University of the Aegean

Doctoral Thesis

Intrusion Detection in Wireless Networks Using Nature Inspired Algorithms

Author:
Constantinos Kolias

Supervisor:
Assist. Prof. Georgios Kambourakis

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Laboratory of Information and Communication Systems Security
Department of Information and Communication Systems Engineering

June 30, 2014

Declaration of Authorship

I, Constantinos Kolias, declare that this thesis titled, 'Intrusion Detection in Wireless Networks Using Nature Inspired Algorithms' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Advising Committee of this Doctoral Thesis:

Assistant Professor, Georgios Kambourakis, Supervisor
Department of Information and Communication
Systems Engineering

Professor, Stefanos Gritzalis, Advisor
Department of Information and Communication
Systems Engineering

Assistant Professor, Panagiotis Rizomiliotis, Advisor
Department of Information and Communication
Systems Engineering

University of the Aegean, Greece
2014

Approved by the Examining Committee:

Stefanos Gritzalis
Professor, University of the Aegean, Greece

Aggelos Rouskas
Associate Professor, University of Piraeus, Greece

Vasilios Katos
Associate Professor, Democritus University of Thrace, Greece

Christos Xenakis
Assistant Professor, University of Piraeus, Greece

Panagiotis Rizomiliotis
Assistant Professor, University of the Aegean, Greece

Georgios Kambourakis
Assistant Professor, University of the Aegean, Greece

Emmanouil Maragkoudakis
Assistant Professor, University of the Aegean, Greece

University of the Aegean, Greece
2014

"It doesn't matter what you do...so long as you change something from the way it was before you touched it into something that's like you after you take your hands away."

Ray Bradbury, Fahrenheit 451

Abstract

Wireless networks are deployed with increasing pace throughout the globe in the last few years. However, due to the open nature of the wireless medium, they are inherently more susceptible to attacks than their wired counterparts. From GSM to UMTS and from WiFi to WiMAX, the security mechanisms implemented by these standards have all proven to be inadequate in terms of privacy and availability. In this context, alternative security mechanisms, such as Intrusion Detection Systems (IDS) have become a vital component of almost every wireless security infrastructure. Misuse Detection IDS may be the preferred choice of network administrators at the moment, due to its high detection, low false positive rate, however Anomaly Detection approaches quickly gain momentum.

Throughout this work particular effort is put into highlighting that current wireless technologies share similar traffic profiles and have comparable vulnerabilities, which are significantly diverse than the wired ones.

Driven by this fact, this doctoral thesis addresses the design of robust IDS, namely Termid, inspired by the protection processes observed in the natural systems, aiming to cover the unique requirements of the wireless realm.

In order to have a benchmark to evaluate the efficiency of the proposed approach, we design and offer publicly the Aegean Wireless Intrusion Detection (AWID) dataset. AWID is a database comprised of normal and attack traffic from the utilization of a real wireless network.

Indeed, the experimental results attest that the prototype intrusion detection mechanism introduced in the context of the current PhD thesis, namely Termid is able to identify wireless attacks, effectively and in a timely manner, before their disastrous results affect the network.

Greek Abstract

Τα ασύρματα δίκτυα αναπτύσσονται με ολοένα και αυξανόμενους ρυθμούς ανά την υφήλιο τα τελευταία χρόνια. Ωστόσο, λόγω της ιδιαίτερης φύσης του ασύρματου μέσου, είναι εγγενώς περισσότερο ευάλωτα σε επιθέσεις σε σχέση με τα αντίστοιχα ενσύρματα. Από το GSM ώς το UMTS και από το WiFi εως το WiMAX, οι μηχανισμοί ασφάλειας που αναπτύχθηκαν για το καθένα από αυτά τα πρότυπα, αποδείχθηκαν μη επαρκείς όσον αφορά την ιδιωτηκότητα και τη διαθεσιμότητα των προσφερόμενων υπηρεσιών. Έτσι, εναλλακτικοί μηχανισμοί ασφάλειας, όπως τα Συστήματα Ανίχνευσης Εισβολών (IDS) έχουν γίνει αναπόσπαστο κομμάτι όλων σχεδόν των ασύρματων υποδομών. Τα συστήματα ανίχνευσης εισβολών που βασίζονται την ανίχνευση κακής χρήσης (Misuse Detection IDS) είναι ίσως η προτιμητέα επιλογή αυτή τη στιγμή λόγω του συνδυασμού υψηλής απόδοσης/χαμηλών σφαλμάτων. Ωστόσο, οι προσεγγίσεις που βασίζονται στην ανίχνευση ανωμαλιών κερδίζουν συνεχώς έδαφος.

Στην παρούσα διατριβή προσπάθεια έχει καταβληθεί ώστε να καταστεί αντιληπτό οτι οι σύγχρονες ασύρματες τεχνολογίες έχουν κοινά χαρακτηριστικά όσον αφορά την κίνηση καθώς και παρόμοιες ευπάθειες, χαρακτηριστικά τα οποία διαφοροποιούνται σημαντικά σε σχέση με τα αντίστοιχα του ενσύρματου μέσου μετάδοσης.

Με βάση αυτές τις διαπιστώσεις η παρούσα διδακτορική διατριβή, συνεισφέρει το σχεδιασμό και την υλοποίηση ενός κατανεμημένου μηχανισμού ανίχνευσης εισβολών, με την ονομασία Termid, το οποίο έχει εμπνευστεί από μηχανισμούς που συναντώνται στη φύση.

Για την αξιολόγηση της αποδοτικότητας της προτεινόμενης λύσης σχεδιάσαμε και διαθέτουμε για δημόσια χρήση το Aegean Wireless Intrusion Detection (AWID) dataset, ένα σύνολο δεδομένων επαλήθευσης αποτελούμενο τόσο από φυσιολογική κίνηση δικτύου όσο και από κίνηση προϊόν επιθέσεων, που προέκυψε από τη χρήση ενός πραγματικού δικτύου 802.11.

Τα αποτελέσματα επιβεβαιώνουν ότι το πρότυπο σύστημα ανίχνευσης εισβολών, Termid, είναι ικανό να αναγνωρίσει επιθέσεις σε ασύρματα δίκτυα, με αποδοτικό και άμεσο τρόπο πριν ακόμη τα αποτελεσματά τους αποδειχθούν καταστροφικά.

Acknowledgements

My deep appreciation is hereby extended to the following who never ceased in providing valuable support and aid until the materialization of this thesis.

Firstly, my sincere gratitude goes to Assist. Prof. Georgios Kambourakis, who has not only been my supervisor, but also a friend to this long journey. Without his unwavering guidance and persistent advice the completion of this work would have been impossible.

I heartily thank to scientific and moral support during my research has not been only inspirational but also determinant in achieving my goals.

In addition, special thanks go to my mentor Prof. Stefanos Gritzalis, who contributed with crucial remarks on several stages of this work and Assoc. Prof. Aggelos Rouskas, Assoc. Prof. Vasilios Katos, Assist. Prof. Emmanouil Maragkoudakis, Assist. Prof. Panagiotis Rizomiliotis, Assist. Prof. Christos Xenakis, members of the examining committee for kindly investing some of their time to review this thesis and offer advices.

I consider myself privileged to be surrounded by dear friends such as, Dr. Dimitrios Damopoulos, Christos Anagnostopoulos, Panagiotis Ouranos, Mihail Papazoglou. Thank you for motivating me and your constantly providing constructive ideas.

A special thanks goes to my parents who not only provided financial, emotional and moral support, but also never ceased to encourage me to keep working hard.

Finally, my utmost admiration goes to my brother, Bill, whose familiarization with state-of-the-art technologies always excited my imagination.

Contents

List of Figures

List of Tables

Abbreviations

Wireless **F**idelity	**WiFi**
Universal **M**obile **T**elecommunications **S**ystem	**UMTS**
Long-**T**erm **E**volution	**LTE**
Wired **E**quivalent **P**rivacy	**WEP**
Intrusion **D**etection **S**ystems	**IDS**
Machine **L**earning	**ML**
Aegean **W**ireless **I**ntrusion **D**etection	**AWID**
IP **M**ultimedia **S**ubsystem	**IMS**
Small **O**ffice and **H**ome **O**ffice	**SOHO**
Wireless **L**ocal **A**rea **N**etwork	**WLAN**
WiFi **P**rotected **A**ccess	**WPA**
Access **P**oint	**AP**
Stations	**STA**
Request **T**o **S**end	**RTS**
Clear **T**o **S**end	**CTS**
Power **S**ave	**PS**
Quality **o**f **S**ervice	**QoS**
Frame **C**heck **S**equence	**FCS**
Initialization **V**ector	**IV**
Cyclic Redundancy **C**heck of **32** bit	**CRC-32**
Virtual **P**rivate **N**etworks	**VPN**
Temporal **K**ey **I**ntegrity **P**rotocol	**TKIP**
Advanced **E**ncryption **S**tandard	**AES**
Extensible **A**uthentication **P**rotocol	**EAP**
Pre-**S**hared **K**ey	**PSK**

Master **S**ession **K**ey	**MSK**
Pairwise **M**aster **K**ey	**PMK**
Pairwise **T**ransient **K**ey	**PTK**
Group **T**ransient **K**ey	**GTK**
Message **I**ntegrity **C**ode	**MIC**
Key **C**onfirmation **K**ey	**KCK**
Key **E**ncryption **K**ey	**KEK**
Group **E**ncryption **K**ey	**GEK**
Group **I**ntegrity **K**ey	**GIK**
Temporal **K**ey **I**ntegrity **P**rotocol	**TKIP**
Cipher Block **C**haining **M**essage **P**rotocol	**CCMP**
Wireless **R**obust **A**uthenticated **P**rotocol	**WRAP**
Offset **C**odebook	**OCB**
Message **I**ntegrity **C**ode	**MIC**
Cipher **B**lock **C**haining	**CBC**
Electronic **C**ode**b**ook	**ECB**
Robust **M**anagement **F**rames	**RMF**
Robust **S**ecurity **N**etwork **I**nformation **E**lements	**RSN IE**
Integrity **G**roup **T**ransient **K**ey	**IGTK**
Security **A**ssociation **Q**uery	**SA Query**
Timeout **I**nformation **E**lement	**TIE**
Data **O**ver **C**able **S**ervice **I**nterface, **B**aseline **P**lus **I**nterface	**DOCSIS BPI+**
Point-to-**M**ulti**P**oint	**PMP**
Base **S**tation	**BS**
Subscriber **S**tations	**SS**
Line-**o**f-**S**ight	**LOS**
Non-**L**ine-**o**f-**S**ight	**NLOS**
Scalable **O**rthogonal **F**requency **D**ivision **M**ultiplexing	**SOFDM**
Radio-**F**requency	**(RF)**
3rd **G**eneration **P**artnership **P**roject	**3GPP**
3rd **G**eneration	**3G**
Mobile **S**tation	**MS**

UMTS **T**errestrial **R**adio **A**ccess **N**etwork	**UTRAN**
Core **N**etwork	**CN**
User **E**quipment	**UE**
International **M**obile **E**quipment **I**dentity	**IMEI**
Radio **N**etwork **C**ontroller	**RNC**
Base **S**tation **C**ontroller	**BSC**
GSM/EDGE **R**adio **A**ccess **N**etwork	**GERAN**
Serving **G**PRS **S**upport **N**ode	**SGSN**
Mobile **S**witching **C**enter	**MSC**
Visitor **L**ocation **R**egister	**VLR**
Authentication **C**enter	**AuC**
Global **S**ystem for **M**obile **C**ommunications	**GSM**
Universal **I**ntegrated **C**ircuit **C**ard	**UICC**
Home **L**ocation **R**egister	**HLR**
Home **N**etwork	**HN**
Physical	**PHY**
Medium **A**ccess **C**ontrol	**MAC**
Convergence **S**ub-layer	**CS**
Common **P**art **S**ub-layer	**CPS**
Service **D**ata **U**nits	**SDU**
Mobile **S**tation	**MS**
Base **S**tation	**BS**
Downlink	**DL**
DL-Medium **A**ccess **P**rotocol	**DLMA**)
Downlink **C**hannel **D**escriptor	**DCD**
Uplink **C**hannel **D**escriptor	**UCD**
Transmission **O**pportunities	**TO**
Single **C**arrier	**SC**
Backo**ff**	**BO**
Truncated **B**inary **E**xponent **B**ackoff	**TBEB**
Connection **Id**entifier	**CID**
SS **B**asic **C**apability	**SBC**
Dynamic **H**ost **C**onfiguration **P**rotocol	**DHCP**

Trivial **F**ile **T**ransfer **P**rotocol	**TFTP**
Type-**L**ength-**V**alue	**TLV**
TFTP Complete	**TFTP-CPLT**
Dynamic **S**ervice **A**ddition	**DSA**
Privacy **K**ey **M**anagement	**PKM**
Authorization **K**ey	**AK**
Security **A**ssociation **Id**entifiers	**SAID**
pre-Primary **A**uthentication **K**ey	**pre-PAK**
Extensible **A**uthentication **P**rotocol	**EAP**
Master **S**ession **K**ey	**MSK**
Primary **A**uthentication **K**ey	**PAK**
EAP **I**ntegrity **K**ey	**EIK**
Pairwise **M**aster **K**ey	**PMK**
Key **E**ncryption **K**ey	**KEK**
Message **A**uthentication **C**ode	**MAC**
Cipher-based **MAC**	**CMAC**
Hashed **MAC**	**HMAC**
Cipher-based **MAC Key** for **U**plink	**CMAC KEY U**
Cipher-based **MAC Key** for **D**ownlink	**CMAC KEY D**
Cipher **B**lock **C**hanning **I**nitialization **V**ector	**CBC IV**
Authentication and **K**ey **A**greement	**AKA**
Global **S**ystem for **M**obile Communications	**GSM**
Universal **I**ntegrated **C**ircuit **C**ard	**UICC**
Home **L**ocation **R**egister	**HLR**
Home **N**etwork	**HN**
User's **E**quipment	**UE**
Universal **S**ubscriber **I**dentity **M**odule	**USIM**
GPRS **S**upport **N**ode	**SGSN**
Serving **N**etwork	**SN**
Mobile **S**tation	**MS**
International **M**obile **S**ubscriber **I**dentity	**IMSI**
Temporary **M**obile **S**ubscriber **I**dentity	**TMSI**
Radio **A**ccess **N**etwork	**RAN**

