication, association and key agreement procedures, they alternately send and receive data frames. We model $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}}) \in (\mathrm{F}_{\boldsymbol{I}} \times \mathrm{F}_{\boldsymbol{R}})$ as the state where the STA receives a valid data frame from the AP and then transmits a data frame to the AP. One major challenge of using the proposed cost based model is how to avoid an infinite state space. If the protocol participants enter an infinite loop through protocol states $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}}) \notin (\mathrm{F}_{\boldsymbol{I}} \times \mathrm{F}_{\boldsymbol{R}})$, then $\Gamma_{\boldsymbol{P}}$, and thus the state space, will keep increasing. We solve this by letting only one of the participants initiate the transfer of data frames once the protocol transitions from a state $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}}) \notin (\mathrm{F}_{\boldsymbol{I}} \times \mathrm{F}_{\boldsymbol{R}})$ to a state $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}}) \in (\mathrm{F}_{\boldsymbol{I}} \times \mathrm{F}_{\boldsymbol{R}})$. While in $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}}) \in (\mathrm{F}_{\boldsymbol{I}} \times \mathrm{F}_{\boldsymbol{R}})$ the participants only transmit a data frame once they receive a data frame. If an attack causes a desynchronization of the participants, then the 802.11 resynchronization mechanisms will enable the AP and STA to resume communication. If, however, the resynchronization mechanisms fail, then the model will deadlock. This property makes checking for protocol deadlock vulnerabilities simple, since SPIN has a built-in test for deadlocks.

Having implemented the model in Promela, we then need to specify the properties to be checked. We use linear temporal logic (LTL) for this purpose. The LTL formula used for checking the cost based model is:

$$\Box((\Gamma_{\boldsymbol{A}} = 0) \vee (\frac{\Gamma_{\boldsymbol{P}}}{\Gamma_{\boldsymbol{A}}} < T))$$

The parameter T specifies the attack efficiency threshold. The most efficient published DoS attack against 802.11 is the quiet attack from [40].

4.5 Experimental Results

We first set the threshold to the efficiency of the quiet attack to verify that the attack is found by the model checker. The result is that SPIN finds the quiet attack. We then proceed to set the threshold to slightly more than the efficiency of the quiet attack to see if a more efficient attack exists. The result is negative, the quiet attack is the most efficient attack against 802.11 in our model. We then gradually lower the threshold in order to find other less efficient attacks. The results are that the model checker is able to find the previously known semantic DoS attacks against 802.11, but we do not find any new attacks.

We then proceed to check for deadlocks. The result is that SPIN finds a new deadlock vulnerability in 802.11i. The adversary transmits a valid 802.11 Open System authentication request from the STA to the AP while the protocol participants are in a state $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}}) \in (\mathrm{F}_{\boldsymbol{I}} \times \mathrm{F}_{\boldsymbol{R}})$. The AP deletes its 802.11i security association with the STA, but stays in 802.11 State 3. All data frames from the STA to the AP are dropped, since they are encrypted and integrity protected with a key that the AP no longer has access to. The AP cannot transmit any data frames to the STA, since it is unable to sign and encrypt them. The 802.11 resynchronization mechanisms are not triggered since both the AP and the STA are in 802.11 State 3.

We proceed to experimentally validate the vulnerability. We set up an 802.11 network using a wireless router with the hostapd software as the AP and a laptop computer with the wpa_supplicant software as the STA. The adversary causes a protocol deadlock by transmitting a single authentication request frame. In practice, the STA is able to recover after approximately 7 minutes due to an internal timeout. This timeout is not specified in the 802.11 standard, however, so there is no guarantee that other implementations would be able to recover. Even with this ability to recover, the attack is far more efficient than any published DoS attacks against 802.11. The author of hostapd and wpa_supplicant has been notified about the vulnerability.

4.6 Conclusions

The history of published attacks against existing wireless protocols shows that the design of robust protocols could greatly benefit from formal analysis tools. We have proposed a formal method for modeling semantic DoS attacks against wireless networks and shown how the model can be used to discover protocol vulnerabilities. By this, we have found a new deadlock vulnerability in 802.11 and experimentally validated it. Our proposed model can facilitate the design of robust protocols by discovering vulnerabilities during the design process.

4.A 802.11 Background and Assumptions

IEEE 802.11 is a standard for wireless local area networks (WLANs). The standard specifies the WLAN medium access control (MAC) and physical (PHY) layers. The original 802.11 standard was published in 1997, and has since been updated numerous times by amendments. The amendments are denoted by letters, e.g. 802.11i specifies

security enhancements to the 802.11 standard. A new version of the 802.11 standard was published in 2007 [28]. 802.11-2007 includes all amendments published prior to 2007. Only a small subset of the relevant background material from the 802.11 standard can be included here due to space constraints. The reader is referred to 802.11-2007 [28] and 802.11w-2009 [30] for more details.

We use an infrastructure 802.11 WLAN as an example to illustrate the application of the cost and efficiency definitions in [21]. One or more stations (STAs) connect to an access point (AP). The AP forwards traffic between the STAs, as well as to and from an external network. We assume that the network is configured to use the extended rate physical layer with orthogonal frequency division multiplexing (ERP-OFDM) specified in 802.11g [53]. The physical layer transfer speed of the network is 54 Mbit/s. We analyze three different security configurations: Plain 802.11 without any security mechanisms, 802.11 with the robust security network (RSN) specified in the 802.11i [29] amendment, and 802.11 with protected management frames specified in the 802.11w [30] amendment. The use of 802.11w implies the use of 802.11i RSN. For the RSN configuration, we assume that the network supports both the transitional key integrity protocol (TKIP) and the counter mode with cipher block chaining message authentication code protocol (CCMP).

An 802.11 network supports three different frame types: Data frames, management frames and control frames. Data frames transport user data, management frames are used for signaling, and control frames are used for time critical signaling such as acknowledgments (ACKs). 802.11 supports reliable delivery of unicast data and management frames using ACKs. Broadcast and multicast frames are not acknowledged.

A STA has to perform the 802.11 authentication and association procedure illustrated in Figure 4.2 before sending and receiving user data. The Open System Authentication is a null authentication which is present due to historical reasons. It does not actually provide any authentication. When the 802.11i RSN is used, the STA and AP also have to perform an extensible authentication protocol over local area networks (EAPOL) 4-way handshake to mutually authenticate and derive a shared security association and session key. The session key is called the pairwise transient key (PTK).

We assume an ideal radio frequency (RF) medium where no collisions or backoff occur. Collisions and backoff would affect both the adversary and the legitimate protocol participants, and would thus add random noise to the quantification of the attack efficiency. The idealized RF medium assumption gives us an exact quantification of the semantic protocol vulnerability, rather than one polluted by random noise. We further assume that STAs always have data to send, and that all data frames transmitted by STAs have a payload length of 1500 bytes. Whenever the length of a certain management or control frame type is variable, we assume that all necessary fields for the network configuration are included, but that all fields are of the minimum length allowed. As an example, the service set identifier (SSID) field can be from 2-34 bytes long. We assume that this field is always present in relevant frames, and that it has a length of 2 bytes. Finally, we assume that the 802.11 distributed coordination function (DCF) is used.

We have to compute the transmission time of individual 802.11 frames before we

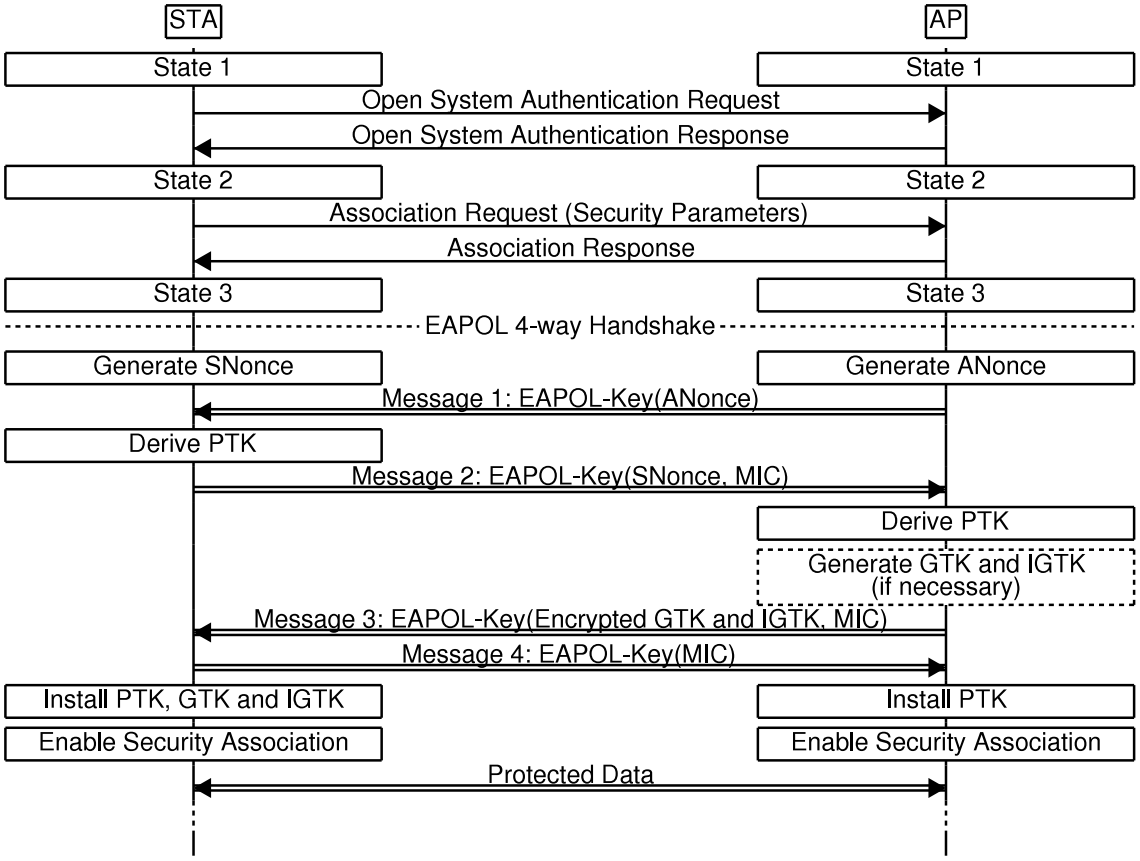

Figure 4.2: 802.11 authentication and association. If 802.11i RSN is used, the STA and AP have to perform an EAPOL 4-way handshake before the transport of user data is enabled.

can compute $\Gamma_{\boldsymbol{A}}$, $\Gamma_{\boldsymbol{P}}$ and E. Section 19.8.3.1 of 802.11-2007 [28] defines the total transmission time T of an 802.11g frame as:

$$\begin{aligned} \mathrm{T} &= \mathrm{T_{PREAMBLE}} + \mathrm{T_{SIGNAL}} + \\ &\quad \mathrm{T_{SYM}} * \mathrm{Ceiling}(\frac{16 + 8 * \mathrm{L} + 6}{\mathrm{N_{DBPS}}}) + \\ &\quad \mathrm{Signal\,Extension} \end{aligned}$$

Table 4.1 shows the timing parameters used in an 802.11g ERP-OFDM network with a transmission speed of 54 Mbit/s. After a frame has been transmitted, all stations wait for a time period defined as the inter-frame space (IFS) before they try to transmit a new frame[2]. If the new frame is an ACK frame, a clear to send (CTS) frame or the second or subsequent fragment of a fragment burst, then the transmitting station waits for a short IFS (SIFS) before trying to transmit. Otherwise, it waits for a DCF IFS (DIFS).