Authentication **V**ectors	**AV**
Expected **Res**ponse	**XRES**
Radio **N**etwork **C**ontroller	**RNC**
Cipher **K**ey	**CK**
Integrity **K**ey	**IK**
Fluhrer **M**antin **S**hamir	**FMS**
Pyshkin **T**ews **W**einmann	**PTW**
Pseudo-**R**andom **G**eneration **A**lgorithm	**PRGA**
Integrity **C**heck **V**alue	**ICV**
Extended **S**ervice **S**et **Id**entification	**ESSID**
Add Block **A**cknowledgement	**ADDBA**
Basic **S**ervice **S**et **I**dentification	**BSSID**
Network **I**nterface Card	**NIC**
Directed **A**cyclic **G**raph	**DAG**
Support **V**ector **M**achine	**SVM**
Artificial **N**eural **N**etwork	**ANN**
Interquartile **R**ange	**IQR**
Swarm **I**ntelligence	**SI**
Ant **C**olony **O**ptimization	**ACO**
Simple **A**nt **C**olony **O**ptimization	**SACO**
Particle **S**warm **O**ptimization	**PSO**
Wavelet **N**eural **N**etwork	**WNN**
Gradient **D**escent	**GD**
Quantum **P**article **S**warm **O**ptimization	**QPSO**
Modified **Q**uantum **P**article **S**warm **O**ptimization	**MQPSO**
Conjugate **G**radient	**CG**
Radial **B**asis **F**unction **N**eural **N**etworks	**RBF**
Standard **P**article **S**warm **O**ptimization	**SPSO**
Binary **P**article **S**warm **O**ptimization	**BPSO**
Self **O**rganizing **M**aps	**SOM**
Dynamic **S**elf-**O**rganizing **M**aps	**DSOM**
Human **I**mmune **S**ystem	**HIS**
Artificial **I**mmune **S**ystem	**AIS**

Antigen **P**resenting **C**ells	**APC**
Unsupervised **N**iche **C**lustering	**UNC**
Long **T**erm **E**volution	**LTE**

Dedicated to my parents

Chapter 1

Introduction

Wireless networks have prevailed in the last few years, managing to unsettle the reign of the once almighty wired ecosystem [1]. Today, end-users demand high quality, omnipresent connectivity, for work, education or entertainment purposes. Wireless networks fulfil these needs as they provide reasonable cost/low effort, wireless connectedness. The proliferation of ultra portable and handheld devices along with the combined coverage radius of multiple wireless technologies, such as Bluetooth, Wireless Fidelity (WiFi), Universal Mobile Telecommunications System (UMTS), and Long-Term Evolution (LTE) finally enable the users to roam from their room to a highway, and constantly be on-line.

However, the high mobility and high flexibility that these settings offer, comes with the price of questionable security. The wireless medium is by nature, unquestionably more open than the wired one. Even thought, all wireless technologies incorporate some sort of embedded security mechanism to protect the communication of their peers, to date not a single case of wireless technology has been reported to be immune to security inefficiencies, in either a theoretical or practical level. A characteristic example is that of Wired Equivalent Privacy (WEP), the security mechanism supporting the first version of 802.11, which soon after its exposure to the public, was found to be vulnerable to numerous attacks, including the efficient calculation of its key. Usually, amendments applied as patches, have increased the security of these systems, at least temporarily. However, with the non-stopping increase of computational resources the attackers quickly gain ground.

Due to the above reasons, the necessity of external mechanisms of protection quickly became apparent. In this context, Intrusion Detection Systems (IDS) such as [2], [3], [4] and [5] provide means of identifying as well as responding to a threat in a timely fashion. Such systems recognize intrusions based on predetermined signatures of known attacks.

However, Machine Learning (ML) based wireless IDSs are always within the scope of researchers since they do not require pre-compiled (static) signatures of attacks like the misuse detection based ones [6] rather, they deduce them automatically. An even more desirable aspect of such systems is their capability of recognizing new, undocumented (zero-day)attacks as suspicious events.

Actually, the concept of intrusion detection is not new. The research in this field was ignited in the 80's with Anderson's paper namely, "Computer Security Threat Monitoring and Surveillance" [7]. Generally speaking, intrusion detection is the scientific field that involves all the mechanisms and methodologies which lead to identification of actions of assault against a system originating from a malicious entity outside or inside it. While there has been a great deal of research concerning the intrusion detection on wired networks, the wireless ones seem heavily neglected.

However, the extraordinary behaviour of wireless communications and the idiosyncrasies of mobile devices generates special requirements which the conventional detection techniques fail to fulfil. While seeking for an elegant solution one may rely on unconventional methods, e.g., the ones the nature follows for complex problem solving. It is true that so far nature-inspired models have been applied into various scientific fields with great success. Yet, it was only until recently that these algorithmic emulations of natural processes have been applied to intrusion detection. Motivated by this fact, as further explained in the next section, the PhD thesis at hand aims in proving the applicability of natural inspired algorithms to the problem of wireless intrusion detection.

1.1 Motivation

From the days of the first premature attempts to develop intrusion detection algorithms until now, the vast majority of techniques and integrated systems has been focused either on protocols for the wired realm or operating systems for desktops. Recognizing the emergence of wireless era, it was only until recently that several researches started exhibiting special interest for intrusion detection in the wireless networks or the mobile platforms [8], [9]. Actually, the related research on the field is surprisingly limited, and has put several problems in abeyance.

Indeed, the wireless and wired domains present drastically contrasting characteristics. On the one hand, the diversity introduced with the new hardware and software architecture has had an impact on the user behaviour and the profile of the information exchanged. On the other hand, the differences go even deeper, involving the transmission speeds and signalling overhead of the protocols themselves. The basic difference

however, has its roots in the very nature of the air medium. Wireless technologies are inherently open to eavesdropping.

Unfortunately, the existence of trustworthy datasets that will not only act as a type of guide for the IDS in their training phase, but also a reliable benchmark in their testing phase, is considered of paramount importance. The research in wireless intrusion detection is stalled partially due to the absence of a well-tailored benchmark. Actually, in practise, the available choices seem to be of poor quality even for the wired realm. As for the publicly available ones, in some cases, their contents are heavily tampered by their creators (e.g., in order to become anonymized), while in other cases, they are outdated or may even contain data that do not correspond to realistic conditions.

The essential differences between the wired and wireless networks make it near-impossible for IDS designed for the one to be applied directly to the other. Interestingly, biology inspired models and other approaches that emulate the nature's behaviour on solving complex problems, have been applied with success to a wide range of research fields, such as engineering, economics, biology, social science and lately intrusion detection. We argue that besides the positive feedback from traditional intrusion detection, these approaches have qualitative characteristics (e.g., high level of adaptability and parallelism) that theoretically make them remarkable candidates in the wireless intrusion detection domain.

The pivotal motivators of this work can be summarized as follows:

- The problem of intrusion detection in wireless networks has been explored sparsely despite the increasing interest of both academia and industry.
- Wireless network characteristics are drastically different from the wired ones.
- Wireless technologies seem to be similar and interconnected. Intrusion detection designed for a single type of wireless network may apply to several others with little modification.
- The lack of benchmark tools may deterrent further research on the field.
- The fact that traditional detection approaches behave poorly in the wireless realm.

1.2 Contributions

The main ambition of this PhD research is to verify the suitability of biology inspired algorithms, for the construction of IDS in the wireless realm. To this end, this research

contributes a novel prototype implementation of a nature inspired classifier, optimized for identifying MAC layer threats on 802.11 networks.

Also, a major contribution of this thesis is the development and public offering of a dataset containing traffic extracted from a real life wireless network, namely the AWID dataset. While this contains traffic extracted from an 802.11 WEP protected network, by generalisation, we anticipate that it will be proven an invaluable resource in the field of wireless intrusion detection in general.

The initial contribution of this work is a comprehensive survey of the known attacks against three different wireless technologies namely, the 802.11, 802.16 as well as UMTS networks.

A theoretical and empirical evaluation of the impact and feasibility of the most critical of these attacks is also included.

An important part of this doctoral thesis is the formulation of simple signatures of known attacks against 802.11 networks. Their characteristics and possible variations are analysed in detail. Their study, will lead to a better understanding of the nature of wireless attacks and inductively, to proper methods to effectively counter them.

The final offering of the thesis is an exhaustive review of anomaly detection with particular interest in nature inspired algorithms for intrusion detection. Genetic algorithms, evolutionary computation, artificial immune systems and swarm intelligence approaches found in literature were reviewed and compared with a goal of exploring their efficiency in IDS field.

More specifically, the contribution of this work with respect to publications in scientific journals and conference proceedings is as follows:

- Examination of signalling oriented attacks that can affect both UMTS and UMTS/WLAN integrated systems with special focus on DoS attacks[1][2]. Precisely, this examination includes:
  - A comprehensive overview of the existing attacks against UMTS networks.
  - A formulation of numerous undocumented theoretical attacks of this class plus protocol vulnerabilities that can be exploited to unleash such situations.

[1]Kambourakis, Georgios and Kolias, Constantinos and Gritzalis, Stefanos and Hyuk-Park, Jong. Signaling-oriented DoS attacks in UMTS networks. In Advances in Information Security and Assurance, pages 280-289. Springer, 2009.

[2]Kambourakis, Georgios and Kolias, Constantinos and Gritzalis, Stefanos and Park, Jong Hyuk. DoS attacks exploiting signaling in UMTS and IMS. Computer Communications, 34(3):226-235, 2011.

  - An overview of DoS attacks launched combinatorially against the core UMTS and IP Multimedia Subsystem (IMS) components.

- A holistic view of attacks and countermeasures found in the literature against the IEEE 802.16 family of standards[3]. The main pillars of contribution of this part of the thesis are:

  - An organization and classification of these attacks based on several criteria.
  - An evaluation of the presented attacks with respect to their threat level
  - An extensive assessment of the various remedies found in literature.

- A comprehensive analysis of the internal mechanisms of Swarm Intelligence-based (SI-based) IDS[4].

- A survey of attacks against several versions of the 802.11 standards[5].

- The construction of AWID, that is a dataset containing traffic extracted from a real-life wireless network, specifically designed to support the early stages of wireless IDS development[5].

- The formulation of signatures of known attacks against 802.11 networks based on the findings of the AWID dataset[5].

- The design and implementation of a classification algorithm, inspired from natural protection mechanisms, optimized for the detection of threats in the wireless realm[6].

1.3 Outline

In the next chapter we singled out three major wireless technologies, namely 802.11, 802.16, UMTS and presents their characteristics with special emphasis on the security infrastructure incorporated in each one.

For each one of these three use cases, the third, fourth and fifth chapter comprehensively surveys their vulnerabilities as well as the documented attacks.

[3]Kolias, Constantinos and Kambourakis, Georgios and Gritzalis, Stefanos. Attacks and countermeasures on 802.16: Analysis and assessment. Communications Surveys & Tutorials, IEEE, 15(1):487-514, 2013.

[4]Kolias, Constantinos and Kambourakis, Georgios and Maragoudakis, M. Swarm intelligence in intrusion detection: A survey. Computers & Security, 30(8):625-642, 2011.

[5]Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Stefanos Gritzalis. Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset. Submitted in Communications Surveys & Tutorials

[6]Kolias, Constantinos and Kambourakis, Georgios. "Termid: A Distributed Ant Colony Based IDS". Under Submision

Chapter 6 conducts a theoretical and empirical evaluation on some of the surveyed attacks presented in chapter 3, 4, and 5. In several cases, the respective tools for launching attacks described only in theoretical level to that date, have been be implemented. The aim of this part of the study was to extract a common denominator and highlight the possible similarities with respect to their methodologies and execution strategy.

Chapter 7 begins with an overview of anomaly detection in general, and moves on to describing the most important approaches of each kind.

Chapter 8 carries on with a focalised survey of nature inspired approaches in the field of intrusion detection. Moreover, several aspects of these approaches are discussed, including the possible challenges involved, their inefficiencies as well as their strong points. This chapter concludes by enumerating the theoretic advantages, the natural inspired approaches have over alternatives.

Chapter 9 begins by discussing the lack of a well-taylored dataset, to be used as a liable benchmark in the field of wireless intrusion detection. In the process, AWID, a dataset specifically designed to meet the special needs of the wireless terrain is presented. Its characteristics are provided in full detail along with the evaluation results of several experiments conducted upon it.

Based on the data contained in AWID, chapter 10 attempts to extract signatures of known attacks (specifically targeting the 802.11 standard). This contribution is expected to lead to a better understanding of the hidden characteristics of certain classes of wireless attacks, thus aiding the construction of more sophisticated signature based and anomaly based IDS.

Chapter 11 details on the structural characteristics of a novel, nature inspired algorithm to be used as the heart of an IDS targeting the wireless realm. Several experiments are conducted which prove that the proposed system is not only faster than the traditional ML approaches, but in the majority of the cases more accurate.

The final chapter completes this PhD thesis by commenting on the results of the conducted research. Additionally, it enumerates the available options on improving the speed and the reliability of nature inspired IDS. Directions for future work are also presented in the end of the chapter.

Chapter 2

Wireless Technologies and Security

In this chapter we analyse the basic structure and security mechanism of three of the most popular wireless technologies. The well established IEEE 802.11 commonly known as Wi-Fi, the IEEE 802.16 often referred to as WiMAX for wide area connectivity, and UMTS the standard technology for 3G wireless services in Europe. The goal was not to provide an exhaustive overview of the internal mechanisms of these protocols, rather to give the basic information required for understanding their possible vulnerabilities and documented attacks. The last two aspects will be discussed in detail in next chapters.

2.1 802.11 Architecture

The 802.11 family of networks are arguably the most popular choice for local area connectivity, as they provide low cost, low effort wireless connectedness. Such networks can be found virtually everywhere, in Small Office and Home Office (SOHO) settings, enterprise environments or even in ad-hoc situations where users simply want to establish "quick-and-dirty" connectivity for data exchange.

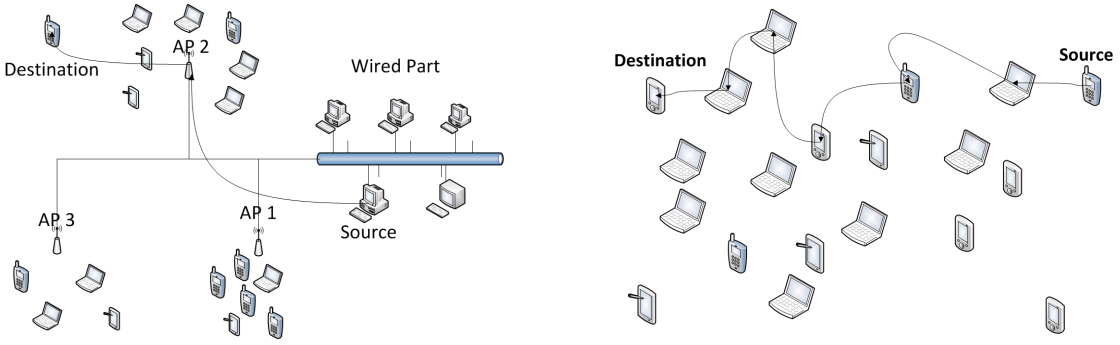
Since the first version of the 802.11 standard [10], dedicated security mechanisms have been incorporated to guarantee safe communication of all the peers in the Wireless Local Area Network (WLAN). Wired Equivalent Protection (WEP) was quickly found to be vulnerable not only to numerous availability attacks, but also to attacks that threat the secrecy of its key, thus quickly became deprecated. Posterior security additions such as WiFi Protected Access (WPA) and WPA2 proved to be more robust in matters of confidentiality, but with the ever-increasing computational power available to

anyone, even these security amendments, are expected to render themselves incapable of protecting even against brute force attacks. In any case, WPA/WPA2 share almost the same vulnerabilities as the early WEP versions as far as availability is concerned. Even the newest amendment, 802.11w [11], which concentrates in patching availability related shortcomings (leading to Deauthentication, Disassociation and Authentication Request attacks for example) has been proved impotent and is still unable to tackle the documented DoS attacks in its entirety [12], [13], [14], [15], [16].

2.1.1 Network Structure

The IEEE 802.11 family networks can be organised in either Infrastructure mode or Ad-Hoc mode. In the first paradigm, the basic organisational unit is a special piece of hardware, namely the Access Point (AP) to which the stations (STA) connect and through which the generated packets are transferred. The AP, in essence, is a transceiver and it connects to the wired counterpart of the network via an Ethernet cable.

On the contrary, in Ad-Hoc mode the STAs communicate with other STAs within their range in a direct way without the requirement for an AP. In this organisational paradigm the nodes of the network also play the role of the router. Generally, security and lack of infrastructure are two opposing forces in WLAN. By definition, Ad-Hoc WiFi networks are less secure than the Infrastructure based ones, but in such scenarios security is typically of secondary concern. These two areas of study have diverse vulnerabilities and the traffic behaviour is significantly dissimilar even under normal conditions.



(A) Infrastructure Mode (B) Ad-Hoc Mode

FIGURE 2.1: Typical Networks in Different Architecture Modes

2.1.2 Frame Types

Data exchange and network management in 802.11 is conducted with the transmission of frames. IEEE 802.11 defines three different types of frames namely, management

frames, control frame, and data frames. Each of them has different structure length and fields.

2.1.2.1 Management Frames

802.11 management frames allow STAs to establish communication with an AP and preserve connectivity with it. The structure of such frames varies depending on its purpose. Such frames can have one of the following subtypes:

- *Authentication* - Frames exchanged in rounds between the STA and the AP, with the purpose of identifying that the client (and the AP) is the claimed one. If successful, the client will enter the authenticated state but will not be granted full connectivity yet.
- *Deauthentication* - A frame sent from the AP to an STA when the network decides to terminate communication with that STA. Such frames may also be broadcasted to force all clients in the network, drop connectivity with the AP (this happens when an AP reboots or shuts down). Moreover, it is possible that the Deauthentication frames are sent from clients to the AP simply to notify about their intention to drop communication. Note that Deauthentication frames are not to be treated as requests and must always be accepted and acted upon.
- *Association Request* - A frame sent by an STA to an AP after successful authentication to request to move on to the associated state, i.e., obtain full connectivity. In this frame the STA reveals information about itself (e.g., supported data rates).
- *Association Response* - A frame sent by an AP to an STA as a response to an Association Request, containing the result of that attempt.
- *Reassociation Request* - A frame sent from an STA to an AP to request transition to full association state if previously had been degraded to the authenticated state. Such situations occur when an authenticated and associated STA roams from the serving AP to another AP of the same network.
- *Reassociation Response* - A frame sent by an AP to an STA as a response to a Reassociation Request frame.
- *Disassociation* - A frame transmitted either by an STA to the AP or by the AP to the STA with the purpose to terminate the association of the STA with the AP. Note that while the client may be disassociated with the AP, it still remains in authenticated state. Typically, such messages are sent when STAs are about to terminate all communication with the AP but wish to accomplish this gracefully.

- *Beacon* - A frame periodically broadcasted by an AP to announce its presence in the neighbourhood and advertise its capabilities. STAs continually scan all radio channels, in a quest for discovery of nearby APs and choose the appropriate AP after consulting its capabilities.
- *Probe Request* - A frame broadcasted by an unauthenticated client in search for a specific AP. If such messages don't specify an AP, they can immediately obtain information about all APs within that STA's range.
- *Probe Response* - After an AP receives a Probe Request frame it is obligated to immediately respond with a Probe Response frame containing details of the AP such as capability information, supported data rates, etc.

2.1.2.2 Control Frames

In 802.11 control frames coordinate access to the wireless medium and play a role in the delivery of data frames from an STA to the AP and vice-versa. A Control frame is constrained in one of the following subtypes:

- *Request to Send* - The RTS/CTS handshake mechanism is an optional element of 802.11 which aims in reducing frame collisions caused by the hidden terminal phenomenon. A Request to Send frame (RTS) is the first message of the handshake. If that mechanism is active the STA is required to send RTS frame to request permission to occupy the channel before transmitting an actual data frame.
- *Clear to Send* - A Clear to Send (CTS) is the second message of the RTS/CTS mechanism and it is send from the AP back to the requesting client as a response to a RTS frame. It specifies a time window during which only the requesting STA is permitted to occupy the channel and all other STAs must back off their transmission.
- *Acknowledgement* - A frame sent by an STA or AP to notify that a specific frame has been received successfully and without errors. If such frame is not received within a specific amount of time then the STA will have to retransmit the corresponding frame.
- *Power Save(PS) Poll* - A frame sent by an STA when it wakens from its power-saving mode to retrieve any frames buffered while in slumber.

2.1.2.3 Data Frames

Data frames are used for transmitting the actual information produced in the higher layers. Different types of data frames exist, based on whether they are sent on a contention based service, they carry additional information, or it has Quality of Service (QoS) enhancements. The basic data frame types are:

- *Data* - The basic frame type for sending and receiving data. These frames are transmitted during the contention-based period.
- *Null Data* - A special kind of data frame that carries no data payload. These frames are transmitted only from an STA towards the AP to state a change in its sleep state. This is accomplished simply by altering the value of the respective power management bit in the frame control field.
- *QoS Data* - An alternative data frame type for relaying data with higher priority.

2.1.3 Frame Structure

In 802.11 all data frames have the same structure which consists of a header, the frame body, and a Frame Check Sequence (FCS). The frame body consists of data which is usually encrypted. This field is the only one of variable length and can take up from 0 to 2,312 bytes. The FCS has length of 4 bytes. It is based on Cyclic Redundancy Check of 32 bit (CRC-32) algorithm and it is applied to bytes of both the header and the body. The header is the most complicated of the fields. It is 30 bytes long and on its own it is comprised of 7 fields: the composite 2 byte Control Frame, the 2 byte Duration, the 6 byte Address 1, Address 2 and Address 3 fields, the 2 byte Sequence Number field, and finally an additional Address field.

The structure of the typical Management frame is very similar to that of a data frame with the exception that frame body may only be comprised by fixed or variable length tagged parameters.

Control frames do not have a body or other variable length fields. Their header is smaller than the corresponding data or management frame headers, since besides the receiver and transmitter address fields it doesn't have any additional address fields (i.e. BSSID address). Moreover, it is deprived of the Sequence Number field. As made clear, the structure (size, fields) of 802.11 frames drastically defers among its types and subtypes. This dynamic nature of frames brings to surface the requirement for a static representation of these as vectors within a given dataset.

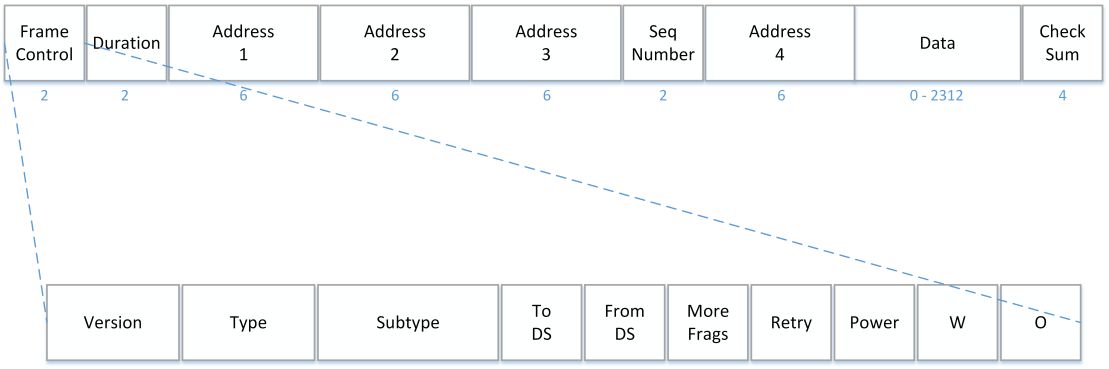


FIGURE 2.2: Structure of Header of 802.11 Fram

2.1.4 Security Procedures

2.1.4.1 WEP

Wired Equivalent Privacy (WEP) was deployed as the security mechanism of the first version of the 802.11 protocol. Its basic objective was to provide a confidentiality levels comparable to that offered in the wired world. However, as proved in practice, WEP failed and this protocol was found susceptible to a number of different attacks, including efficient cracking of its secret key. With the introduction of 802.11i, WEP became officially deprecated. Still, a large number of 802.11 networks base their protection solely on WEP.

WEP supports two methods of authentication namely, open system and shared key. In the first case, the client does not need to provide any credentials in order to connect to the AP and this process should not be considered as a real authentication. It is frequently used in conjunction with a white list of MAC addresses, or when the connection is meant to be totally open to everyone. On the other hand, shared key authentication completes after the exchange of four messages: (a) the client sends an Authentication Request message which contains the MAC address of the client and the MAC address of the AP, (b) the AP responds with a challenge message which contains a 128 bits random number, (c) the client sends a response message which contains the random number encrypted with the WEP shared key. The AP then decrypts the previous message using its shared key. If the number contained in the decrypted message matches the random number previously send, then the AP considers that the client is in possession of the shared key. To conclude this process, the AP responds with (d) an Authentication Response message containing the outcome of the authentication. It is clear that the authentication procedure described above is strictly unidirectional meaning the AP can authenticate the client but not vice-versa.

WEP relies on the RC4 [17] algorithm for confidentiality while the Cyclic Redudancy Check of 32bits (CRC-32) [18] mechanism is employed for message integrity. The entire confidentiality structure in WEP has been built around a static key also known as root key. WEP supports two different key sizes and as a result two versions exist, namely WEP-40 and WEP-104.

WEP-40 supports key sizes of 40 bits. As expected, this key is never used for direct packet encryption but is the basis (seed) for the generation of a session key. Only data frames are protected while management and control frames remain unguarded. Encryption of each packet is a multi-step process that consists of the following steps:

- A 24-bit long Initialization Vector (IV) is generated usually in a sequential way.
- Next, the root key is concatenated with the IV forming the per packet key. Note that while the root key is static the IV is different for each encryption attempt, therefore the per packet key is different for each packet.
- This key (which is a 64 bit sequence) is fed as a seed to the RC4 algorithm producing a key sequence which is known as keystream.
- As a final step, the keystream is XORed with the concatenation of the plaintext of the packet and its CRC-32 value. Thus, the ciphertext of the specific packet is generated. The encryption process on WEP-104 is analogous except for key size which in this case is 104 bits.

2.1.4.2 WPA

WiFi Protected Access (WPA) is a suite of security technologies that was introduced in 802.11x amendment, as a patch, in order to mitigate some of the weaknesses of WEP. Since the original security measures were found weak, in practice, even against attackers with moderate level of skills, many network administrators started taking privacy into their own hands by deploying third-party security solutions (e.g., 802.1X and Virtual Private Networks (VPN)). The lack of native, reliable wireless security mechanism triggered the development of 802.11i by the WiFi alliance and IEEE. WPA was treated as a transitional step since the more robust 802.11i (frequently referred as WPA2) security subprotocol was still under development. Actually, today WPA is a subset of 802.11i but it maintains forward compatibility with it.

WPA's security foundations lay in its stronger encryption mechanisms such as the Temporal Key Integrity Protocol (TKIP), or the Advanced Encryption Standard (AES)

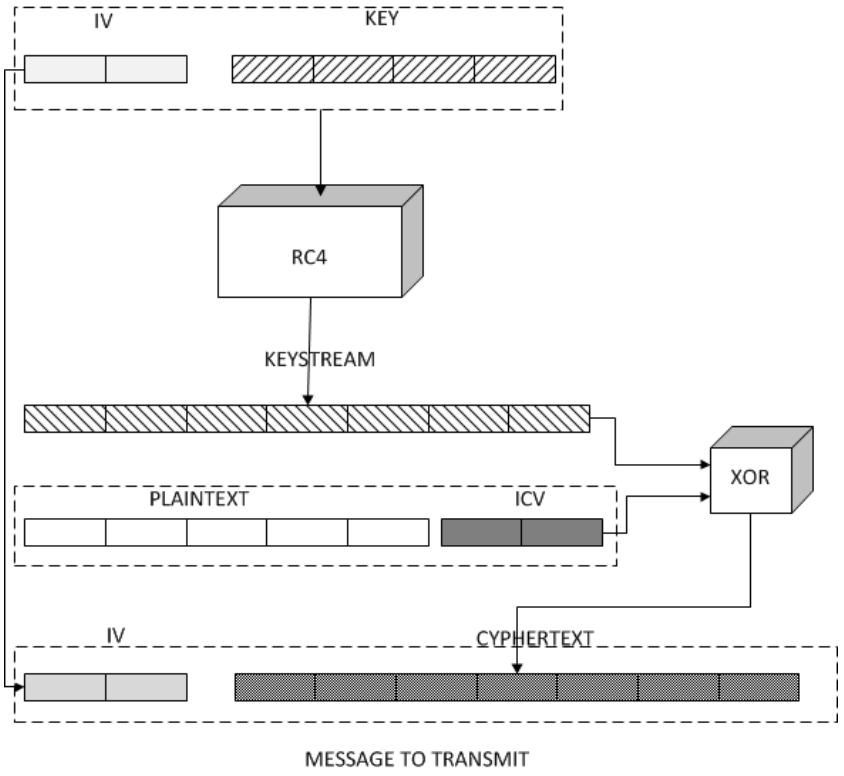


FIGURE 2.3: The Encryption Process on WEP

which is employed as an alternative. WPA effectively addresses critical issues such as mutual authentication, through the utilization of the 802.1X protocol and the Extensible Authentication Protocol (EAP), more appropriate IV lengths, Michael, a cryptographically robust integrity check mechanism, and a secure rekeying function among others. WPA depends on central servers such as RADIUS for user authentication, access control as well as management. While this practice is typically adopted in enterprise environments for home users, a variation of WPA, namely WPA Pre-Shared Key (WPA-PSK) is sufficient. In essence, WPA-PSK is a simplified version of WPA which is based on the use of a passphrase as a pre-shared secret key among the users similarly to the case of WEP.