[2]The STAs will also wait for one or more time slots after the IFS due to the collision avoidance mechanism in 802.11, but as mentioned previously, we assume that no backoff occurs.

$T_{PREAMBLE}$	T_{SIGNAL}	T_{SYM}	N_{DBPS}	Signal Extension	SIFS	DIFS
16μs	4μs	4μs	216	6μs	10μs	28μs

Table 4.1: Timing parameters for 802.11g networks (54 Mbit/s)

Finally, we have to determine the frame length L and compute τ_{m} and τ_{o} for the frames used in the analysis in Section 4.B. Table 4.2 shows the results. For a detailed description of the 802.11 frame fields and lengths, see Section 7 of 802.11-2007 [28], 802.11r-2008 [55] and 802.11w-2009 [30].

4.B Analysis of Semantic DoS Attacks Against 802.11

We use the definitions introduced in [21] to assess the efficiency of several published DoS attacks against 802.11. The attacks being assessed are the deauthentication attack [5], the disassociation attack [5], the virtual carrier sense (CS) attack [5], the authentication request attack [19], the association request attack [19], the quiet attack [40] and the TKIP Countermeasures attack [20]. We give a detailed analysis of the deauthentication attack to illustrate the method of analysis. The analysis of the other attacks is presented as a summary.

The deauthentication attack described in [5] is mounted by sending a management frame of subtype deauthentication to the STA or AP. Figure 4.3 illustrates the attack against a STA, and Figure 4.4 illustrates the attack against an AP. The 802.11 authentication and association procedure must be performed to recover from the attack. If 802.11i is used, an EAPOL 4-way handshake also has to be performed. 802.11w prevents the deauthentication attack, since the deauthentication frames are integrity protected.

Table 4.2 shows that the transmission time of a deauthentication frame is 34μs in all three scenarios. The adversary cost $\Gamma_{\boldsymbol{A}_{\mathrm{deauth}}}$ is thus:

$$\Gamma_{\boldsymbol{A}_{\mathrm{deauth}}} = \tau_{\mathrm{m}}(\mathrm{deauth}) = \ 34\mu\mathrm{s}$$

The deauthentication attack can be directed against either the STA or the AP. We consider the attack against a plain 802.11 network, a network with 802.11i RSN and a network with 802.11w protected management frames, for a total of six combinations. Note that 802.11 uses a shared broadcast medium that can only be accessed by one participant at any given time. Thus, we add $\tau(\sigma_{\boldsymbol{A}})$ to $\Gamma_{\boldsymbol{P}}$ for every message $\sigma_{\boldsymbol{A}}$ transmitted by the adversary. The rationale for this adjustment is that the adversary denies service to the protocol participants while transmitting a message. We calculate $\Gamma_{\boldsymbol{P}}$ as follows:

Frame Type (σ)	$L_{802.11}$	$L_{802.11i}$	$L_{802.11w}$	$\tau_m(\sigma)_{802.11}$
ACK	14	14	14	30μs
Action (SA Query)	N/A	N/A	48	N/A
ARP Request (TKIP)	64	84	84	38μs
Authentication	34	34	34	34μs
Association Request	46	74	74	34μs
Association Response	57	57	57	38μs
Beacon	98	126	126	42μs
CTS	14	14	14	30μs
Data (CCMP)	1528	1544	1544	254μs
Deauthentication	30	30	46	34μs
Disassociation	30	30	46	34μs
EAPOL Msg 1	N/A	157	157	N/A
EAPOL Msg 2	N/A	153	153	N/A
EAPOL Msg 3	N/A	191	215	N/A
EAPOL Msg 4	N/A	125	125	N/A

Frame Type (σ)	$\tau_m(\sigma)_{802.11i}$	$\tau_m(\sigma)_{802.11w}$	$\tau_o(\sigma)$
ACK	30μs	30μs	10μs
Action (SA Query)	N/A	34μs	28μs
ARP Request (TKIP)	42μs	42μs	28μs
Authentication	34μs	34μs	28μs
Association Request	38μs	38μs	28μs
Association Response	38μs	38μs	28μs
Beacon	46μs	46μs	28μs
CTS	30μs	30μs	10μs
Data (CCMP)	258μs	258μs	28μs
Deauthentication	34μs	34μs	28μs
Disassociation	34μs	34μs	28μs
EAPOL Msg 1	50μs	50μs	28μs
EAPOL Msg 2	50μs	50μs	28μs
EAPOL Msg 3	58μs	62μs	28μs
EAPOL Msg 4	46μs	46μs	28μs

Table 4.2: L, τ_m and τ_o for relevant frame types. τ_o equals SIFS for ACK and CTS frames, and DIFS for all other frames. The broadcast ARP Request frame is protected using TKIP and the unicast data frame is protected using CCMP when the 802.11i RSN is enabled. Some management frame types contain additional information elements when the RSN is enabled, which can be observed as an increased L parameter.

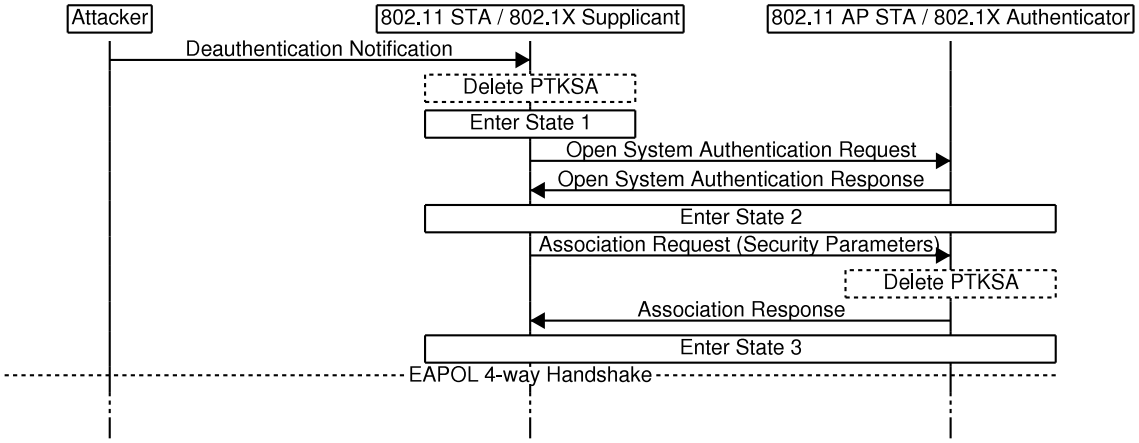

Figure 4.3: The deauthentication attack against a STA.

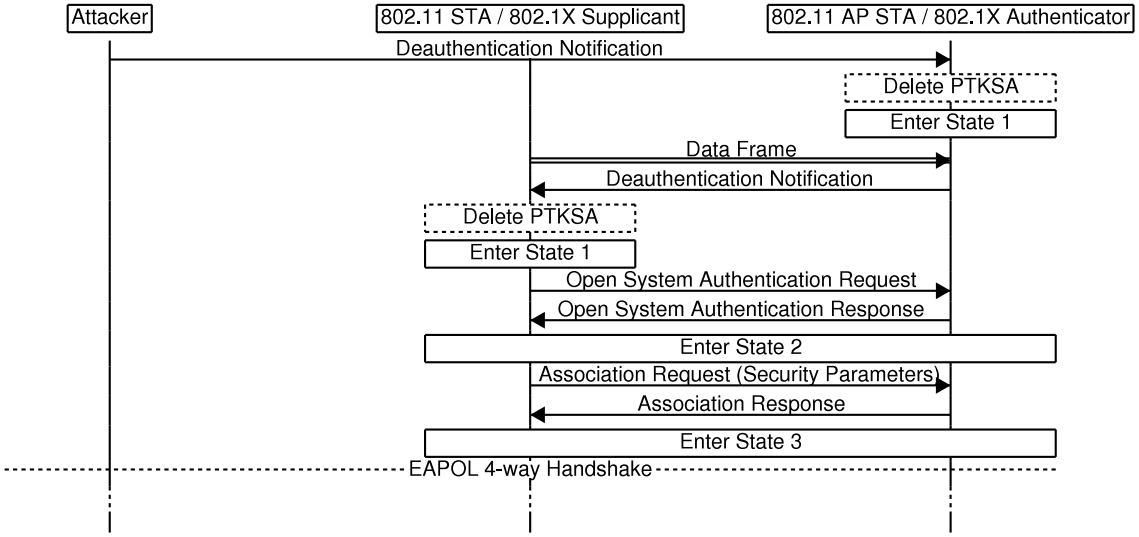

Figure 4.4: The deauthentication attack against an AP.

$$
\begin{aligned}
\Gamma_{\boldsymbol{P}_{\mathrm{deauthSTA(802.11)}}} &= \tau(\mathrm{deauth}) + \tau(\mathrm{ACK}) + \\
&\quad \tau(\mathrm{auth}) + \tau(\mathrm{ACK}) + \\
&\quad \tau(\mathrm{auth}) + \tau(\mathrm{ACK}) + \\
&\quad \tau(\mathrm{assocreq}) + \tau(\mathrm{ACK}) + \\
&\quad \tau(\mathrm{assocresp}) + \tau(\mathrm{ACK}) \\
&= 514\mu\mathrm{s}
\end{aligned}
$$

$$
\begin{aligned}
\Gamma_{\boldsymbol{P}_{\text{deauthAP(802.11)}}} &= \tau(\text{deauth}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{data}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{deauth}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{auth}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{auth}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{assocreq}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{assocresp}) + \tau(\text{ACK}) \\
&= 938\mu\text{s} \\
\Gamma_{\boldsymbol{P}_{\text{deauthSTA(802.11i)}}} &= \tau(\text{deauth}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{auth}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{auth}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{assocreq}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{assocresp}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{EAPOL1}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{EAPOL2}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{EAPOL3}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{EAPOL4}) + \tau(\text{ACK}) \\
&= 994\mu\text{s} \\
\Gamma_{\boldsymbol{P}_{\text{deauthAP(802.11i)}}} &= \tau(\text{deauth}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{data}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{deauth}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{auth}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{auth}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{assocreq}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{assocresp}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{EAPOL1}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{EAPOL2}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{EAPOL3}) + \tau(\text{ACK}) + \\
&\quad \tau(\text{EAPOL4}) + \tau(\text{ACK}) \\
&= 1422\mu\text{s} \\
\Gamma_{\boldsymbol{P}_{\text{deauthSTA(802.11w)}}} &= \tau(\text{deauth}) + \tau(\text{ACK}) \\
&= 102\mu\text{s} \\
\Gamma_{\boldsymbol{P}_{\text{deauthAP(802.11w)}}} &= \tau(\text{deauth}) + \tau(\text{ACK}) \\
&= 102\mu\text{s}
\end{aligned}
$$

802.11w provides integrity protection for deauthentication frames. We assume that the adversary is not able to send a valid integrity protected frame. Thus, once the recipient has acknowledged reception of the frame it will be silently discarded due to a failed integrity check.

Attack ($\sigma_{\boldsymbol{A}}$)	$\Gamma_{\boldsymbol{A}_{802.11}}$	$\Gamma_{\boldsymbol{A}_{802.11i}}$	$\Gamma_{\boldsymbol{A}_{802.11w}}$
Deauth(STA)	34μs	34μs	34μs
Deauth(AP)	34μs	34μs	34μs
Disassoc(STA)	34μs	34μs	34μs
Disassoc(AP)	34μs	34μs	34μs
Virtual CS	30μs	30μs	30μs
Authentication	34μs	34μs	34μs
Association	34μs	38μs	38μs
Quiet	42μs	46μs	46μs
TKIP	76μs	84μs	84μs
Attack ($\sigma_{\boldsymbol{A}}$)	$\Gamma_{\boldsymbol{P}_{802.11}}$	$\Gamma_{\boldsymbol{P}_{802.11i}}$	$\Gamma_{\boldsymbol{P}_{802.11w}}$
Deauth(STA)	514μs	994μs	102μs
Deauth(AP)	938μs	1422μs	102μs
Disassoc(STA)	310μs	790μs	102μs
Disassoc(AP)	734μs	1218μs	102μs
Virtual CS	32797μs	32797μs	32797μs
Authentication	734μs	1218μs	1222μs
Association	208μs	3001560μs	416μs
Quiet	67108864μs	67108864μs	67108864μs
TKIP	212μs	60001316μs	60001320μs

Table 4.3: The adversary cost $\Gamma_{\boldsymbol{A}}$ and protocol cost $\Gamma_{\boldsymbol{P}}$ for the assessed DoS attacks. The high protocol cost of the association attack against 802.11i is due to the AP initiating the EAPOL 4-way handshake and waiting for an answer to the first EAPOL message. After trying to send EAPOL message 1 three times, and waiting for 1 second after each transmitted message, the AP gives up and sends a deauthentication frame to the STA. The association attack against 802.11w causes the AP to initiate a security association (SA) Query procedure with the STA to verify the liveness of its SA.

We compute $\Gamma_{\boldsymbol{A}}$ and $\Gamma_{\boldsymbol{P}}$ for the other attacks using the same procedure as for the deauthentication attack. The results are presented in Table 4.3.

We can now compute the efficiency E of the DoS attacks. Figure 4.5 illustrates the results.

4.C Discussion

The cost functions and attack efficiency definition proposed in [21] can be used to quantitatively compare known DoS attacks, as illustrated in Section 4.B. The analysis shows that the quiet attack [40] is the most efficient published DoS attack

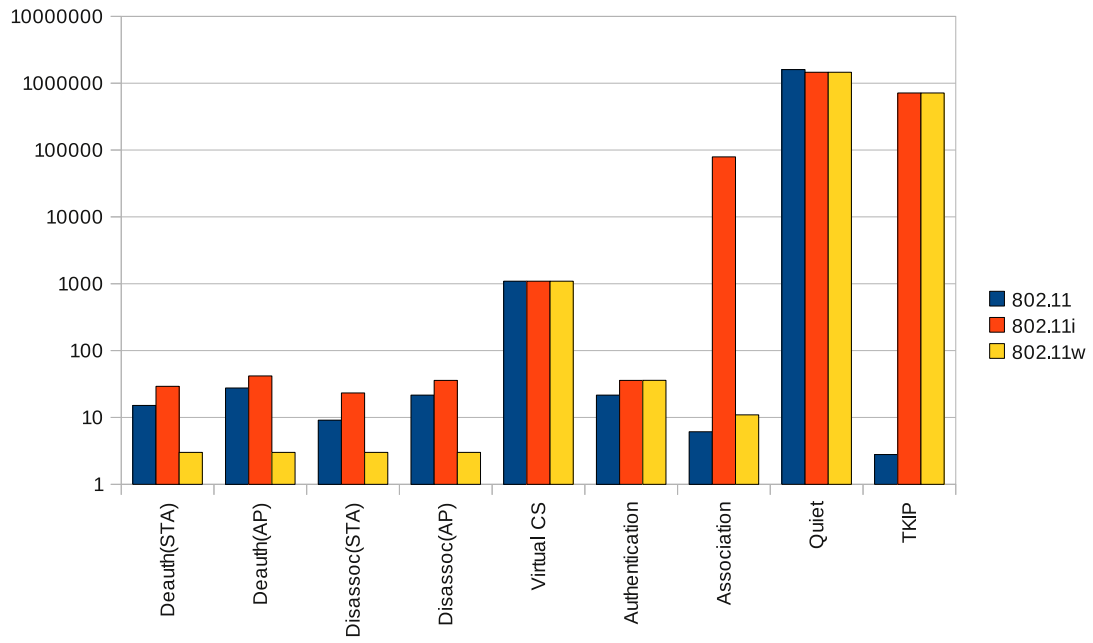

Figure 4.5: The efficiency E of the assessed DoS attacks. We use a logarithmic scale to illustrate the large differences in efficiency. The most efficient attacks are several orders of magnitude more efficient than any of the other attacks. Note that $E = 3$ for the DoS attacks that are prevented by the security mechanisms. This is due to the time slots and acknowledgment mechanisms in 802.11. The frame transmitted by the adversary will be acknowledged by the recipient, and during this time period the network will not provide service to the legitimate participants. It is thus impossible to prevent attacks with $E \leqq 3$ against 802.11 without significant changes to the standard.

against 802.11, closely followed by the TKIP Countermeasures attack [20]. The 802.11w amendment prevents the deauthentication, disassociation and association attacks, but provides no protection against the most efficient DoS attacks.

However, the main motivation of our work is to facilitate formal modeling of semantic protocol vulnerabilities. All of the analysis in this Appendix is purely theoretical, based on the protocol description. The analysis could thus be carried out at the design stage, before the protocol is implemented. Errors discovered at an early stage can be amended much more cheaply than errors discovered after implementation and deployment.

We are currently using Promela to implement a formal model of 802.11 that can be used to identify other, unknown protocol vulnerabilities or to verify the absence of semantic protocol vulnerabilities with the help of the SPIN model checker [56]. The finite state transducers are implemented as processes. The cumulative cost functions $\Gamma_{\boldsymbol{A}}$ and $\Gamma_{\boldsymbol{P}}$ are implemented as global variables that are incremented by the process cost functions $\gamma_{\boldsymbol{A}}$, $\gamma_{\boldsymbol{I}}$ and $\gamma_{\boldsymbol{R}}$. Message channels deliver the output messages of one process as input messages to another process. E is used in a linear

temporal logic (LTL) formula that specifies the acceptable attack efficiency threshold. As an example, given the LTL formula $\Diamond(E > 10)$, the SPIN model checker finds all attacks with $E > 10$, or proves that no such attacks are possible against the implemented protocol model. The model checker can also determine if an attack can cause a protocol deadlock. Preliminary results indicate that we are able to discover sophisticated multi-message attacks with high efficiency using our model. Such attacks are difficult to discover by manual analysis, and their discovery confirms the value of using a formal model.

Our proposed method could benefit from a more sophisticated adversary model. Future work includes modeling an adversary who can delete and intercept messages, break certain cryptographic primitives, and corrupt protocol participants. Modifying the adversary can be done independently of the protocol model, since we use a separate finite state transducer for the adversary.

The practical $\Gamma_{\boldsymbol{P}}$ of DoS attacks might be higher than the theoretical $\Gamma_{\boldsymbol{P}}$ presented in Section 4.B. One reason for this is that authentication and key agreement such as the EAPOL 4-way handshake include computationally expensive operations. To make $\Gamma_{\boldsymbol{P}}$ more realistic, one might include the computational cost (time) in addition to the transmission cost (time) to recover from an attack. Another reason is that higher layer protocols are involved. If the protocol initiator uses higher layer protocols such as the Dynamic Host Configuration Protocol (DHCP) [47] and the Address Resolution Protocol (ARP) [46], then adding the cost of these protocol messages to $\Gamma_{\boldsymbol{P}}$ would give a more realistic result.

Chapter 5

Paper D

Published in:

Martin Eian and Stig F. Mjølsnes

"A Formal Analysis of IEEE 802.11w Deadlock Vulnerabilities"

Proceedings of the 31^{st} *Annual IEEE International Conference on Computer Communications (IEEE INFOCOM 2012)*

IEEE, 2012
ISBN 978-1-4673-0775-8

Abstract

Formal methods can be used to discover obscure denial of service (DoS) vulnerabilities in wireless network protocols. The application of formal methods to the analysis of DoS vulnerabilities in communication protocols is not a mature research area. Although several formal models have been proposed, they lack a clear and convincing demonstration of their usefulness and practicality. This paper bridges the gap between theory and practice, and shows how a simple protocol model can be used to discover protocol deadlock vulnerabilities. A deadlock vulnerability is the most severe form of DoS vulnerabilities, thus checking for deadlock vulnerabilities is an essential part of robust protocol design. We demonstrate the usefulness of the proposed method through the discovery and experimental validation of deadlock vulnerabilities in the published IEEE 802.11w amendment to the 802.11 standard. We present the complete procedure of our approach, from model construction to verification and validation. An Appendix includes the complete model source code, which facilitates the replication and extension of our results. The source code can also be used as a template for modeling other protocols.

5.1 Introduction

Wireless network access protocols are used in numerous safety critical applications, such as life critical medical devices, supervisory control and data acquisition (SCADA) systems, smart grid applications, intelligent transportation systems (ITS), emergency communications and alarm systems. Network availability is important for safety critical applications, since loss of availability can cause physical damage. An adversary can disrupt the availability of a wireless network using denial of service (DoS) attacks.

The most widely deployed wireless protocols are vulnerable to DoS attacks. Throughout the last decade researchers have published DoS attacks against IEEE 802.11 local area networks (LANs) [5, 19, 20, 40], IEEE 802.16 wide area networks (WANs) [49] and GSM and UMTS mobile networks [50]. One of the most common forms of wireless DoS attacks is *semantic* attacks, i.e. to send valid protocol messages that cause one or more protocol participants to lose state synchronization. Semantic attacks can be highly efficient, since the participants may have to spend a significant amount of time to detect and correct the lost synchronization. The most severe semantic DoS attacks can cause a protocol deadlock. A deadlock state is a global state where the protocol participants are not able to recover to a functional state.

In this paper, we apply formal methods for the analysis of deadlock vulnerabilities in the IEEE 802.11 medium access control (MAC) layer [28] with the 802.11i [29] and 802.11w [30] amendments. The motivation for using 802.11w as our target protocol is that it has been subject to extensive manual analysis. The 802.11w designers found a deadlock vulnerability in an early draft of 802.11w. The protocol specification was modified because deadlock vulnerabilities were considered unacceptable. The 802.11w amendment has also been subject to manual analysis by independent researchers [19, 57]. Thus, we consider 802.11w as an appropriate subject for our investigation.