2.1.4.3 WPA2

The IEEE 802.11i, also referred to as WPA2, was an amendment to the original IEEE 802.11 standard, incorporating the appropriate additions, to increase the security of the protocol. The final draft was ratified on June 24, 2004 and it was finally incorporated into the IEEE 802.11-2007 standard.

In WPA2 all keys derive from a single master key, placed in the highest level of the hierarchy. There are two types of that key which depend on the utilized method of

authentication. If the authentication method is based on a pre-shared key, the top key is simply the pre-shared key itself and it is referred as Pre Shared Key (PSK). If the authentication method is based on the 802.1X framework, the top key is called Master Session Key (MSK). These top level keys are used to generate the primary keying material in WPA2 which is the Pairwise Master Key (PMK). In the case of a pre-shared key based network the PMK is equal to the PSK, while in the 802.1X based network scenario the PMK is produced from a portion of MSK. The PMK is never used for encryption or integrity checks directly; rather it contributes to the generation of expendable keys. In the next level of the keying hierarchy exists the Pairwise Transient Key (PTK) and the Group Transient Key (GTK). These keys are produced during the authentication process with the STA and are unique for each client. These keys derive respectively from the PMK or the GMK as well as other random number negotiated with the client. The PTK key is then split into five subkeys, i.e., temporal encryption key, two temporal Message Integrity Code (MIC) keys, EAPOL-Key Key Confirmation Key (KCK), EAPOL-Key Encryption Key (KEK). These are the bottom level keys in the WPA2 hierarchy. The KCK and KEK are used to protect EAPOL-Key frames while the temporal key is used to encrypt/decrypt unicast network traffic. The GTK on the other hand, is split into two keys the Group Encryption Key (GEK) which is used for encrypting/decrypting multicast traffic and the Group Integrity Key (GIK) which is used for verifying the MIC of multicast/broadcast traffic.

As far as protection of the network traffic, WPA2 supports three alternative protocols. Temporal Key Integrity Protocol (TKIP), Counter-Mode/Cipher Block Chaining Message Authentication Code Protocol (CCMP), and Wireless Robust Authenticated Protocol (WRAP). The TKIP is based on RC4 encryption algorithm (which has been proven to be insecure) and is regarded as a transitional step from WEP. Actually, in comparison to WEP, it does provide a higher level of security and it is also backward compatible with it. WRAP on the other hand, is based on the Offset Codebook (OCB) mode of AES. This mode is considered more secure but may be subject to licensing issues, thus is no longer considered an actual choice. The most viable solution comes in the form of CCMP, which is based on the AES algorithm in its CCM mode. This cipher mode, breaks the plaintext in chunks of 128 bits and encrypts them with a key of the same size.

When CCMP is employed, the integrity of messages depends upon Message Integrity Code (MIC). MIC is constructed with the Cipher Block Chaining (CBC-MAC) which at the beginning it encrypts a nonce block, the source address and the packet number and then XORs the result with each succeeding block. The MIC is attached at the end of the plaintext and it is encrypted along with it.

The process of authentication is mutual in WPA2. That means that authentication guarantees that both the supplicant and the authenticator share the same PMK. Additionally, it ensures about the freshness of the peers, and contributes to a mutual agreement concerning the cipher suite to be used. In the end, this protocol generates a fresh PTK and with these keys, a secure communication channel is established for subsequent data transmissions. The authentication process consists of a series of messages, exchanged between the supplicant and the authenticator, commonly known as 4-way handshake. The first message is sent by the authenticator towards the supplicant. It consists of the MAC address of the authenticator, a nonce produced by the authenticator, the ID of the PMK, the current sequence number, and other bits that state that this message is the first one of the process. This message is not encrypted or integrity protected. Upon acceptance of the first message, the supplicant checks whether the sequence number contained in the message has been already used for that PMKSA. If not, the supplicant produces a nonce, creates the PTK, constructs the second message of the process, and transmits it to the authenticator. This message is integrity protected but it is not encrypted. In more detail, that message contains the MAC of the supplicant, the nonce the supplicant produced, the same sequence number contained in the first message, the security parameters agreed upon earlier stages of association, the MIC over the entire message, as well as bits informing about the type of the message. When this second message reaches the authenticator, it first checks that the sequence number is valid, then it creates the PTK itself, extracts the KCK and validates the correctness of the MIC. Finally, it verifies that the security parameters contained on that message are the same as the ones agreed during association. If all the checks pass, the authenticator moves to the generation of the GTK, its encryption with the KEK and it embodies it to the third message of the handshake.After receiving the third message, the supplicant checks that the sequence number is greater than the one contained in the first message, the nonce field matches the one received in the first message, then verifies that the cipher parameters are the same and finally that the MIC value is correct. The final message is also sent from the authenticator to the supplicant. It acts as a confirmation of the previous message. When this is received the supplicant is able to used the derived PTK and the received GTK for encrypting and decrypting unicast or multicast traffic.

2.1.4.4 802.11w

While the 802.11i focuses on confidentiality and integrity of the wireless communication, it has been proven rather thrifty on the availability. It is a fact that DoS attacks discovered even from the WEP apply in the WPA/WPA2 settings too without modification. The common denominator of most of these vulnerabilities is the fact that management

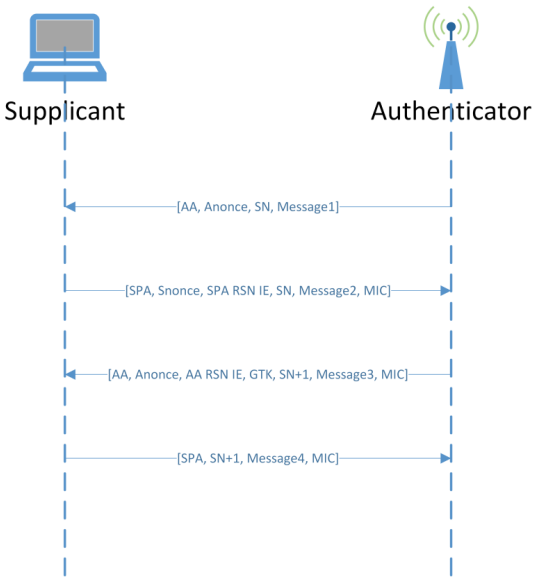


FIGURE 2.4: 4-Way Handshake Protocol

frames are unprotected thus easily issued even by unauthorized entities. For this reason, the 802.11w amendment, which was approved in 2009, focused on these issues and addressed them by introducing the Robust Management Frames (RMF) mechanism which is merely the cryptographically protected version of some of the management frames (Deauthentication, Disassociation, and Action management). In 802.11w the Robust Security Network Information Elements (RSN IE) field is extended by two bits (bits six and seven) to advertise the new capabilities which indicate that 802.11w is supported. More specifically, the sixth bit corresponds to the flag, Management Frame Protection Required while bit seven corresponds to the flag, Management Frame Protection Capable.

Unicast management frames are protected by the PTK, while for broadcast management frames, a new encryption key had to be introduced namely, the Integrity Group Transient Key (IGTK). The IGTK is used in a MIC information element. In further detail, the MIC is comprised of a packet ID, IGTK key ID, a serial number (IPN), and a cryptographic hash derived from a packet's MAC header and payload. IPN protects against replayed frames which are dropped if the IPN has been used in the past. To tackle Association Request attacks, the Security Association Query (SA Query) mechanism has been introduced. This mechanism makes use of two new management frames, namely SA Query Request, and SA Query Response which are exchanged between STA and AP as a follow up of every Association Request issued by the STA. The association procedure carries on, only if the SA Query Response message is verified by the AP. For the cases where an STA is already associated yet the AP receives a new Association Request message, the Timeout Information Element (TIE) is introduced. The AP replies

with a rejection notice and remains blindfolded to every Association Request from that client for a time interval equal to the one specified in the TIE field.

2.2 802.16 Architecture

WiMAX was partially based on the Data Over Cable Service Interface, Baseline Plus Interface (DOCSIS BPI+) protocol [19] which has been originally designed for cable modems. The first version of the standard, i.e., IEEE 802.16-2001 [20] only supported point-to-multipoint (PMP) fixed wireless access between a Base Station (BS) and several registered Subscriber Stations (SS). Since IEEE 802.16-2001 operates in the 10-66 GHz frequency range, this technology required line-of-sight (LOS) communication. The next version of the standard namely, IEEE 802.16-2004 [21] extended the frequency range into the 2-11 GHz band, thus enabling nonlineof-sight (NLOS) communication. Among other improvements in this version mesh mode was introduced. Until now, the most prominent version of the standard, namely IEEE 802.16e-2005 [22] specifies numerous major improvements including the full mobility support. Subscribers are now characterized as Mobile Stations MSs (in the following we use the terms MS and SS interchangeably). This became possible as the standard employs Scalable Orthogonal Frequency Division Multiplexing (SOFDM) in the physical layer. Additionally, it supports advanced security features such as mutual authentication for both the BS and MS. 802.16j-2009 [23] added support for multihop relays. Finally, the latest version of the standard, namely 802.16m-2011 [24] (also known as WiMAX release 2), added support for data rates as high as 100 Mbps for mobile nodes and 1 Gbps for stationary users.

2.2.1 Protocol Stack

The IEEE 802.16 protocol is organized primarily in the Physical (PHY) and the Medium Access Control (MAC) layers. The MAC layer can be further divided into three sublayers, namely the Service Specific Convergence Sub-layer (CS), the Common Part Sublayer (CPS) and the Security Sublayer. CS is the sub-layer that communicates with higher layers to acquire network data. In the process it transforms these data into MAC Service Data Units (SDUs). The format of the CS payload itself is CS depended. CPS provides basically the core MAC functionality being responsible for functions such as bandwidth allocation, connection establishment, and connection maintenance. The Security Sub-layer, addresses procedures such as authentication, authorization, key establishment, distribution and management. Also, it is responsible for encryption and

decryption of traffic passing from the PHY to the MAC layer and vice versa. The security mechanisms applied in the Security Sub-layer will be discussed in greater detail later in this section.

2.2.2 Network Entry

During the Initial network entry many critical parameters are negotiated between the Mobile Station (MS) and Base Station (BS). From a security point of view, the entire procedure is extremely receptive to violations since for most of its part the security measures contemplated by the specification have not taken place and important negotiation parameters are transmitted in cleartext. This section describes the basic steps that occur during the initial entry of an MS to the network. The overall procedure is summarized in Figure 2.5.

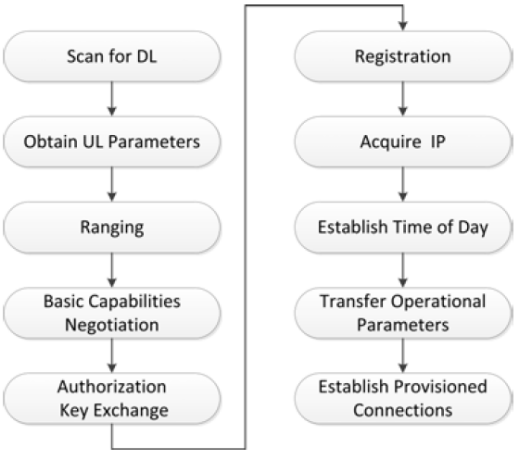


FIGURE 2.5: Initial Network Entry

Upon initial network entry (or after loss of signal) the first action an MS does is to acquire a Downlink (DL) channel. The MS shall begin scanning the DL frequency band for possible channels of operation until it finds a valid DL signal. This step ends once the PHY has achieved synchronization.

To do so, at least one DL-Medium Access Protocol (DL-MAP) message must be received by the MS. The DL-MAP informs the MS about the DL-Burst Profiles. The MAC remains in synchronization as long as it keeps receiving the DL-MAP and Downlink Channel Descriptor (DCD) messages for the channel. An MS may use the information contained in the DCD to determine if the channel corresponds to its needs. Also, the MS shall search for a Uplink Channel Descriptor (UCD) message from the BS (this is transmitted periodically to all the available UL channels) for retrieving the transmission parameters of a possible UL channel.

After the MS has obtained the UL and DL parameters it will attempt to acquire the correct timing offset and make power adjustments through the process of ranging. The MS shall use the information contained in the UL-MAP (or UCD) message to find an initial ranging interval. Usually, the BS allocates an initial ranging interval consisting of many Transmission Opportunities (TO). For Single Carrier (SC) and OFDM PHY, the MS shall construct an RNG-REQ message. Then the MS can transmit the RNG-REQ message in one of the known TO. Typically, there are only 3 TO in a 5 ms frame thus there is high probability of a collision to occur.

To reduce collisions the 802.16 specification dictates that the nodes should pass a period of inactivity of random duration known as Backoff (BO). If a collision occurs, the MS will eventually become aware of it since the corresponding RNG-RSP message will not arrive to the device within the expiration of the T3 timer (set to 200 ms by default). The collided nodes will attempt to resend the RNG-REQ message after a random waiting time but the waiting time interval will be doubled (until a maximum value is reached). This process will repeat (as long as MSs collide) up to a defined maximum number of retries. The aforementioned process is known as Truncated Binary Exponent Backoff (TBEB). Once the RNG-REQ message has been received by the BS, the latter will construct an RNG-RSP message and send it using the Initial Ranging Connection Identifier (CID). This message exchange shall result in the MS acquiring Basic and Primary Management CIDs as well as information about RF power level adjustment, offset frequency adjustment and timing offset corrections. Figure 2.6 presents the entire initial ranging procedure.