The main contribution of our work is a demonstration of *how* formal methods can be used to find deadlock vulnerabilities. In particular, we investigate how to automatically discover vulnerabilities through the construction and verification of a formal protocol model. Our work bridges the gap between theory and practice by giving a detailed description of how to construct and verify a simple and useful protocol model, including the complete model source code. The proposed approach to modeling and verification could help protocol designers discover deadlock vulnerabilities at an early stage of the design process.

Several formal models for the analysis of protocol DoS vulnerabilities have been proposed [4, 17, 58]. To the best of our knowledge, none of the proposed models have been demonstrated to be both easy to implement and able to discover deadlock vulnerabilities in protocols.

The rest of this paper is structured as follows: Section 5.2 introduces relevant parts of the 802.11 standard and related work. Section 5.3 constructs the model. Section 5.4 presents the verification results from the model checker. Section 5.5 is an experimental validation of the verification results. Section 5.6 discusses the results. Section 5.7 concludes the paper. The Appendix includes the complete source code of our model.

5.2 Background and Related Work

The IEEE 802.11 standard for wireless LANs (WLANs) was ratified in 1997 and accepted as an ISO standard in 1999. The most recent version of the standard is 802.11-2007 [28], which includes the 802.11i-2004 amendment [29]. 802.11i-2004 specifies security enhancements for the 802.11 MAC layer. The 802.11w-2009 amendment specifies protection of management frames of subtype Deauthentication, Disassociation and Action using the 802.11i security mechanisms [30]. Figure 5.1 shows the connection establishment protocol in an 802.11 network with the 802.11i and 802.11w amendments.

The 802.11 standard specifies three protocol states for 802.11 authentication and association. The 802.11 association is established prior to the security association (SA). Figure 5.2 shows the 802.11 states and transitions. The access point (AP) stores a separate state for every corresponding station (STA). Every STA stores a state for the AP it communicates with. The state is represented by two Boolean state variables, an authentication variable and an association variable. In State 1 both the variables are *false*. In State 2 the authentication variable is *true* and the association variable is *false*. In State 3 both variables are *true*.

Every frame type is categorized as Class 1, 2 or 3. Data frames are Class 3 frames. If an AP or STA is in State 1 and receives a Class 2 or Class 3 frame, then the recipient will discard the frame and respond with a Deauthentication notification with Reason Code 6 or 7. If a Class 3 frame is received while in State 2, then the recipient will discard the frame and respond with a Disassociation notification with Reason Code 7. The purpose of these responses is to resynchronize the protocol participants in the case of a state mismatch. The 802.11 standard does not clearly specify how to handle the situation where the AP is in State 3 and then receives an

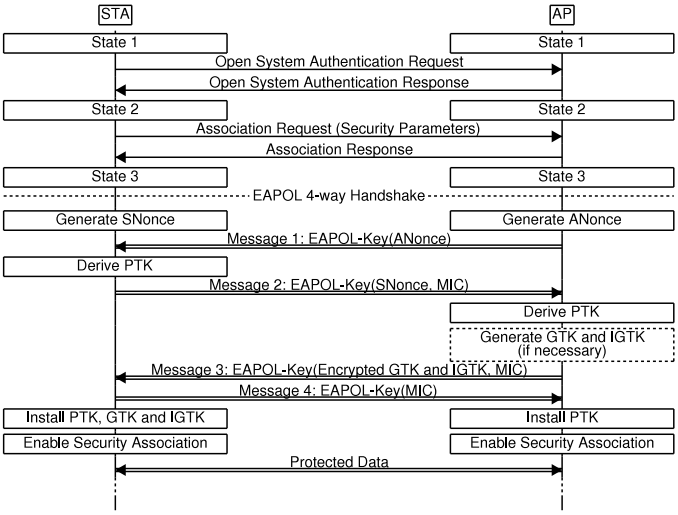

Figure 5.1: 802.11 with 802.11i and 802.11w connection establishment. A station (STA) connects to an access point (AP). The Open System Authentication exchange is a null authentication, present due to legacy reasons. A successful 802.11 association triggers an Extensible Authentication Protocol over local area networks (EAPOL) 4-way handshake that provides authenticated key agreement. The 4-way handshake creates a shared pairwise transient key (PTK) which is used to protect unicast traffic between the STA and AP. The group temporal key (GTK) and integrity GTK (IGTK) are used to protect broadcast data frames and management frames from the AP to the STA. A message integrity code (MIC) is used to protect EAPOL messages 2, 3 and 4. Message 1 is not protected.

Authentication Request. The textual description states that the AP should enter State 2, while the state transition diagram does not allow such a transition. Some implementations by Cisco follow the textual description and let the AP transition to State 2 [19]. Conversely, the open source hostapd [44] implementation leaves the AP in State 3. We model both of these behaviors since the 802.11 standard is open to interpretation on this point.

The IEEE 802.11 Task Group w (TGw) started developing the 802.11w amendment in 2005. In 2007, a TGw member discovered a deadlock vulnerability in the 802.11w draft [18]. Figure 5.3 illustrates an attack exploiting this vulnerability.

The presence of a deadlock vulnerability in 802.11w was considered unacceptable by TGw. An SA Query procedure was thus added to the 802.11w draft as a countermeasure. When an AP receives an Association Request from a STA, the AP does not immediately accept the Association Request if the STA is associated to the AP and there is a valid SA between them. The AP starts a timer and responds with an Association Response. The Association Response informs the STA that it must wait until the timer expires before it tries to associate again. The AP then sends a protected SA Query Request to the STA. If a protected SA Query Response is received, then the previous Association Request is ignored and the timer is canceled. If no response is received before the timer expires, then the AP will accept the next Association Request from that STA. The STA sends a new Association Request,

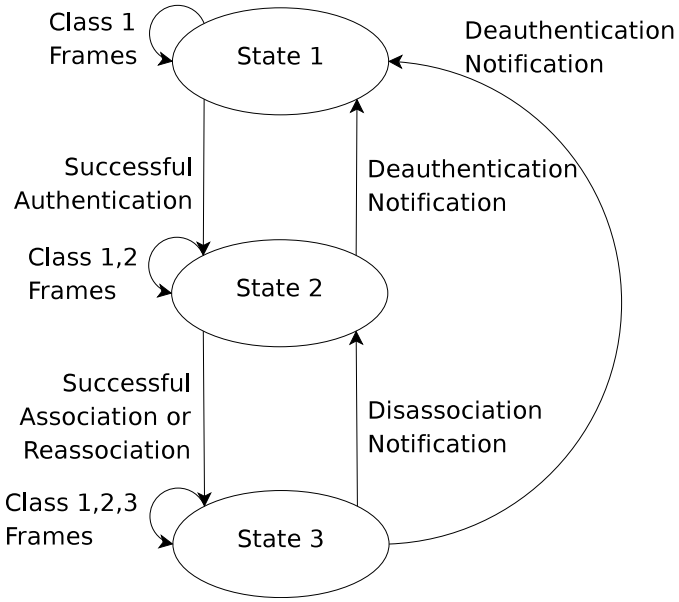

Figure 5.2: The 802.11 states and transitions. Note that the Deauthentication and Disassociation notifications are protected in 802.11w. If a recipient has an active security association and receives an unprotected Deauthentication or Disassociation frame, then it will silently discard that frame.

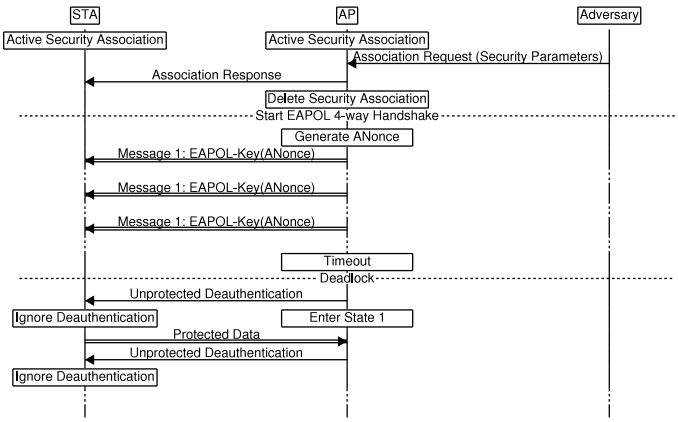

Figure 5.3: The 802.11w association attack discovered by TGw. The AP deletes its security association after a successful 802.11 association. The AP starts the EAPOL 4-way handshake, but gets no response from the STA. After a timeout, the AP enters State 1, where it discards all Class 2 and 3 frames. The AP has deleted its security association and PTK, and is thus not able to send a protected Deauthentication notification to the STA. Every unprotected Deauthentication notification from the AP is discarded by the STA.

and the AP sends an Association Response. Finally, the AP sends a protected Disassociation notification to the STA before it deletes its SA and proceeds to the EAPOL 4-way handshake.

In previous work, we described and analyzed this deadlock vulnerability and the SA Query procedure [19]. Part of the conclusions in [19] is that if an adversary is able to delete frames, then he can cause a deadlock even if the SA Query procedure is used. Ahmad and Tadakamadla later improved the association attack [57]. They proposed to use a virtual jamming attack [5] to prevent the STA from responding to the SA Query Requests. They also provided experimental validation of their attack using the hostapd software [44] as the AP and wpa_supplicant [52] as the STA. One issue with their analysis is that they ignored that the AP will send a protected Disassociation notification before it deletes the SA, which is a mandatory requirement in 802.11w [30, p. 74]. The STA will delete its SA and enter State 2 when it receives the protected Disassociation notification, thus avoiding a deadlock. The reason why their attack worked in practice is that this feature is not yet implemented in hostapd. A comment in the hostapd source code states that the feature will be implemented. Thus, the attack proposed by Ahmad and Tadakamadla will not cause a deadlock against a standards compliant 802.11w implementation.

802.11w also specifies an *optional* deadlock recovery mechanism [30, p. 75]. If a STA has a valid SA and receives an unprotected Deauthentication or Disassociation frame with Reason Code 6 or 7 from the AP, then the STA may initiate an SA Query procedure. If the SA Query procedure fails, then the STA may delete its SA and enter State 1. We model 802.11w both with and without this mechanism, since it is an optional part of the specification.

5.3 Model Construction

We construct a model of 802.11 with the 802.11i and 802.11w amendments using the formal method proposed in [21]. Promela is used as the modeling language and the Spin model checker is used to verify the model properties [56]. We use an iterative method to develop our model. The method starts with a simple model, uses the model checker to verify the model properties, and then refines the model. We use a high level of abstraction in order to keep the model state space as small as possible.