After ranging has successfully taken place, the MS will send to the BS an SS Basic Capability (SBC)-REQ message to inform it of its basic capabilities. The BS responds with an SBC-RSP message containing only the capabilities both the MS and BS can support. Upon successful capabilities negotiation, MS authorization and key exchange follows. The details of this procedure are analysed in greater depth in the next section.

Registration is the process that takes place after successful authorization. During this step the MS gets the Secondary Management CID. This means that the MS is actually granted entry into the network. This process involves the exchange of a pair of REG-REQ and REG-RSP messages. When both messages are successfully received, the BS will authorize the MS to forward traffic to the network.

Typically, the MS shall invoke Dynamic Host Configuration Protocol (DHCP) mechanisms for receiving all relative parameters, establishing IP connectivity and obtaining an IP address. The versions of the IP that are supported by the MS are indicated in the REG-REQ message with the default value being IPv4. Both the MS and BS need to be synchronized as the management system requires the current date and time for

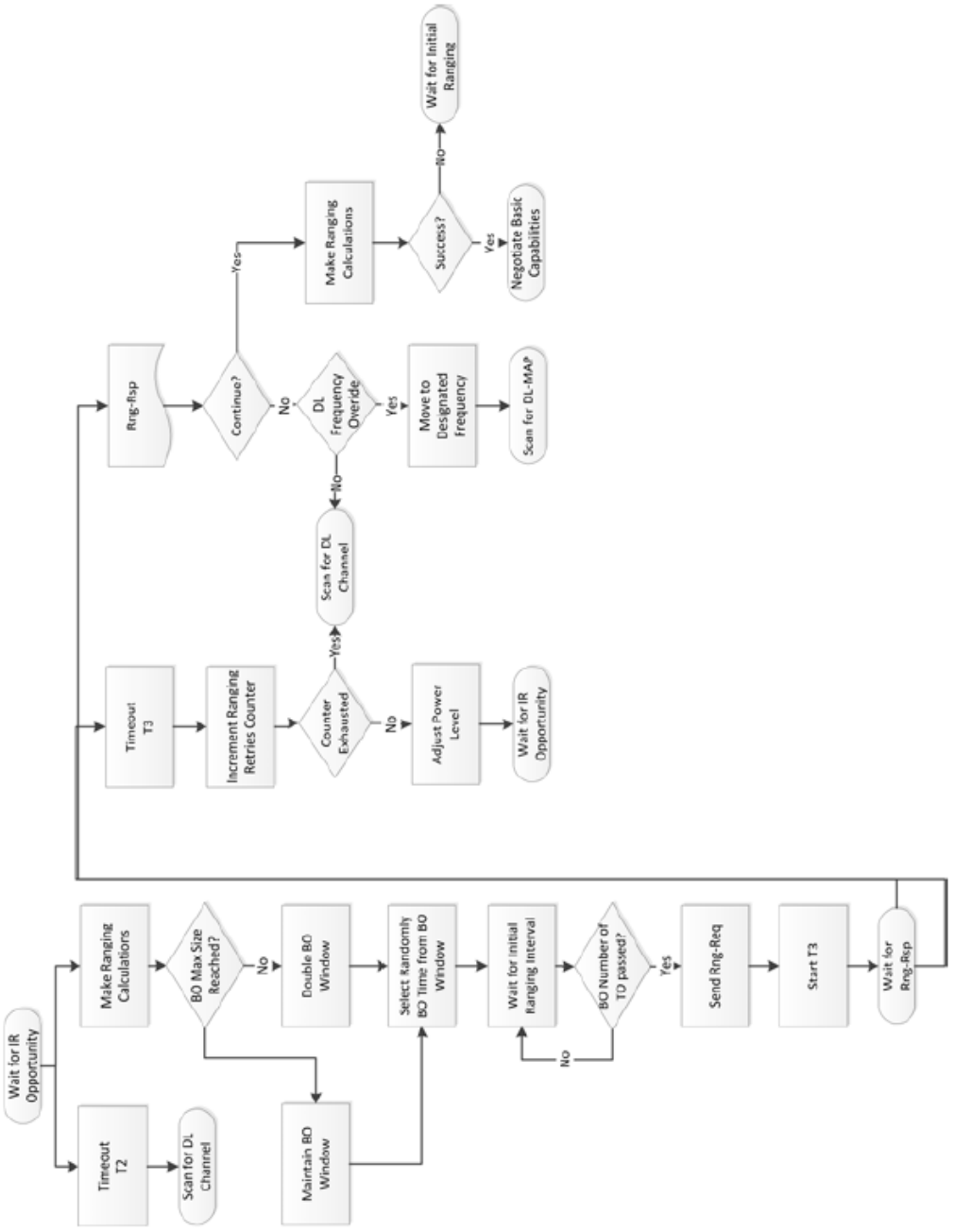


FIGURE 2.6: Flowchart of Initial Ranging

time-stamping logged events. For this reason, request and response messages are exchanged with a time server. The MS's secondary management connection is utilized for this process. This step is crucial for ongoing operation although not obligatory for a successful registration.

Next, the MS shall receive the MS Configuration File using Trivial File Transfer Protocol (TFTP) on the secondary management connection. This file consists of a number of configuration settings that are encoded in Type-Length-Value (TLV) format. The MS must notify the BS of successful receipt of this message by transmitting a Configuration File TFTP Complete (TFTP-CPLT) message on the primary management connection. As a final step, the BS shall send several Dynamic Service Addition (DSA)-REQ messages to the MS for creating new service flows. The MS responds with DSA-RSP messages.

2.2.3 Security Mechanisms

IEEE 802.16 relies on the Security Sub-layer to provide security to the end-subscriber and the network. This part of the protocol is where all the necessary cryptographic transformations are applied to the MAC PDUs. This is necessary to provide: (a) privacy, confidentiality and authentication to the subscribers, and (b) protection from theft of service to the service providers. The basic mechanism of security enforcement in 802.16 is the Privacy Key Management (PKM) protocol. Both 802.16-2009 and 802.16e-2005 support two versions of the PKM protocol. Mainly, PKM is responsible for authorization of subscribers and distribution of the keying material to the MS. Secondarily, it controls the application of the negotiated encryption algorithms to the data traffic. Actually, the PKM tasks can be divided into three distinct undertakings namely, authentication, key exchange, and encryption with a brief step for the key derivation that takes place in between the authorization and key exchange phases. These processes are described in further detail hereunder.

2.2.3.1 Authorization

The step of authorization happens first in the PKM protocol. The messages exchanged in this step differ for the two versions of the protocol. In PKMv1 the authorization process is initiated by the Authentication Information message which is sent by the MS to the BS. This message contains the MS manufacturer's X.509 certificate, (the manufacturer may have issued itself this certificate) and it is strictly informative. This message is then followed by an Authorization Request message sent again by the MS to the BS. This is comprised of the following information:

- The manufacturer-issued X.509 certificate of the MS
- a description of the cryptographic capabilities the MS supports
- the MSs Basic CID

The purpose of this message is to request an Authorization Key (AK), and at the same time be assigned with the Security Association Identifiers (SAID) (matching the corresponding Static SAs) that the client has the right to participate in. In response, the BS validates the MS's identity and determines the encryption algorithm (among the commonly supported ones), activates an AK for the MS, and constructs an Authorization Reply message. The latter is sent to the MS and it consists of the following fields:

- the active (for this particular MS) AK which is encrypted with that MSs public key,
- a 4-bit key sequence number used to distinguish between successive generations of AKs,
- the lifetime of this key, and
- the identities (i.e., the SAIDs) and properties of the Primary and Static SAs for which the MS is authorized to obtain keying information.

It is obvious that authentication in PKMv1 is one way, meaning that the BS can authenticate the MS but not viceversa. This introduces a vulnerability that soon became the cause of many attacks. As a result, PKMv2 was introduced and the authorization part is slightly modified to support mutual authentication. As in PKMv1, the second version of the PKM protocol dictates that the authorization process must start with the transmission of the informative message Authentication Information. This message is the same as in PKMv1. Following the Authentication Information the MS must issue an Authorization Request. The format of this message is modified in PKMv2. The Authorization Request includes:

- The manufacturer-issued X.509 certificate of the MS
- a description of the cryptographic algorithms supported by the MS
- the SSs Basic CID
- 64-bit random number generated by the MS.

The last field is the only new one added since the original PKM protocol to the Authorization Request message. In response, a BS sends back an Authorization Reply message. This message has been rectified more extensively and its fields now include:

- The BSs X.509 certificate; It is used to verify the BSs identity and to guarantee the authenticity of this message
- The pre-PAK key which is encrypted with the MSs public key; Only the owner of the corresponding private key will be able to decrypt it
- A 4-bit PAK sequence number, used to distinguish between successive generations of AKs
- The PAK lifetime
- the identities (i.e., the SAIDs) and properties of the SAs for which the MS is authorized to have keying material
- the 64-bit random number generated by the MS, originally contained in the Authorization Request message; This field is included to ensure that the Authorization Reply corresponds to the correct Authorization Request message

Additional fields are:

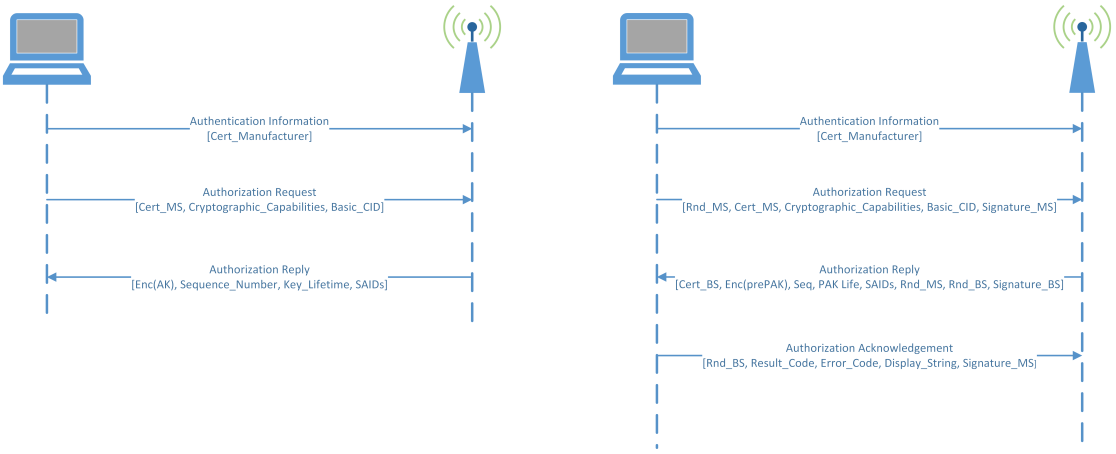
- a new 64-bit random number generated by the BS
- the RSA signature over the entire message

This allows the MS to verify that the BS is indeed the author of the Authorization Reply message. This process is performed using with the public key of the BS which is acquired by the certificate contained in the message. In other words, this extra field allows for mutual authentication. Following the message above the MS replies with an Authorization Acknowledgement message (or an Authentication Reject message in the case where the BS will reject the MS). The Authorization Acknowledgement includes:

- the 64-bit random number originally contained in the PKMv2 RSA Reply message
- an Authentication result code which can be “success” or “failure”
- an Error code which indicates the reason for the reject
- an optional Display string which includes a phrase for the reason for the reject rather than just a code number

- an RSA signature over the entire message

A comparison between the flow of messages in PKMv1 and PKMv2 is included in figure 2.7.



(A) Authorization Step of PKMv1 Protocol (B) Authorization Step of PKMv2

FIGURE 2.7: Message Flow in PKMv1 vs. PKMv2

2.2.3.2 Key Derivation

After the PKM authentication phase, normally the MS is in possession of some keying material. A small step where the MS derives the appropriate keying material takes place before the PKM can proceed to the next phase. The key derivation process is different between the two versions of the PKM protocol. Actually, the keys in 802.16 form a hierarchy. A key of a higher level is used to produce the key of the immediately lower level. All key generations in PKMv2 are produced using the Dot16KDF function. This function takes 3 arguments:

- A keying material of a higher level,
- A string used to alter the output of the algorithm
- A number used to indicate the length of the generated key.

More specifically, the RSA-based authorization process results in the creation of the pre-Primary Authentication Key (pre-PAK) while the Extensible Authentication Protocol (EAP) based authentication process produces the Master Session Key (MSK). These two keys constitute the basis of all other keying material and they are placed in the top of the key hierarchy. In RSA-based authorization a pre-PAK is used to generate the Primary Authentication Key (PAK). Optionally, the EAP Integrity Key (EIK) can

also be generated from the pre-PAK. EIK is used for transmitting authenticated EAP payload. In EAP-based authorization the 512 bits MSK, is simply truncated to 160 bits to derive the Pairwise Master Key (PMK). One of the PAK, PMK or both (according to the authentication method that was used) will be provided as input to the Dot16KDF function to produce the AK. The Key Encryption Key (KEK) is derived directly from the AK. Message Authentication Code (MAC) keys are used to sign management messages. This procedure is performed to guarantee the authenticity of these messages. The IEEE 802.16 supports two MAC modes namely Cipher-based MAC (CMAC) and Hashed MAC (HMAC). The one to be used is negotiated during the MS Basic Capabilities negotiation phase. Different MAC keys exist for UL and DL messages. The Cipher-based MAC Key for Uplink (CMAC KEY U) is used for signing messages in the uplink while the Cipher-based MAC Key for Downlink (CMAC KEY D) is used for the same purpose in the downlink. This only applies for the cipherbased MAC mode. For the hash-based MAC mode corresponding keys exist, i.e. HMAC KEY U, HMAC KEY D). In the case of HMAC these keys are derived directly from Dot16KDF function while in the case of CMAC and for versions later than 802.16e a corresponding prekey is generated first.

Also there are different keys for broadcast and unicast messages. In any case, MAC keys are derived directly from AK by simply using different string and key size arguments in the Dot16KDF function for each mode. Figure 2.8 depicts a diagram of the complete key derivation flow.