As specified in [21], our model consists of three entities: the AP, the STA and the adversary. Each entity is defined as a finite state transducer. We use the 802.11 frames as the transducer input and output messages. The AP and STA entities are modeled according to the protocol specification. The AP and STA transducers are deterministic except for a few cases where protocol timeouts are modeled nondeterministically. The adversary transducer is completely nondeterministic. It selects a frame type randomly and transmits this frame to either the AP or STA. The adversary may transmit its frames at any time during model execution. The adversary is only allowed to transmit frames that are not protected by the security mechanisms. The total number of frames it can transmit is limited by an upper bound. Since a model checker enumerates every possible model state, it discovers every possible attack from the adversary.

We model the AP, STA and adversary entities as Promela **proctype** declarations. We construct the model so that it stops execution if the protocol is in a state where it is unable to recover. The model thus stops execution in the cases where the actual protocol executes an infinite loop. Since we can use Spin to verify the safety property "invalid endstates", this construction facilitates the discovery of deadlock vulnerabilities. By carefully balancing the transmission of Data frames, we achieve the desired model property: that the model stops execution if the protocol is unable to recover. The AP and STA do not transmit a Data frame unless they first receive a valid Data frame. There are two exceptions to this rule. First, the AP sends a Data frame once it receives a valid EAPOL message 4 at the end of the EAPOL 4-way handshake. Second, the STA sends a Data frame if it uses the wrong 802.11 channel and then receives a Beacon frame instructing it to switch back to the correct channel.

The entity state machines for the AP and STA consist of the 802.11 State, the 802.11 channel, the SA state, the SA Query state, the last frame sent and the last frame received. We use the last frame sent and the last frame received as part of the entity states because the 802.11 standard does not specify state transitions for every single frame. For example, consider the transition "Successful Authentication" in Figure 5.2. This transition consists of two frames: an Authentication Request from the STA to the AP, and an Authentication Response from the AP to the STA. If the STA does not change its state after sending the Authentication Request, then it immediately sends another Authentication Request. The AP and STA thus need to store these frames in order to behave correctly. Conversely, the adversary state is determined only by the number of frames that it has transmitted. The total number of frames that the adversary can transmit is limited by a configurable upper bound. The adversary process terminates once the upper bound is reached.

Our next challenge is to model protocol timeouts. If possible, we model protocol timeouts using internal signaling messages. Consider the case that the AP sends EAPOL message 1, but the STA does not expect this message. In our model, the STA knows that the message will be discarded, thus it can send an internal signaling message to the AP to trigger the AP timeout procedure. We use the same approach to handle SA Query timeouts. However, we cannot use this approach when we process frames that could have been sent by the adversary. We use the Promela **timeout** statement to handle these cases. The Promela timeout statement is a global Boolean variable that is true, i.e. executable, if and only if no other model statement is executable. In other words, the execution of a timeout statement implies that either a protocol deadlock or a protocol timeout has occurred. We place the timeout statement inside the STA proctype declaration, since all of the AP timeouts are handled using internal signaling messages. When the timeout statement is executed, we check if the protocol is in a state where the STA would experience a protocol timeout. If the STA experiences a protocol timeout, then it follows the timeout procedure. Otherwise, the model execution is halted. Finally, we allow our adversary to halt its execution at any time, and to resume execution once the timeout statement is executed. The rationale for this construction is that the adversary process is always executable until it terminates. If the adversary is not allowed to halt its execution, then we would not be able to discover deadlock vulnerabilities where a

protocol timeout occurs and the adversary then sends additional frames after the timeout.

We include the complete model source code in the Appendix. The Promela code in the Appendix can be used together with the model checker Spin to replicate our results or to investigate the effects of protocol modifications.

5.4 Model Verification

We use Spin version 6.0.1 with the iSpin version 1.0.4 interface to verify our model using the safety property "invalid endstates". The model checker finds no deadlocks when the optional deadlock recovery mechanism described in Section 5.2 is enabled. We disable this mechanism and repeat the model verification. The model checker then finds several attacks that cause a deadlock in 802.11w. We analyze the attacks and discover three underlying protocol vulnerabilities.

The first vulnerability is that the AP can be disconnected after sending EAPOL message 3. This results in a state where the STA has a valid SA and the AP does not have a valid SA. Figure 5.4 illustrates an attack against this vulnerability. The attack targets the EAPOL 4-way handshake used in the initial connection establishment. The adversary has to send its Deauthentication notification[1] before the STA sends EAPOL message 4. Furthermore, the attack only works if the STA activates its security association before it receives the Deauthentication notification from the AP. The attack cannot be mounted against an established connection. However, if a different attack is able to cause a state reset and a new connection establishment, then a combination of attacks can cause a deadlock against an established connection.

The second vulnerability is that if the AP is in 802.11 State 2, then an SA Query procedure will not be triggered when it receives an Association Request. The AP transitions to State 2 after a successful 802.11 authentication if the implementation follows the textual description in the 802.11 standard. Figure 5.5 illustrates an attack against the second vulnerability. The attack can be mounted against an established connection.

The third vulnerability is that frame deletion is possible through the use of Beacon frames with the Channel Switch Announcement element specified in 802.11h [41]. This type of Beacon frame can make the STA switch to a different channel. The STA does not receive any frames from the AP when it is on the wrong channel. Beacon frames are not protected by 802.11w, thus they can be forged by the adversary. Figure 5.6 illustrates an attack against this vulnerability. In theory, this attack only works against a STA with 802.11h spectrum management support operating in the 5 GHz band. However, the results in [40] show that the attack may also work against drivers that support 802.11h even when they operate in the 2.4 GHz band. The attack can be mounted against an established connection.

[1] A Disassociation notification could also be used, the end result would be the same.

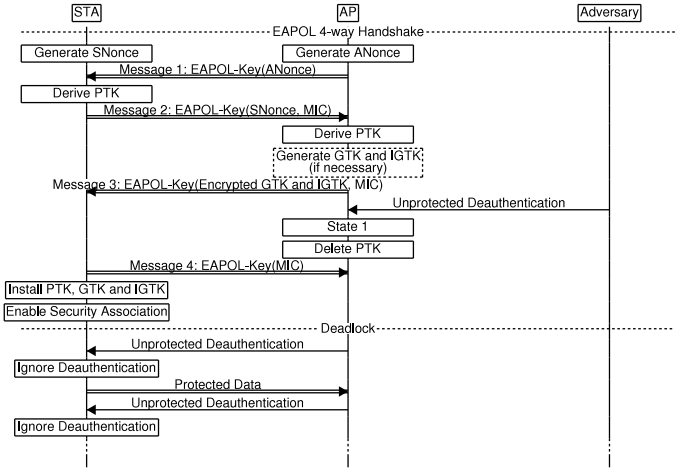

Figure 5.4: The first deadlock vulnerability in 802.11w. The adversary listens for an EAPOL 4-way handshake. Once the AP transmits EAPOL message 3, the adversary sends an unprotected Deauthentication notification to the AP. The AP thus deletes its derived PTK and transitions to State 1. The STA sends EAPOL message 4 to the AP, and the STA then installs the PTK and enables the security association. Since EAPOL message 4 is a Class 3 frame, the AP responds with an unprotected Deauthentication notification. This frame is discarded by the STA.

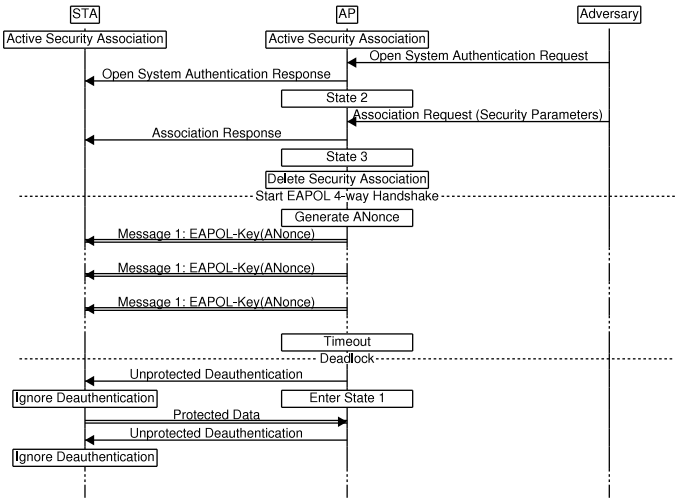

Figure 5.5: The second deadlock vulnerability in 802.11w. The adversary sends an Authentication Request to the AP, which transitions to State 2. The adversary then sends an Association Request to the AP, which then deletes its SA and initiates an EAPOL 4-way handshake. Once the 4-way handshake timeout occurs, the AP sends an unprotected Deauthentication notification to the STA. This frame is discarded by the STA.

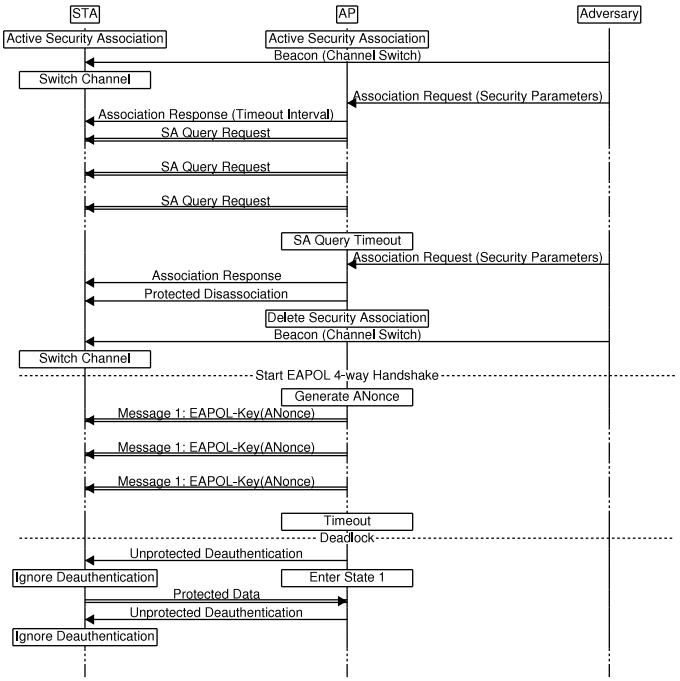

Figure 5.6: The third deadlock vulnerability in 802.11w. The AP, STA and adversary are on 802.11 channel A. The adversary broadcasts a Beacon frame with a Channel Switch Announcement element on channel A. The Beacon frame instructs the STA to switch to channel B. The adversary then transmits an Association Request to the AP, waits for the SA Query timeout, and then transmits another Association Request to the AP. Finally, the adversary broadcasts a Beacon frame with a Channel Switch Announcement element on channel B. The Beacon frame instructs the STA to switch to channel A. The STA does not receive any of the frames between the first and last Beacon frames, since it is on a different channel.

5.5 Experimental Validation

We implement the attacks described in Section 5.4. We then test the attacks against an 802.11w implementation in order to validate the model verification results and to determine if the attacks are practical.

We use the C programming language with the libraries from the aircrack-ng [27] tool suite as our implementation framework. Linksys WRT160NL wireless routers are used as the AP and STA, with the OpenWrt development version r27551 firmware [43]. Hostapd v0.8.x is installed on the AP and wpa_supplicant v0.8.x is installed on the STA. Both the AP and STA use the ath9k wireless driver in 802.11g mode (2.4 GHz band). The ath9k driver is currently the only driver that supports 802.11w with hostapd and wpa_supplicant. We use hostapd and wpa_supplicant because they provide an open source implementation of 802.11w. Therefore, we can modify their behavior in order to test different configurations and interpretations of the standard. We use a laptop with the Fedora Linux 14 operating system as the adversary. An Asus WL-167g wireless network card is used with the rt73usb driver in monitor mode to capture and inject 802.11 frames.

The first attack works as expected. The adversary waits until it receives EAPOL message 3, and then immediately transmits a Deauthentication frame to the AP. The Deauthentication frame consistently arrives earlier than EAPOL message 4 from the STA. Furthermore, the STA installs the PTK and activates the security association before it processes the Deauthentication frame from the AP. However, wpa_supplicant does implement the optional deadlock recovery mechanism in 802.11w. The result of the attack is that the STA initiates an SA Query procedure, reaches a timeout, and then resets its state and reconnects. We disable the recovery mechanism and repeat the experiment. The attack causes a protocol deadlock, which validates the formal protocol analysis results. In the following experiments the recovery mechanism is *disabled*.

The second attack should not work against hostapd, since its interpretation of the 802.11 standard is to leave the AP in State 3 after a successful authentication. We modify hostapd so that it transitions to State 2 after a successful authentication and mount our attack. The result is that the AP still performs the SA Query procedure, so the attack does not work as expected. We examine the hostapd source code and discover that the criteria for starting an SA Query procedure do not follow the 802.11w specification. In particular, hostapd does not check that the STA is associated before it starts the SA Query procedure. We modify hostapd to make it conform to the 802.11w specification. With our modification, hostapd only starts an SA Query procedure if the STA is already associated. We then repeat the experiment, and the attack causes a protocol deadlock.

The third attack does not work as expected. The AP and STA are initially on 802.11 channel 11. The first Beacon frame from the adversary instructs the STA to switch to channel 1. The STA performs the channel switch, but it resets and reconnects after a short while. The cause of this behavior is that the STA does not receive any Beacon frames on channel 1, thus it assumes that the connection is lost. We modify our adversary to switch to channel 1 and send a Beacon frame every 100 ms while it waits for the SA Query timeout. We then repeat the experiment, and the attack causes a protocol deadlock.

Finally, due to the fact that the STA resets and reconnects after a channel switch, we investigate a combination of the first and third attacks. We use the Beacon frame from the third attack to reset the STA state, and then mount the first attack against the following connection establishment. The attack causes a protocol deadlock. The use of a Beacon frame with the Channel Switch Announcement element enables the first attack to work against an established connection.

We conclude that the deadlock attacks found during the formal protocol analysis described in Section 5.4 are practical, and that the model verification results are valid.

5.6 Discussion

802.11w is still vulnerable to deadlock attacks, even though the SA Query procedure is designed to prevent such attacks. We are able to find three deadlock vulnerabilities in 802.11w using formal methods. We then find attacks that exploit these vulnerabilities

and demonstrate that the attacks are practical. One consequence of our results is that the optional deadlock recovery mechanism specified in 802.11w should be a mandatory requirement. If the recovery mechanism is not implemented, then the protocol implementation is vulnerable to deadlock attacks.

However, our main contribution is not the discovery of deadlock vulnerabilities in 802.11w. Our main contribution is a demonstration of how formal protocol analysis can be used to find such vulnerabilities, and that a useful formal protocol model can be constructed with a modest amount of resources. If the 802.11w amendment had been formally analyzed during development, then the deadlock vulnerabilities could have been detected and corrected before publication. Correcting protocol vulnerabilities during the design stage requires far less time and resources than correcting them after the standard has been approved and implemented. A common objection to the use of formal methods is that it requires too much time and resources. We give a counterexample to this objection by showing that a simple model can provide useful and significant results.

Another advantage of constructing a formal protocol model during the design process is that the model is an unambiguous and precise protocol specification. As discussed in Section 5.2, the 802.11 standard is open to interpretation on several points due to vagueness and ambiguities. This can cause interoperability problems and vulnerabilities. Furthermore, our experience is that the construction of a formal protocol model gives the protocol designers a better understanding of the protocol details.

Our model is not complete, since it only covers a subset of all frame types defined in the 802.11 standard. Thus, the model cannot be used to verify the *absence* of deadlock vulnerabilities in its current form.

The model source code [59], included in the Appendix, can be used to replicate and extend our results. It can also be used as the basis for a more comprehensive model of 802.11w or as a template for the construction of other protocol models. The protocol model can be extended with the cost model proposed in [21]. This extension would enable the discovery of less severe semantic DoS attacks that do not cause a protocol deadlock.

5.7 Conclusions

We have discovered and validated three new deadlock vulnerabilities in 802.11w with the help of formal methods. The results show that a simple protocol model can give useful and significant results, thus bridging the gap between theory and design. We hope that this work can contribute to a more widespread use of formal analysis during protocol design, and thus improve the robustness of wireless network protocols.