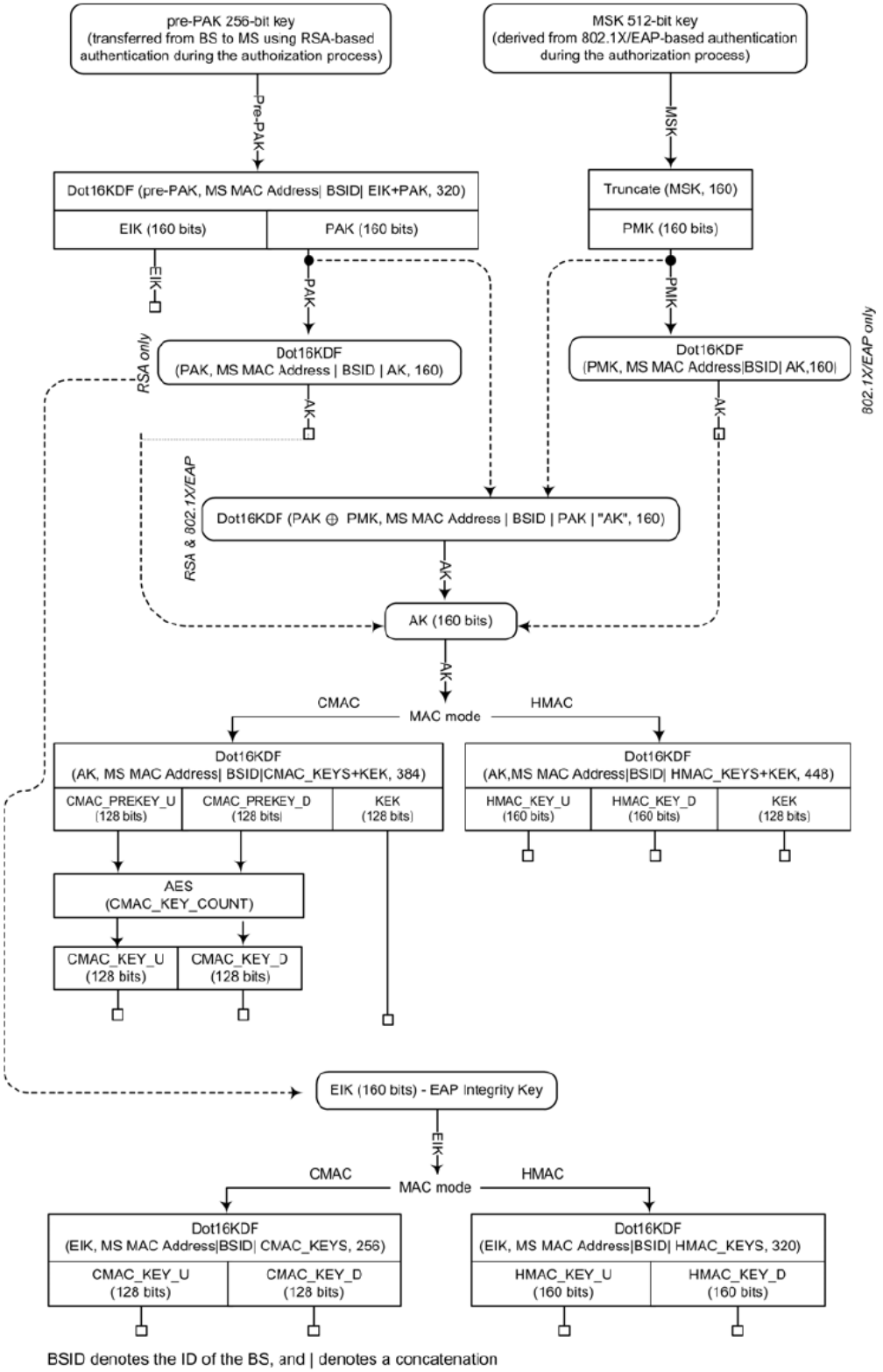


FIGURE 2.8: Complete Message Exchange and Key Derivation in 802.16

2.2.3.3 Handshake

The next phase is a three way handshake. It's main role is to confirm that both the MS and BS have indeed the correct AK from the previous procedure. Additionally, the handshake protocol takes care of secondary procedures such as key activation and SA parameters negotiation, security parameters confirmation etc. For this purpose the BS shall send an PKMv2 SA-TEK-Challenge which simply includes:

- a random number
- sequence number for the new AK
- the ID of the new AK
- Key Lifetime all protected by
- the HMAC/CMAC

The SS shall respond with a PKMv2 SA-TEK-Request to the BS. This message includes:

- the random number the MS received from the PKMv2 SA-TEK-Challenge
- a random number the MS produces,
- the sequence number of the new AK,
- the ID of the AK,
- the cryptographic suites supported by the MS
- the security capabilities of the MS
- the HMAC/CMAC of the entire message

Upon reception of PKMv2 SA-TEK-Request, a BS first confirms that the AK ID contained in the message refers to a valid AK and then it verifies the HMAC/CMAC. After that the BS will check if the random value sent matches the one contained in the PKMv2 SA-TEK-Request. If any of the three aforementioned tests fails the BS will simply ignore the message. Finally, the BS will make sure that the security capabilities encoded in the Security Negotiation Parameters attribute are the same with the security capabilities provided by the MS through the SBC-REQ message. If not, the BS should report the inconsistency to higher layers but might as well accept the message. If the validation of the PKMv2 SA-TEK-Request is successful, the BS shall send a PKMv2

SA-TEK-Response back to the SS. This message includes all the fields of the PKMv2 SA-TEK-Request message plus a TLV list of the SAs, their identifiers (SAID) any additional properties of the SA (e.g., type, cryptographic suite) that the SS is granted access to. The TEK-Parameters attribute in that list contains keying material such as the TEKs remaining key lifetime, its key sequence number and the Cipher Block Channing Initialization Vector (CBC IV). The HMAC/CMAC is the last field of this message.

2.2.3.4 TEK Transportation

As already mentioned, TEK is responsible for the encryption of traffic. The BS alone is in charge for the creation of this key, thus it must securely transmit it to the MS. The pair of PKM-REQ: Key Request and PKM-REP: Key Reply messages exist for this purpose. PKM-REQ is comprised of the following fields:

- The Key Sequence Number, which allows the BS to determine the AK used for the production of the corresponding UL HMAC/CMAC Key
- the ID of the SA whose TEK is requested
- the HMAC/CMAC digest over the entire PKM-REQ message payload

After verifying the authenticity of the message the BS responds with a PKM-REP message. The fields of this message are the following:

- Key Sequence Number
- SAID
- TEK-Parameters (Older)
- TEK Key Lifetime
- Key Sequence Number
- CBC-IV
- TEK-Parameters (Newer)
- TEK Key Lifetime
- Key Sequence Number
- CBC-IV,
- HMAC/CMAC digest over the entire message payload

It is to be noted that a unique state machine is maintained by the MS for each SAID contained in the PKM-RSP message. Each state machine is responsible for the initial establishment of TEK as well the periodic refreshing of those keys.

2.2.3.5 Traffic Encryption

After successful TEK exchange, both the MS and BS are able to encrypt/decrypt traffic, using this key. Note that the generic MAC header is not included in the encryption. Multiple encryption algorithms are supported. When DES algorithm in CBC mode is used, the CBC IV for the DL, shall be calculated by performing the XOR function to IV parameter included in the PKM-REP message and the current frame number. For the the UL, the CBC IV shall be calculated by performing the XOR function to IV parameter included in the PKM-REP message and the number of the frame where the relevant UL-MAP has been transmitted. If the AES algorithm in CCM mode is used then the MAC PDU payload shall always be prepended with a 4-byte packet number which will never be encrypted. Also, the MAC PDU shall be appended an 8-byte integrity check value which will be included in the encryption. Last, if AES in CBC mode is used then the CBC IV is calculated as the result of the the IV parameter included in the PKM-RSP message XORed with the concatenation of:

- the 48-bit MAC PDU header
- a 32-bit PHY Synchronization value of the MAP that a data transmission occurs
- the XOR value of the 48-bit MS's MAC address and the Zero Hit Counter

The complete sequence of messages exchanged during the PKMv2 protocol is illustrated in figure 2.9.

2.3 UMTS Architecture

The increasing demand for high quality multimedia services along with the need for modern pervasive applications has given birth to the UMTS [25]. UMTS is the product of a collaborative effort of international organizations, members of the 3rd Generation Partnership Project (3GPP) consortium. Today, 3rd Generation (3G) mobile networks based on the UMTS standard are deployed in Europe and USA (in lesser extend) with great success. Users of these networks benefit from the higher quality of voice and video calls, higher transfer rates, communication with the internet, and enjoy advance

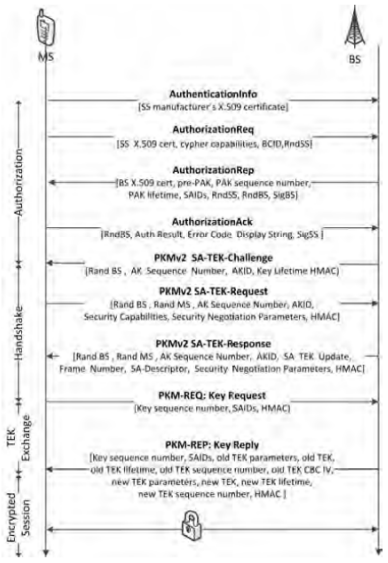


FIGURE 2.9: PKMv2 Phases and Messages

applications and value-added services and in some cases carry out wireless security-sensitive transactions like e-banking, stock trading, and e-shopping.

Unfortunately, the primary target of the designers of UMTS was to maintain maximum compatibility with the 2G systems. Additionally, its designers took into account the possible constraints in computational power mobile devices may have, and for that reason, they adopted relatively lightweight security techniques such as symmetric encryption [26]. Thus, even though UMTS is characterized by many major security enhancements comparing to its 2G predecessor the GSM it still presents architectural weaknesses.

2.3.1 Network Structure

In an abstract level the typical UMTS network architecture is comprised by three core components:

- Mobile Station (MS)
- UMTS Terrestrial Radio Access Network (UTRAN)
- Core Network (CN)

The MS, often refereed to as User Equipment (UE), is usually a mobile device (such as a smartphone) with radio access capabilities. Each device accessing a UMTS network, is equipped with a SIM card, which has imprinted several network specific constants such as the private cryptographic keys, the International Mobile Equipment Identity

(IMEI) which is a globally unique identification number, and the International Mobile Subscriber Identity (IMSI), which is an identification number for the user. To provide an increased degree of anonymity an expendable identification variable, namely the Temporary Mobile Subscriber Identity (TMSI) is produced and used instead of IMSI.

The UTRAN, is comprised by one or several Node B stations, which are tower formations equipped with antennas, each connected to a Radio Network Controller (RNC) through the lub interface. The RNC is a component that takes care of radio resource management, caries out some of the mobility management functions and is responsible for the encryption of traffic. The RNC also interacts with the Base Station Controller (BSC) component of the traditional GSM/EDGE Radio Access Network (GERAN), and enhances its capabilities.

The CN is comprised primarily by the Serving GPRS Support Node (SGSN) and the Mobile Switching Center (MSC) components which are the ones bridging it with the RNC components of the UTRAN. These are also responsible for forwarding packets and circuit switched information to and from UTRAN. The two components play a role in the authentication procedure and location management. For the latter procedure an additional component, namely the Visitor Location Register (VLR) is necessary. In essence, it is a database that contains the user identities associated to their location.

The CN may act as the Home Network for a subscriber if that user has been registered to it, or as Serving Network if a user registered to another network roams in it.

The HLR (also known as HSS), maintains statistics about its subscribers and generates billing and most importantly authentication information by interacting with the Authentication Center (AuC). It usually serves thousands of users, in most cases concurrently, for both both phone call and data/SMS services. Therefore, this is considered a point of increased interest for both the administrators as well as the malicious users.

Figure 2.10 displays the basic architecture of a UMTS network.

2.3.2 Security Mechanisms

The UMTS security architecture defines a set of procedures that the user's mobile equipment as well as the network should execute in order to achieve increased message confidentiality and integrity during their communication. In the heart of the UMTS security architecture lies the user authentication mechanism known as Authentication and Key Agreement (AKA) [25]. This mechanism is somewhat similar to the authentication in Global System for Mobile Communications (GSM). The idea to use public keys in the

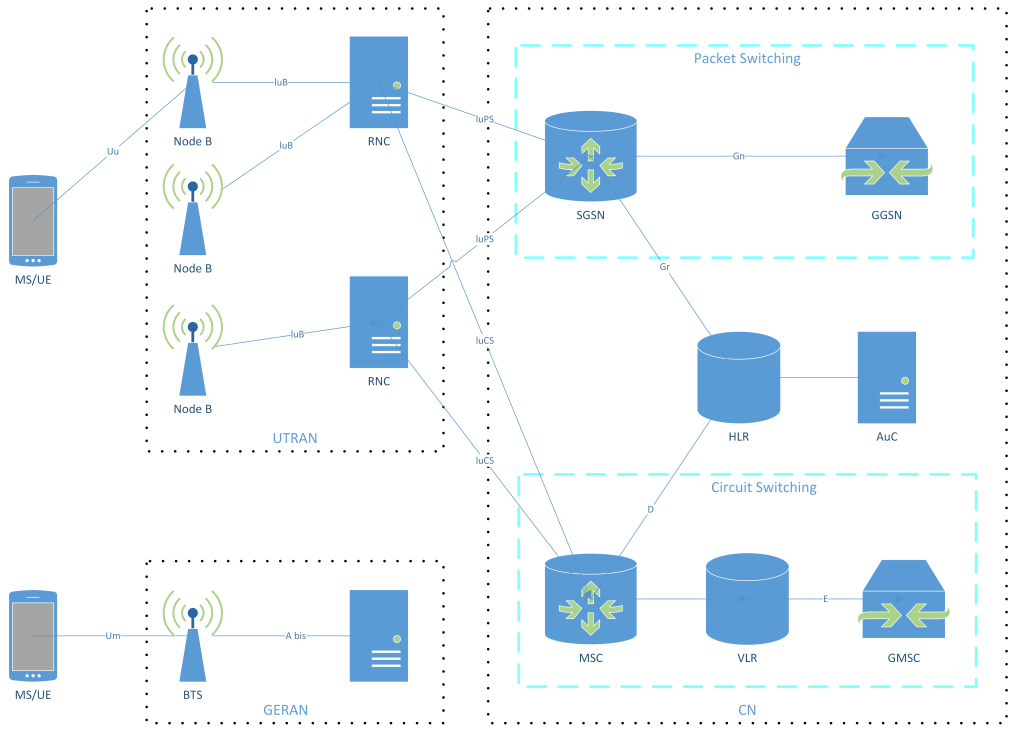


FIGURE 2.10: Basic Architecture of UMTS Network

process of authenticating the users, was abandoned, mainly due to backwards compatibility (with GSM) and for performance considerations. The authentication in UMTS is based on a 128-bit symmetric secret key, namely K_i, which is stored in the user's tamper-resistant Universal Integrated Circuit Card (UICC) and in the corresponding Home Location Register (HLR) of the user's Home Network (HN). The AKA scheme is a combination of the well known challenge response-protocol found in GSM and the authentication mechanism based on sequence number as defined by the ISO organization [27]. The network entities that take part in the user's authentication procedure are:

- The UE and more specifically the Universal Subscriber Identity Module (USIM) application stored in the UICC.
- The Serving GPRS Support Node (SGSN) of the HN or the Serving Network (SN).
- The HLR of the user's HN.