5.A Promela model of 802.11w

```
#define cs 5
#define att 4
/* cs = channel (queue) size
 * att = Adversary upper bound (number of frames)
 */

mtype = {authreq, authresp, assocreq, assocresp,
         deauth, disassoc, data, eapol1, eapol2,
         eapol3, eapol4, eaptime, csw, saqreq,
         saqres, saqtime, DUMMY};

/* Is optional deadlock recovery enabled? */
bool recovery = false;
/* Is 802.11h enabled? */
bool dot11h = true;
/* Transision to State 2 after Authentication? */
bool authstate2 = false;
/* Allow attacks during connection establishment */
bool setup = false;

/* Set to true once connection is established */
bool established = false;
bool timeoutflag = 0;

typedef Msg {
  mtype type;
  short class; /* 1, 2 or 3 */
  bool mic;    /* True if frame has valid MIC */
  short ch;    /* 802.11 channel (frequency) */
};

typedef State80211 {
  bool au;   /* 802.11 authentication state */
  bool as;   /* 802.11 association state */
  bool sa;   /* 802.11i RSN Security Association */
  short ch;  /* 802.11 channel (frequency) */
  mtype lm;  /* Last message sent */
  mtype lmr; /* Last message received */
};

/* Message channels (queues) */
chan toAP = [cs] of { Msg };
chan toSTA = [cs] of { Msg };

proctype AP() {
  /* SA Query state */
  bool saqtimeout = 0; bool saqactive = 0;
  Msg m; State80211 s;
  s.au = 0; s.as = 0; s.sa = 0; s.ch = 1;
  s.lm = DUMMY; s.lmr = DUMMY;
  do
    /* Start EAPOL 4-way handshake */
    :: atomic{
```

```
    (s.au && s.as && !s.sa && s.lm == assocresp) ->
    m.type = eapol1; m.class = 3; m.ch = s.ch;
    m.mic = 0; s.lm = m.type; toSTA ! m;}
  :: atomic{
    nempty(toAP) -> toAP ? m; s.lmr = m.type;}
     if
       /* Class 2 or 3 frame received in State 1 */
       :: atomic{
         (!s.au && !s.as && m.class != 1 &&
          m.ch == s.ch && m.type != saqreq) ->
         m.type = deauth; m.class = 1;
         m.ch = s.ch; m.mic = s.sa;
         s.lm = m.type; toSTA ! m; s.sa = 0;}
       /* Class 3 frame received in State 2 */
       :: atomic{
         (s.au && !s.as && m.class == 3 &&
          m.ch == s.ch && m.type != saqreq) ->
         m.type = disassoc; m.class = 2;
         m.ch = s.ch; m.mic = s.sa;
         s.lm = m.type; toSTA ! m;
         s.sa = 0;}
       /* Authentication Request */
       :: atomic{
         (m.type == authreq && m.ch == s.ch) ->
         m.type = authresp; m.class = 1;
         m.ch = s.ch; m.mic = 0;
         s.lm = m.type; toSTA ! m;
         s.au = 1;
         if
           /* Enter State 2 */
           :: (authstate2) -> s.as = 0;
           /* Do not change association status */
           :: (!authstate2) -> skip;
         fi}
       /* Association Request */
       :: atomic{
         (s.au && m.type == assocreq &&
          m.ch == s.ch) ->
         if
           :: (!s.sa || !s.as) ->
              m.type = assocresp;
              m.class = 2; m.ch = s.ch;
              m.mic = 0; s.lm = m.type;
              toSTA ! m; s.au = 1;
              s.as = 1; s.sa = 0;
           /* SA Query */
           :: (s.as && s.sa && !saqtimeout &&
               !saqactive) ->
              m.type = saqreq; m.class = 3;
              m.ch = s.ch; m.mic = 1;
              s.lm = m.type; toSTA ! m;
              saqactive = 1;
           :: saqactive -> skip;
           :: (s.as && s.sa && saqtimeout) ->
              m.type = disassoc; m.class = 2;
```

```
            m.mic = 1; s.lm = m.type;
            toSTA ! m; s.sa = 0;
            m.type = assocresp;
            m.class = 2; m.ch = s.ch;
            m.mic = 0; s.lm = m.type;
            toSTA ! m; saqtimeout = 0;
        fi
        }
    /* EAPOL message 2 */
    :: atomic{
        (s.au && s.as && !s.sa &&
         m.type == eapol2 &&
         m.ch == s.ch && m.mic) ->
        m.type = eapol3; m.class = 3;
        m.ch = s.ch; m.mic = 1;
        s.lm = m.type; toSTA ! m;}
    /* EAPOL message 4 */
    :: atomic{
        (s.au && s.as && !s.sa &&
         m.type == eapol4 &&
         m.ch == s.ch && m.mic) ->
        s.sa = 1; m.type = data;
        m.class = 3; m.ch = s.ch;
        m.mic = 1; s.lm = m.type;
        established = 1; toSTA ! m;}
    /* EAPOL 4-way handshake timeout */
    :: atomic{
        (m.type == eaptime) ->
        m.type = deauth; m.class = 1;
        m.mic = s.sa; s.lm = m.type;
        toSTA ! m; s.au = 0; s.as = 0;
        s.sa = 0;}
    /* Deauthentication processing */
    :: atomic{
        (m.type == deauth && m.ch == s.ch) ->
        if
          :: ((!s.sa && !m.mic) ||
              (s.sa && m.mic)) ->
             s.au = 0; s.as = 0; s.sa = 0;
          :: else -> skip;
        fi}
    /* Disassociation processing */
    :: atomic{
        (s.au && m.type == disassoc &&
         m.ch == s.ch) ->
        if
          :: ((!s.sa && !m.mic) ||
              (s.sa && m.mic)) ->
             s.as = 0; s.sa = 0;
          :: else -> skip;
        fi}
    /* SA Query Request */
    :: atomic{(m.type == saqreq && m.mic) ->
       if
         :: (s.au && s.as && s.sa &&
```

```
                m.ch == s.ch) ->
               m.type = saqres; m.class = 3;
               m.ch = s.ch; m.mic = 1;
               s.lm = m.type; toSTA ! m;
            /* STA SA Query timeout */
            :: else ->
               m.type = saqtime;
               s.lm = DUMMY; toSTA ! m;
          fi}
        /* Received SA Query Response */
        :: atomic{
          (s.au && s.as && s.sa &&
           m.type == saqres &&
           m.ch == s.ch && m.mic) ->
          saqtimeout = 0; saqactive = 0;}
        /* SA Query timeout */
        :: atomic{
          (m.type == saqtime) ->
          saqtimeout = 1; saqactive = 0;}
        /* Received data, send data */
        :: atomic{
          (s.au && s.as && s.sa &&
           m.type == data &&
           m.ch == s.ch && m.mic) ->
          m.type = data; m.class = 3;
          m.ch = s.ch; m.mic = 1;
          s.lm = m.type; toSTA ! m;}
        /* Invalid frame received, silently discard*/
        :: else -> skip;
      fi
  od
}

proctype STA() {
  Msg m; State80211 s;
  s.au = 0; s.as = 0; s.sa = 0; s.ch = 1;
  s.lm = DUMMY; s.lmr = DUMMY;
  do
    :: timeout -> timeoutflag = 1;
       if
         /* Authentication/association timeout */
         :: atomic{
           ((s.lm == authreq ||
             s.lm == assocreq) &&
            s.ch == 1 && s.lmr != saqreq) ->
           s.lm = DUMMY;}
         /* Timeout after association */
         :: atomic{
           (s.au && s.as && !s.sa
            && s.ch == 1) ->
           m.type = deauth; m.class = 1;
           m.ch = s.ch; m.mic = 0;
           s.lm = m.type; s.lmr = DUMMY;
           s.au = 0; s.as = 0;
           s.sa = 0; toAP ! m;}
```

```
    fi
/* Initiate authentication */
:: atomic{
  (!s.au && !s.as && s.lm != authreq) ->
  m.type = authreq; m.class = 1;
  m.ch = s.ch; m.mic = 0;
  s.lm = m.type; toAP ! m;}
/* Initiate association */
:: atomic{
  (s.au && !s.as && s.lm != assocreq) ->
  m.type = assocreq; m.class = 2;
  m.ch = s.ch; m.mic = 0;
  s.lm = m.type; toAP ! m;}
/* Reset if on wrong channel */
:: atomic{
  s.ch == 6 ->
  s.ch = 1; s.au = 0; s.as = 0; s.sa = 0;
  s.lm = DUMMY;}
:: atomic{
  nempty(toSTA) ->
  toSTA ? m; s.lmr = m.type;}
   if
     /* Class 2 or 3 frame received in State 1 */
     :: atomic{
       (!s.au && !s.as && m.class != 1 &&
        m.ch == s.ch && m.type != saqreq) ->
       m.type = deauth; m.class = 1;
       m.ch = s.ch; m.mic = s.sa;
       s.lm = m.type; toAP ! m;
       s.sa = 0;}
     /* Class 3 frame received in State 2 */
     :: atomic{
       (s.au && !s.as && m.class == 3 &&
        m.ch == s.ch && m.type != saqreq) ->
       m.type = disassoc; m.class = 2;
       m.ch = s.ch; m.mic = s.sa;
       s.lm = m.type; toAP ! m;
       s.sa = 0;}
     /* Channel switch */
     :: atomic{
       (m.type == csw) ->
       if
         :: (s.ch == 1) -> s.ch = 6;
         :: (s.ch == 6) -> s.ch = 1;
            m.type = data; m.class = 3;
            m.ch = s.ch; m.mic = s.sa;
            s.lm = m.type; toAP ! m;
      fi
      }
     /* Authentication Response */
     :: atomic{
       (!s.au && !s.as && s.lm == authreq &&
        m.type == authresp && m.ch == s.ch) ->
       m.type = assocreq; m.class = 2; m.ch = s.ch;
       m.mic = 0; s.lm = m.type; toAP ! m;
```

```
    s.au = 1; s.as = 0;}
  /* Association Response */
  :: atomic{
    (s.au && !s.as && s.lm == assocreq &&
     m.type == assocresp && m.ch == s.ch) ->
    m.type = DUMMY; s.au = 1; s.as = 1; s.sa = 0;}
  /* EAPOL message 1 */
  :: atomic{
    (s.au && s.as && m.type == eapol1) ->
    if
      :: (!s.sa && s.lm == assocreq &&
          m.ch == s.ch) ->
         m.type = eapol2; m.class = 3;
         m.ch = s.ch; m.mic = 1;
         s.lm = m.type; toAP ! m;
      /* EAPOL 4-way handshake timeout */
      :: else -> m.type = eaptime; toAP ! m;
                 s.lm = DUMMY;
    fi}
  /* EAPOL message 3 */
  :: atomic{
    (s.au && s.as && m.type == eapol3) ->
    if
      :: (!s.sa && s.lm == eapol2 &&
          m.ch == s.ch && m.mic) ->
         m.type = eapol4; m.class = 3;
         m.ch = s.ch; m.mic = 1;
         s.lm = m.type; toAP ! m; s.sa = 1;
      /* EAPOL 4-way handshake timeout */
      :: else -> m.type = eaptime; toAP ! m;
                 s.lm = DUMMY;
    fi}
/* Deauthentication */
:: atomic{(m.type == deauth &&
           m.ch == s.ch) ->
    if
      :: ((!s.sa && !m.mic) ||
          (s.sa && m.mic)) ->
         s.au = 0; s.as = 0;
         s.sa = 0; s.lm = DUMMY;
      :: (!recovery && s.sa && !m.mic) ->
         skip;
      :: (!s.sa && m.mic) -> skip;
      /* Start SA Query */
      :: (recovery && s.sa && !m.mic) ->
         m.type = saqreq; m.class = 3;
         m.ch = s.ch; m.mic = 1;
         s.lm = m.type; toAP ! m;
    fi}
/* Disassociation */
:: atomic{(s.au && m.type == disassoc &&
           m.ch == s.ch) ->
    if
      :: ((!s.sa && !m.mic) ||
          (s.sa && m.mic)) ->
```

```
                    s.as = 0; s.sa = 0;
                    s.lm = DUMMY;
                :: (!recovery && s.sa && !m.mic) ->
                    skip;
                :: (!s.sa && m.mic) -> skip;
                /* Start SA Query */
                :: (recovery && s.sa && !m.mic) ->
                    m.type = saqreq; m.class = 3;
                    m.ch = s.ch; m.mic = 1;
                    s.lm = m.type; toAP ! m;
             fi}
        /* SA Query Request */
        :: atomic{(m.type == saqreq && m.mic) ->
          if
            :: (s.au && s.as && s.sa &&
                m.ch == s.ch) ->
               m.type = saqres; m.class = 3;
               m.ch = s.ch; m.mic = 1;
               s.lm = m.type; toAP ! m;
            /* AP SA Query timeout */
            :: else -> m.type = saqtime;
                       s.lm = DUMMY;
                       toAP ! m;
          fi}
         /* STA SA Query timeout */
         :: atomic{
           (m.type == saqtime) -> s.au = 0;
           s.as = 0; s.sa = 0; s.lm = DUMMY;}
        /* Received data, send data */
        :: atomic{
          (s.au && s.as && s.sa && m.mic &&
           m.type == data && m.ch == s.ch) ->
          m.type = data; m.class = 3; m.ch = s.ch;
          m.mic = 1; s.lm = m.type; toAP ! m;}
        /* Invalid frame received, silently discard*/
        :: else -> skip;
      fi
  od
}

proctype Adversary() {
  short pkts = 0; /* Frames transmitted */
  Msg m; m.type = DUMMY; m.class = 1;
  m.ch = 1; m.mic = 0;
  do
  :: (pkts >= att) ->
       break;
  :: pkts < att  && (setup || established) ->
     if
       /* Exit before frame limit is reached */
       :: m.type == DUMMY -> break;
       :: m.type == DUMMY && (pkts > 0) ->
          if
            :: timeoutflag ->
               timeoutflag = 0;
```

```
            fi
        :: m.type == DUMMY ->
            /* Change attack frame type */
            if
              /* Deauthentication */
              :: m.type = deauth -> m.class = 1;
              /* Disassociation */
              :: m.type = disassoc -> m.class = 2;
              /* Authentication Request */
              :: m.type = authreq -> m.class = 1;
              /* Authentication Response */
              :: m.type = authresp -> m.class = 1;
              /* Association Request */
              :: m.type = assocreq -> m.class = 2;
              /* Association Response */
              :: m.type = assocresp -> m.class = 2;
              /* Channel Switch Announcement */
              :: dot11h -> m.type = csw; m.class = 1;
            fi
        :: m.type != DUMMY ->
            if
              /* Attack AP */
              ::atomic{pkts++; toAP ! m; m.type = DUMMY;}
              /* Attack STA */
              ::atomic{pkts++; toSTA ! m; m.type = DUMMY;}
            fi
      fi
  od
}

init {
  atomic{run AP();  run STA(); run Adversary();}
}
```

List of Acronyms

3G	third generation (mobile network)
4G	fourth generation (mobile network)
ACK	acknowledgment
AKA	authenticated key agreement
AKMP	Authentication and Key Management Protocol
AP	access point
ARP	Address Resolution Protocol
BIP	Broadcast/Multicast Integrity Protocol
BSS	basic service set
CBC-MAC	cipher-block chaining message authentication code
CCMP	CTR with CBC-MAC Protocol
COTS	commercial off the shelf
CRC	cyclic redundancy code
CS	carrier sense
CTR	counter mode
CTS	clear to send
DCF	distributed coordination function
DDoS	distributed denial of service
DHCP	Dynamic Host Configuration Protocol
DIFS	distributed (coordination function) interframe space
DoS	denial of service
EAP	Extensible Authentication Protocol

EAPOL Extensible Authentication Protocol over LANs

ERP-OFDM extended rate PHY using orthogonal frequency division multiplexing

ESS extended service set

GTK group temporal key

GTKSA group temporal key security association

ICMP Internet Control Message Protocol

ICV integrity check value

IE information element

IFS interframe space

IGTK Integrity GTK

IP Internet Protocol

ITS intelligent transportation systems

LAN local area network

LTE Long Term Evolution (4G)

LTL linear temporal logic

MAC medium access control

MFP Management Frame Protection

MIC message integrity code

NIC network interface card

PC point coordinator

PDU protocol data unit

PHY physical layer

PMK pairwise master key

PMKSA pairwise master key security association

PSK preshared key

PTK pairwise transient key

PTKSA pairwise transient key security association

QoS quality of service

RF	radio frequency
RSN	robust security network
SA	security association
SCADA	supervisory control and data acquisition
SIFS	short interframe space
SSID	service set identifier
STA	station
SYN	synchronize
TCP	Transmission Control Protocol
TGw	IEEE 802.11 Task Group w
TID	traffic identifier
TKIP	Temporal Key Integrity Protocol
TLA+	Temporal Logic of Actions specification language
TLC	TLA+ model checker
TSC	TKIP sequence counter
U-NII	Unlicensed National Information Infrastructure
WAN	wide area network
WEP	wired equivalent privacy
WLAN	wireless local area network
WMM	Wi-Fi Multimedia

Bibliography

[1] C. Wullems, K. Tham, J. Smith, and M. Looi, "A trivial denial of service attack on IEEE 802.11 direct sequence spread spectrum wireless LANs," in *Wireless Telecommunications Symposium, 2004*, may 2004, pp. 129 – 136.

[2] W. Eddy, *RFC 4987: TCP SYN Flooding Attacks and Common Mitigations*, 2007, http://tools.ietf.org/html/rfc4987.

[3] S. Tritilanunt, C. Boyd, E. Foo, and J. Gonzalez Nieto, "Toward non-parallelizable client puzzles," in *Cryptology and Network Security*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2007, vol. 4856, pp. 247–264.

[4] C. Meadows, "A formal framework and evaluation method for network denial of service," *IEEE Computer Security Foundations Workshop*, vol. 00, p. 4, 1999.

[5] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions," in *Proceedings of the 12th USENIX Security Symposium*. Berkeley, CA, USA: USENIX Association, 2003.

[6] R. M. Needham, "Denial of service: an example," *Commun. ACM*, vol. 37, no. 11, pp. 42–46, 1994.

[7] D. Dolev and A. C. Yao, "On the security of public key protocols," Stanford, CA, USA, Tech. Rep., 1981.

[8] J. Leiwo, T. Aura, and P. Nikander, "Towards network denial of service resistant protocols," in *Proceedings of the IFIP TC11 Fifteenth Annual Working Conference on Information Security for Global Information Infrastructures*. Kluwer, B.V., 2000, pp. 301–310.

[9] C. Meadows, "Formal methods for cryptographic protocol analysis: Emerging issues and trends," *Selected Areas in Communications, IEEE Journal on*, vol. 21, no. 1, pp. 44–54, Jan 2003.

[10] V. Ramachandran, "Analyzing DoS-resistance of protocols using a cost-based framework," Yale University, Tech. Rep., 2002.

[11] J. Smith, *Denial of Service: Prevention, Modelling and Detection*, Brisbane, Australia, 2007, PhD Thesis, Queensland University of Technology.

[12] S. Tritilanunt, *Protocol engineering for protection against denial-of-service attacks*, Brisbane, Australia, 2009, PhD Thesis, Queensland University of Technology.

[13] A. Mahimkar and V. Shmatikov, "Game-based analysis of denial-of-service prevention protocols," in *Proceedings of the 18th IEEE workshop on Computer Security Foundations*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 287–301.

[14] M. Fallah, "A puzzle-based defense strategy against flooding attacks using game theory," *Dependable and Secure Computing, IEEE Transactions on*, vol. 7, no. 1, pp. 5 –19, 2010.

[15] S. Lafrance and J. Mullins, "Using admissible interference to detect denial of service vulnerabilities," in *Sixth International Workshop in Formal Methods. Electronic Workshops in Computing (eWiC) by British Computer Society (BCS)*, 2003, pp. 1–19.

[16] K. Pelechrinis, M. Iliofotou, and V. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1 –13, 2010.

[17] P. Narayana, R. Chen, Y. Zhao, Y. Chen, Z. Fu, and H. Zhou, "Automatic vulnerability checking of IEEE 802.16 WiMAX protocols through TLA+," in *Secure Network Protocols, 2006. 2nd IEEE Workshop on*, 2006, pp. 44 –49.

[18] J. Epstein, *SA Teardown Protection for 802.11w, IEEE TGw DCN 2441, Rev 3*, 2007, https://mentor.ieee.org/802.11/file/07/11-07-2441-03-000w-sa-teardown-protection.ppt.

[19] M. Eian, "Fragility of the robust security network: 802.11 denial of service," in *Proceedings of the 7th International Conference on Applied Cryptography and Network Security*, ser. Lecture Notes in Computer Science, vol. 5536. Springer-Verlag, 2009, pp. 400–416.

[20] ——, "A practical cryptographic denial of service attack against 802.11i TKIP and CCMP," in *Proceedings of the Ninth International Conference on Cryptology And Network Security*, ser. Lecture Notes in Computer Science, vol. 6467. Springer-Verlag, 2010, pp. 62–75.

[21] M. Eian and S. F. Mjølsnes, "The modeling and comparison of wireless network denial of service attacks," in *Proceedings of the 3rd ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handhelds*, ser. MobiHeld '11. New York, NY, USA: ACM, 2011, pp. 7:1–7:6.

[22] M. Eian and S. F. Mjølsnes, "A formal analysis of IEEE 802.11w deadlock vulnerabilities," in *Proceedings of the* 31^{st} *Annual IEEE International Conference on Computer Communications (IEEE INFOCOM 2012)*. IEEE, 2012.

[23] IEEE, *IEEE Std 802.11-1999, IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, New York, NY, USA, 1999.

[24] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," in *Proceedings of the 4th Annual Workshop on Selected Areas of Cryptography*, 2001, pp. 1–24.

[25] E. Tews, R.-P. Weinmann, and A. Pyshkin, "Breaking 104 bit WEP in less than 60 seconds," Cryptology ePrint Archive, Report 2007/120, 2007, http://eprint.iacr.org/.

[26] A. Bittau, M. Handley, and J. Lackey, "The final nail in WEP's coffin," in *SP '06: Proceedings of the 2006 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 386–400.

[27] C. Devine, T. d'Otreppe, and M. Beck, "Aircrack-ng," 2011, http://www.aircrack-ng.org.

[28] IEEE, *IEEE Std 802.11-2007, IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, New York, NY, USA, 2007.

[29] ——, *IEEE Std 802.11i-2004, IEEE 802.11-1999 Amendment 6: Medium Access Control (MAC) Security Enhancements*, New York, NY, USA, 2004.

[30] ——, *IEEE Std 802.11w-2009, IEEE 802.11-2007 Amendment 4: Protected Management Frames*, New York, NY, USA, 2009.

[31] J. Epstein, *SA Teardown Protection, IEEE TGw DCN 2461, Rev 8*, 2007, https://mentor.ieee.org/802.11/file/07/11-07-2461-08-000w-sa-teardown-protection-text.doc.

[32] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, *RFC 3748: Extensible Authentication Protocol (EAP)*, 2004, http://tools.ietf.org/html/rfc3748.

[33] IEEE, *IEEE Std 802.11X-2004*, New York, NY, USA, 2004.

[34] Wireshark, http://www.wireshark.org.

[35] M. D. Aime, G. Calandriello, and A. Lioy, "Dependability in wireless networks: Can we rely on WiFi?" *IEEE Security and Privacy*, vol. 5, no. 1, pp. 23–29, 2007.

[36] S. Glass and V. Muthukkumarasamy, "A study of the TKIP cryptographic DoS attack," in *ICON 2007: Proceedings of the 15th IEEE International Conference on Networks*. New York, NY, USA: IEEE, 2007, pp. 59–65.

[37] E. Tews and M. Beck, "Practical attacks against WEP and WPA," in *WiSec '09: Proceedings of the second ACM conference on Wireless network security*. New York, NY, USA: ACM, 2009, pp. 79–86.

[38] IEEE, *IEEE Std 802.11e-2005*, New York, NY, USA, 2005.

[39] F. M. Halvorsen, O. Haugen, M. Eian, and S. F. Mjølsnes, "An improved attack on TKIP," in *NordSec '09: Proceedings of the 14th Nordic Conference on Secure IT Systems*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 120–132.

[40] B. Könings, F. Schaub, F. Kargl, and S. Dietzel, "Channel switch and quiet attack: New DoS attacks exploiting the 802.11 standard," in *LCN 2009: Proceedings of the IEEE 34th Conference on Local Computer Networks*, 2009, pp. 14–21.

[41] IEEE, *IEEE Std 802.11h-2003, IEEE 802.11-1999 Amendment 5: Spectrum and Transmit Power Management Extensions in the 5 GHz band in Europe*, New York, NY, USA, 2003.

[42] D. Harkins, *Attacks against Michael and Their Countermeasures. IEEE 802.11 Working Group Document 03/211r0*, New York, NY, USA, 2003. [Online]. Available: http://web.archive.org/web/20051226133200/http://www.ieee802.org/11/Documents/DocumentHolder/3-211.zip

[43] The OpenWrt Project, "OpenWrt," 2011, http://www.openwrt.org.

[44] J. Malinen, "hostapd: IEEE 802.11 AP, IEEE 802.1X / WPA / WPA2 / EAP / RADIUS Authenticator," 2011, http://hostap.epitest.fi/hostapd.

[45] J. Zarate, "Tomato Firmware," 2009, http://www.polarcloud.com/tomato.

[46] D. C. Plummer, *RFC 826: An Ethernet Address Resolution Protocol*, 1982, http://tools.ietf.org/html/rfc826.

[47] R. Droms, *RFC 2131: Dynamic Host Configuration Protocol*, 1997, http://tools.ietf.org/html/rfc2131.

[48] Cisco Systems Inc., *Enterprise Mobility 4.1 Design Guide*, San Jose, CA, USA, 2009.

[49] T. Han, N. Zhang, K. Liu, B. Tang, and Y. Liu, "Analysis of mobile WiMAX security: Vulnerabilities and solutions," in *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, 2008, pp. 828 –833.

[50] G. Kambourakis, C. Kolias, S. Gritzalis, and J. Hyuk-Park, "Signaling-oriented DoS attacks in UMTS networks," in *Advances in Information Security and Assurance*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2009, vol. 5576, pp. 280–289.

[51] C. Sankaran, "Network access security in next-generation 3GPP systems: A tutorial," *Communications Magazine, IEEE*, vol. 47, no. 2, pp. 84 –91, 2009.

[52] J. Malinen, "Linux WPA/WPA2/IEEE 802.1X Supplicant," 2011, http://hostap.epitest.fi/wpa_supplicant.

[53] IEEE, *IEEE Std 802.11g-2003*, New York, NY, USA, 2003.

[54] G. Holzmann, *Spin model checker, the: primer and reference manual*, 1st ed. Addison-Wesley Professional, 2003.

[55] IEEE, *IEEE Std 802.11r-2008*, New York, NY, USA, 2008.

[56] G. J. Holzmann, "The model checker SPIN," *IEEE Trans. Softw. Eng.*, vol. 23, pp. 279–295, May 1997.

[57] M. S. Ahmad and S. Tadakamadla, "Short paper: security evaluation of IEEE 802.11w specification," in *Proceedings of the fourth ACM conference on Wireless network security*, ser. WiSec '11. New York, NY, USA: ACM, 2011, pp. 53–58.

[58] J. Mitchell, A. Roy, P. Rowe, and A. Scedrov, "Analysis of EAP-GPSK authentication protocol," in *Proceedings of the 6th International Conference on Applied Cryptography and Network Security*, ser. Lecture Notes in Computer Science, vol. 5037. Springer-Verlag, 2008, pp. 309–327.

[59] M. Eian and S. F. Mjølsnes, "802.11w promela model," 2011, http://www.item.ntnu.no/∼eian/80211w.pml.