The authentication procedure in UMTS is mutual, which means that both the network is authenticated to the UE and the UE is authenticated to the network. After successful authentication the two ends agree on the use of two additional 128-bit symmetric keys. These keys are derived from the master key K_i and renewed every time the user is authenticated. The procedure typically initiates after the Mobile Station (MS) attaches to the network and sends its identity. Note, that the user can be identified either by

a permanent ID, i.e., the International Mobile Subscriber Identity (IMSI) or, usually, a temporary one known as Temporary Mobile Subscriber Identity (TMSI). During the process, the user's ID is forwarded from the Radio Access Network (RAN) sub-network to the core network, that is, the SGSN serving that particular area. In any case, the latter entity may send an Authentication Data Request message to the HLR of the user's HN in order to acquire Authentication Vectors (AV) required to authenticate the user. This happens only in cases that no AV for that particular user is available locally in the SGSN. For instance, the user attaches for the first time to this SGSN or the available in the SGSN AVs for that user have been already consumed. Since the HLR possesses the master key (K_i) for each user i is capable of creating the corresponding AVs. The vectors are sent back to the SGSN in charge by making use of a control message known as Authentication Data Response. Each vector can be used only once except the case the SGSN does not receive an answer from the MS.

After the SGSN in charge acquires some AVs (they are sent usually in batch), it sends an Authentication Request to the user. The request contains two parameters:

- A RAND which is a random number
- The AUTN, i.e., the authentication token. These parameters are transferred in the tamper resistant environment of the UICC/USIM and stored for further processing

The USIM is also aware of the K_i, and uses it along with the received parameters RAND and AUTN to perform a series of calculations similar to those that took place for the generation of the corresponding AV in the HN's HLR. The outcome of this procedure enables USIM to verify that the AUTN parameter was indeed created by the HLR of the HN and also that it is fresh (i.e., it is not a message replay). In case that the above verifications have a positive outcome the RES (result) parameter is calculated and sent back to the corresponding SGSN by utilizing a User Authentication Response message. Upon that, the SGSN compares the received RES with the XRES (Expected Response) which is contained in the initial AV. If the two values match then the user is granted access to the network.

Moreover, as already mentioned, two other keys that will be used for confidentiality and data integrity are calculated by the USIM. Using a security mode command the same keys, which are contained in the initial AV, are transmitted by the SGSN to the corresponding Radio Network Controller (RNC). These keys are known as Cipher Key (CK) and Integrity Key (IK). Note that while these keys are part of the corresponding AV and thus immediately available to the SGSN, the USIM has to calculate them by itself. An overview of the authentication sequence described above is depicted in Figure

2.11. It is to be noted that this section presents only the fundamental information on UMTS security architecture required for comprehending the concepts described in the next chapters. For an in depth analysis the reader should always consult [25].

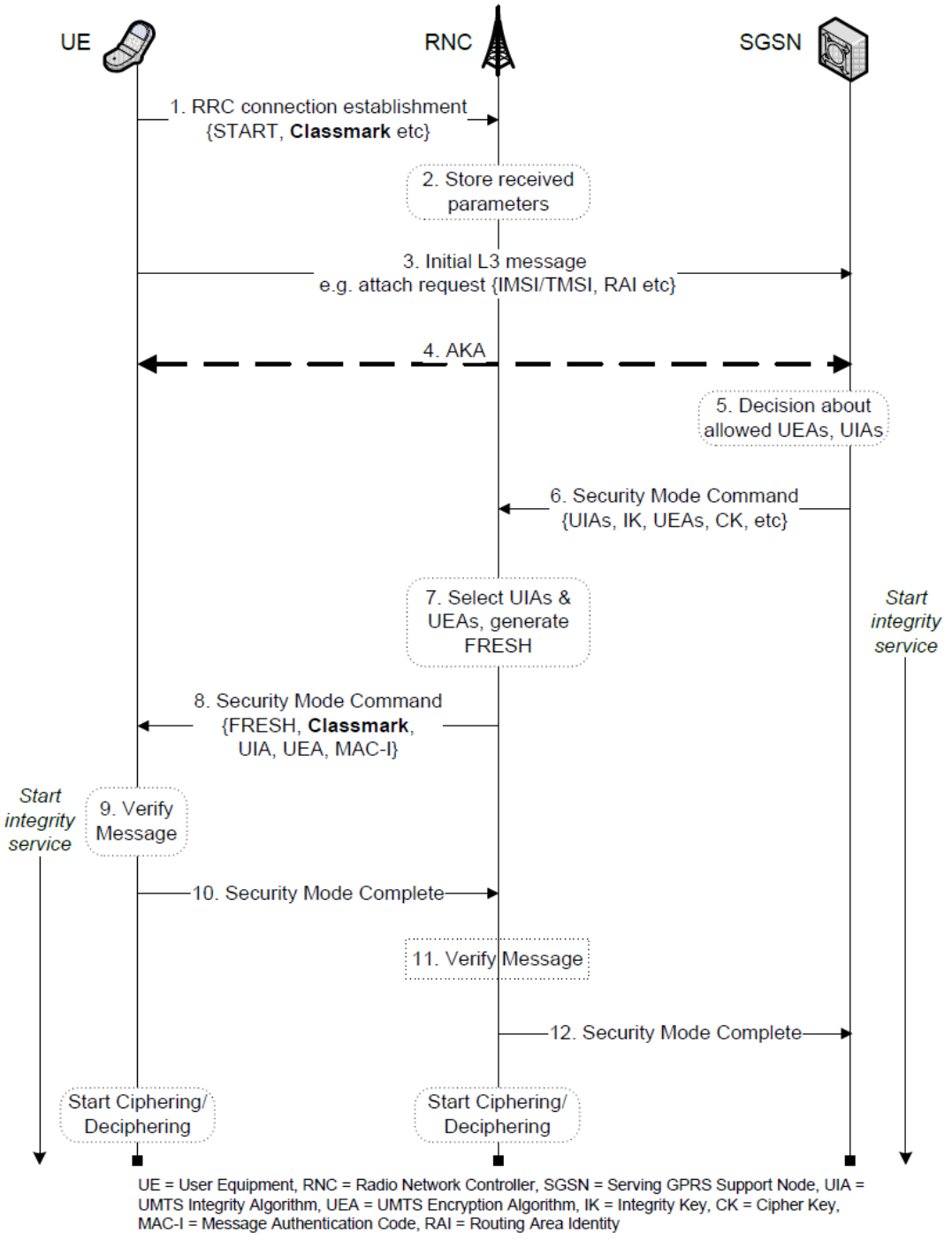


FIGURE 2.11: Initiation of Security Services in UMTS

Chapter 3

Mac Layer Attacks Against IEEE 802.11

Protection against wireless attacks is an active research topic but until now, most of the research is focused on Wireless networks such as WLANs, Wireless Sensor Networks (WSN) or 2G mobile networks such as GSM. Many different vulnerabilities have been discovered, and the corresponding attacks have been described, unfolding at different layers including physical, MAC, network or application ones. For instance, a jamming attack [28] is unleashed in the physical layer and can prove very hazardous since it cannot be addressed by adopting a more sophisticated network security architecture design or by cryptographic techniques. In [29] the authors explore the feasibility and effectiveness of jamming attacks in wireless networks and propose detection schemes. Other examples include: for the data link layer exhaustion attacks [30], and unfairness attacks [31], for the network layer black hole attacks [32], [33] and smurf attacks [34], and for the transport layer flooding attacks [35], and desynchronization attacks [34].

In this chapter a purpose-centric categorisation of attacks against 802.11 is adopted. As expected, the majority of attacks analysed, are against WEP protected networks as this protection mechanism has critical vulnerabilities. Therefore, most of the attacks described here, have nowadays part of publicly available penetration testing tools. However, to our knowledge, the attacks given in sections 3.0.5.5 and 3.0.5.10 are not known to be implemented by any commonly available tool and remained strictly theoretic.

3.0.3 Key Retrieving Attacks

The goal of key retrieving attacks is to reveal the Secret Key of the network. These attacks are considered critical not only because they reveal the most valuable asset of

the network (i.e., the secret key) to malicious entities but also because they may be conducted in a totally passive way. In such cases, the attacker simply needs to monitor the network for specific packets, usually the ones exchanged during an authentication session, and use them in the key cracking process which is executed offline. However, even thought the passive mode of such attacks renders them totally untraceable, the attackers often choose to actively take part in them, in order to speed up the process. This possibly decloaks the aggressor and opens a window of opportunity to detection methods.

3.0.3.1 FMS Attack

The Fluhrer, Mantin and Shamir (FMS) attack [36] is the first document WEP cracking method. This attack is based on the weakness of the RC4 algorithm. In FMS the attacker needs to monitor and store a single encrypted packet. Since, the first byte of the keystream is predictable, the attacker is enabled to make assumptions about a subsequent Key byte. By repeating the process, all the possible values of that byte will be revealed, but the actual will be the one revealed with higher frequency. From that point on the same cycle is repeated for the rest of the bytes of the key.

3.0.3.2 KoreK Family of Attacks

The cryptanalyst with the pseudonym KoreK published seventeen attacks for cracking the WEP key. The KoreK attacks are based on similar mathematical principles as the FMS, but use several different techniques in order to reduce the size of the key search space. The KoreK attacks follow similar execution methodology as the FMS as they also relay on statistical methods to vote for actual keys from a list of possible ones. In that way, they require a significant amount of IVs be captured before the cracking process is completed. Usually, a method of artificially increasing the amount of IVs transmitted (e.g., ARP injection) is employed so that the attack proves practical speed-wise. For a detailed analysis of this family of attacks the reader should consult [37].

3.0.3.3 PTW Attack

The Pyshkin, Tews, Weinmann (PTW) attack [38] was based on Klein's attack against the generic version of RC4 [39]. The PTW requires dramatically less IVs/data frames than the previously described methods, but is constrained to the ARP packets, thus making techniques such as ARP injection necessary.

3.0.3.4 ARP Injection

ARP injection is not actually an attack itself and if executed alone it cannot pose a threat to the privacy of the network key. However, it is solely used as the stepping stone for the Key cracking methods, that require many IVs (e.g., the FMS one). Its purpose is to manipulate the network in such a way so that new IVs are produced constantly and in large numbers even if no actual data is transferred in the network. The forcefully generated IVs will then be captured by the attacker and be fed to the respective Key cracking algorithms in a subsequent offline step.

In an ARP injection the aggressor constructs an ARP Request packet with broadcast IPs, encrypts it with a Pseudo-Random Generation Algorithm (PRGA) and finally transmit it. Upon reception, the AP will re-broadcast it to network, producing in that way a new IV for each request.

3.0.3.5 Dictionary Attack

The Dictionary Attack is, at the moment, the only reliable method for retrieving weak WPA/WPA2 keys. It has also been used for cracking WEP keys in less extend. Initially, the aggressor monitors the network hoping to catch a live handshake. Alternatively, she can force a handshake to take place immediately, by issuing deauthentication frames (usually a single or a very small number). Next, the attacker attempts to reproduce the third message of the captured 4-way-handshake based on keys contained in a large database, usually refereed to as dictionary. Essentially, this is a brute force practise. As such, it is considered effective only if the dictionary utilized contains the passphrase and its efficiency heavily depends on the computational power the attacker possesses.

3.0.4 Keystream Retrieving Attacks

In WEP protected networks it is possible for someone to benefit even from the knowledge of just the keystream. Knowledge of the keystream empowers the attacker to forge packets, encrypt them and inject them to the victim network. Moreover, as a bi-product of some keystream retrieving attacks the plaintext of some portions of packets is revealed, allowing an intruder to learn details about the topology of the network under attack.

3.0.4.1 ChopChop Attack

The ChopChop attack [40] enables an attacker to retrieve the m last bytes of both the keystream and the plaintext of a packet. This attack initially attempts to deduce the

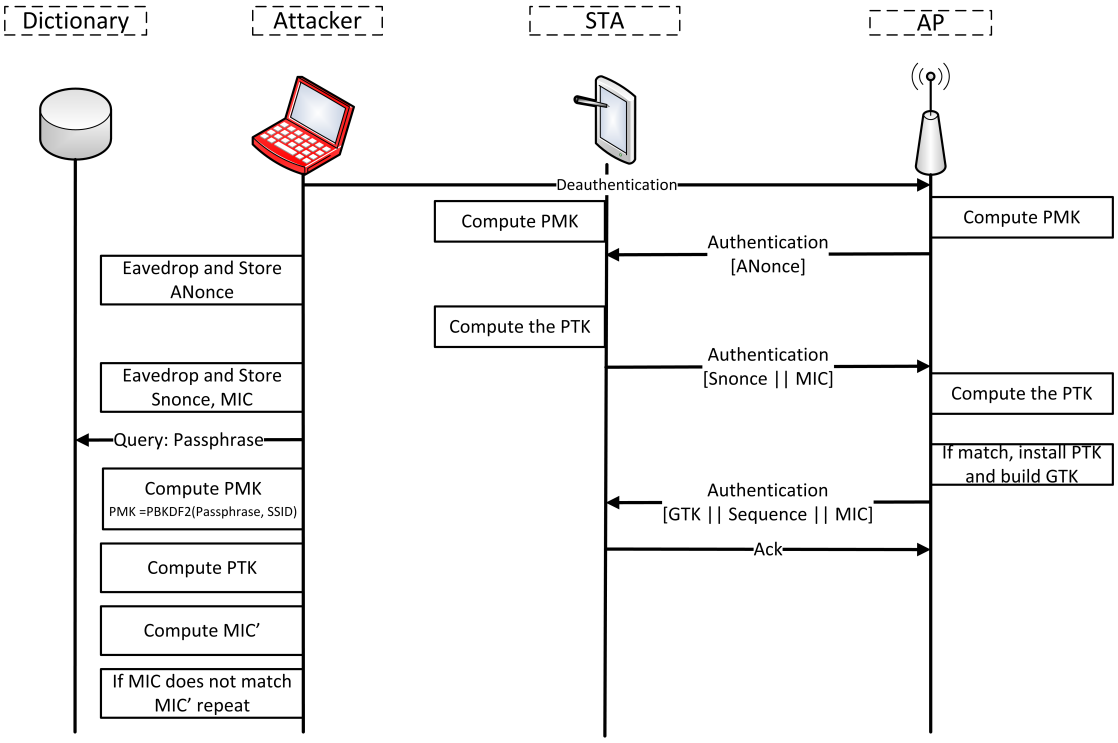


FIGURE 3.1: Dictionary Attack

plaintext of the last byte of a message, after 'chopping' the corresponding byte of the chipertext. In this truncated form, the frame will have invalid Integrity Check Value (ICV). At this point, the attacker initiates a cycle that involves guessing the plaintext value of the 'chopped' byte, XORing it and sending the modified message back to the network. Theoretically, the AP must reply with a failure message, indicating that the ICV is invalid every time the attackers guess is wrong. In that way the AP is abused as an oracle to inform about the validity of her guess. In the end, the attacker will know the plaintext of the truncated byte, and the keystream as well. Statistically, only $128m$ guesses are required on average ($256m$ guesses maximum) to retrieve the last m bytes of a packet.

3.0.4.2 Fragmentation Attack

The fragmentation attack can reveal a portion of the keystream by taking advantage of the fragmentation mechanism of 802.11. This mechanism allows large packets to be broken into several smaller ones that can be sent independently. The prerequisites are that the intruder must have first falsely authenticated herself to the network and captured at least one data packet from it. Since the first 8 bytes of plaintext are predictable, the attacker can deduce exactly 8 bytes of keystream with high probability. Unfortunately, 8 bytes of keystream leave room for only 4 bytes of data (since the ICV itself requires

another 4 bytes) which are insufficient for constructing anything meaningful. By employing the fragmentation mechanism, she constructs a number of 8 byte fragments with content of her choice. Finally, she sends these small sized packets through the AP to the broadcast address. Typically, the AP will reassemble the fragments and broadcast them as a single packet this time. Since the plaintext of this packet is known beforehand, the attacker will become able to retrieve keystream as large as that packet's length.

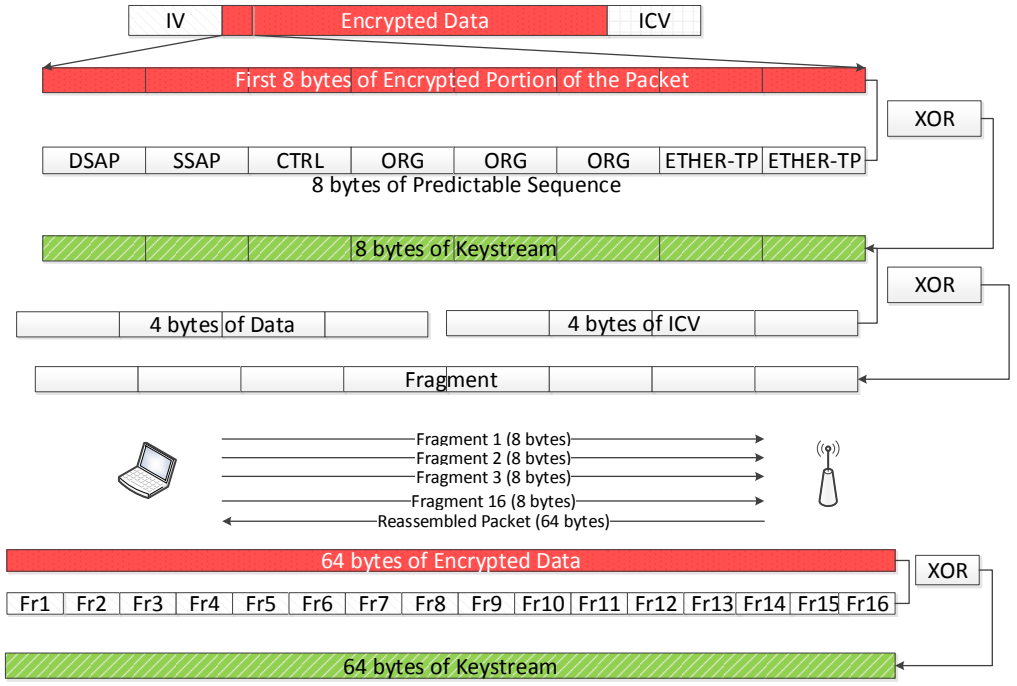


FIGURE 3.2: Fragmentation Attack

3.0.4.3 Cafe Latte Attack

The Cafe Latte attack was the first one used for retrieving the WEP key, without even requiring from the victim-client to be anywhere near the victim-network (e.g., he could be seating in a Cafe enjoying his latte). The authors in [41] proved that the only requirement is that the client was once authenticated to the victim-network and is currently probing for it. Initially, the attacker collects the probe messages from her victim and poses as one of the corresponding APs. The victim being lured by the identical Extended Service Set Identification (ESSID), authenticates and associates with the attacker's rogue AP. This is possible since WEP does not incorporate a mechanism for authenticating the AP. At this point, the victim will self assign a private IP and then will start sending encrypted gratuitous ARP packets. By manipulating that ARP packet she will try to deduce the

IP address of that client. After that she will be able to send a flood of encrypted ARP Requests, thus artificially increasing the IVs.

3.0.4.4 Hirte Attack

Hirte Attack is another "AP-less" method for retrieving the WEP. It can be perceived as a mix of the Cafe Latte and fragmentation attacks. Similarly to the Cafe Latte attack, the attacker must first acquire an encrypted packet (a gratuitous ARP packet or an IP packet) after setting up a Honeypot. Then, by breaking that packet into fragments and changing their order, she transforms it to an ARP Request one. From that point on a flooding of these messages can take place to harvest IVs for offline cracking attacks.

3.0.5 Availability Attacks

This category of attacks is usually deployed against specific clients or against the entire network (e.g. the AP) and leads to loss of service. Most of the attacks can be mounted by simply broadcasting a number of forged 802.11 management messages. This process is considered trivial, in versions of the standard up to 802.11n [42] as the management messages are transmitted unprotected. In all cases, the effects of these attacks apply as long as the corresponding attack takes place. A comprehensive survey of DoS attacks in 802.11 is included in [10].

3.0.5.1 Deauthentication Attack

The deauthentication attack is considered the most potent of the DoS ones due to its simplicity and efficiency. The Deauthentication frames are unprotected management frames and so, they can easily be spoofed by an ill-motivated entity. Surprisingly, the specification dictates that upon issuing of such frames, the client must be immediately exiled from the network without any further validating process. In real life situations, even if the client will loose connectivity with the network, typically he will attempt to re-authenticate immediately. The re-authentication cycle is usually very brief but of course, the attack may be mounted repeatedly depriving service for a longer period of time.

3.0.5.2 Disassociation Attack

The disassociation attack is similar to the Deauthentication one in matters of executional simplicity and effects. In this case, the attacker will issue a Disassociation message

instead. Theoretically, such attacks are less efficient because of the less procedures involved to return to the associated state. Thus, the duration of loss of service is expected to be briefer and more packets are required to be transmitted to maintain a steady loss of service.

3.0.5.3 Deauthentication Broadcast Attack

The Deauthentication Broadcast Attack works in the same way as the simple Deauthentication one, but instead of a specific client address the packets are forged with the the broadcast address. All clients in the network will receive this message and drop the connection immediately. Since all the clients of the network, will possibly attempt to authenticate at the same time, the system is also expect to be stressed mildly.

3.0.5.4 Disassociation Broadcast Attack

Similarly to the Deauthentication Broadcast Attack, this attack also causes loss of connectivity to all clients in the network, with the transmission of a Disassociation frame this time. The re-association process is briefer than the re-authentication one and less computationally intensive, thus the symptoms of this attack mode are expected to be less severe.

3.0.5.5 Block ACK flood

The Block ACK flooding attack is based on the Add Block Acknowledgement (ADDBA) mechanism introduced in the 802.11n amendment. This mechanism enables a client to transmit data as a single block instead of several smaller segments. An ADDBA message has to be send on behalf of the client to notify for its intention to initiate such a transaction. This message contains the corresponding frame sequence numbers of the respective segments and from that point on the AP will only accept frames that fall within that range. If the attacker forges a ADDBA frame with the client's MAC and randomly selected, but large sequence number, then all traffic originating from that client will be ignored until the sequence numbers indicated in the invalid ADDBA frame have been reached.

3.0.5.6 Authentication Request Flooding Attack

In this case the aggressor attempts to exhaust the resources an AP by causing an artificial excess to the maximum number of clients, the victim AP can support simultaneously.

More specifically, the client association tables of the AP, are updated with the receipt of every Authentication Request message, even if a client does not manage to complete the authentication process. In such attacks, the intruder will have to emulate a large number of non existing clients and issue a stream of authentication frames on behalf of each one of them. The client association table of the target AP will probably become corrupt, overflown with fake entries, which will quite possibly, lead the AP to become unable to associate new, legitimate STAs.

3.0.5.7 Fake Power Saving Attack

The fake power saving attack was originally described in [43]. It is based on the missuse of the Null Data frames and the abuse of the Power Management mechanism. This mechanism helps to reduce the power consumption of an STA by making it function in a energy constrained mode. The transition to doze mode is initiated with a transmission of a Null Data frame that has the Power Save bit set to 1, from that client. When in doze mode the STA is unable to receive or transmit any frames and the AP temporarily buffers all traffic destined to it. Since the Null Data frames do not carry any payload they are not protected and can easily be forged. By simply transmitting a tampered Null Data frame, the AP will considered the attacker's target STA asleep, thus stop transmitting packets to it. The fake power saving attack is the only availability attack that is not based on the transmission of management frames. Actually, the structure and role of the null data frames has been questioned in [44, 45].

3.0.5.8 CTS Flooding Attack

The Request to Send (RTS) - Clear to Send (CTS) is an optional mechanism for controlling the access to the RF medium. When enabled, the STA has to request access to the medium for a specified amount of time with an RTS frame. That privilege is granted with the issuing of a CTS frame by the AP. In a CTS Flooding attack a malicious outsider may constantly transmit CTS frames to itself or another STA of her choice, forcing, in that way, the rest of the clients in the network to suspend their own transmissions.

3.0.5.9 RTS Flooding Attack

An RTS Flooding Attack also abuses the RTS/CTS mechanism but works in reverse way to the CTS Flooding one. It transmits a series of spoofed RTS frames with a

large transmission duration window. The attacker is hoping to monopolise the wireless medium, by forcing the rest STAs to back-off from transmitting. [46] provides an empirical evaluation of the different flavours of CTS and RTS attacks.

3.0.5.10 Beacon Flooding Attack

The Beacon Flooding attack may be used in two different flavours to achieve either annoyance or denial of entry of new clients to the network. In the first case, the attacker will send a stream of fake Beacon messages with non existing ESSIDs, hoping to overflood the list of available networks, making it troublesome for the end-user to locate his preferred one manually. In the second case, the attacker will transmit a flood of spoofed beacon frames with a specific ESSID but with alternative (non existing) BSSIDs. Depending on the implementation, most probably the STA(s) will start a process of validating if each of the synonymous ESSIDs.

3.0.5.11 Probe Request Flooding Attack

The goal of a Probe Request Flooding Attack is to stress the resources of a victim AP. According to the 802.11 specification the network must resolve every Probe Request by issuing Probe Response. These messages contain details about the network and the capabilities of the AP. An attacker may send a constant stream of fake Probe Request packets towards the AP. If this is done in high volumes the AP will consume most of its resources into serving non existing clients, and eventually it will fail serving its legit ones.

3.0.5.12 Probe Response Flooding Attack

This attack also takes advantage of the Probe mechanism although it works in a reverse way to the Probe Request one. This time, the attacker targets specific clients rather than the AP and by acting like an AP, she broadcasts fake Probe Response messages with false, misleading details. In theory, this effectively prohibits these STAs from connecting to the AP.

3.0.6 Man-in-the-Middle Attacks

The goal of Man-in-the-Middle attacks is to provide an ill motivated entity with full access to the traffic of the clients of a network and enable her to stealthy modify the

data they send or receive. In such cases, all traffic sent and received passes through the attacker, who for the most time, plays the role of a forwarder of traffic.

3.0.6.1 Honeypot

In the context of wireless networks, Honeypots are wireless networks operated by malicious administrators with the aim of luring clients to connect to them and then unleash some type of attacks on them with ease. When the clients connect all traffic is visible to the attacker. More importantly, the attacker may use penetration testing to discover security holes to the connected clients and unleash higher level attacks against them.

3.0.6.2 Evil Twin

An Evil Twin is special case of Honeypot that advertise an existing ESSID lure users of the valid network into connecting to it. Evil Twin APs are possible due to the fact that (a) APs with the same ESSID may co-exist in the same area and (b) the client usually connects to the AP with the strongest signal disregarding the Basic Service Set Identification (BSSID). Initially, the attacker brings up a fake AP (usually a software one) that advertises the same ESSID with a valid one in the vicinity. Preferably, the impersonated networks must be open (e.g., networks of coffee shops, airports etc.) or at least their credentials should be easily acquired by the attacker (e.g., hotel wireless connection). Naturally, if the attacker's Network Interface Card (NIC) transmits with a stronger signal then the client will prefer to connect to that fake network. As in the case of a normal Honeypot from that point on the attacker can launch higher level attacks or simply monitor the traffic.

3.0.6.3 Rogue Access Point

Rogue APs are unauthorized access points (i.e., either hardware or software AP) within corporate, home or office premises usually set up by undisciplined insiders of that network. In most cases these users simply aim in bending the rules of a strict security policy. Another possibility however is that these devices operate under the supervision of traitors, with an ulterior purpose to leave a backdoor open for outsiders. Rogue APs are usually connected to the wired counterpart of the network.

Chapter 4

Mac Layer Attacks Against 802.16

In this chapter attacks against IEEE 802.16 are analysed. These are organised according to a taxonomy adopted from [47], which is in accordance to the exploited process of the 802.16 protocol stack. Many of the described attacks are, as of this writing, simply theoretical. One can notice that the majority of the documented attacks against 802.16 aim in causing DoS to the network or just commotion to the end-user.

4.1 Ranging Attacks

As already mentioned in section 2.2.2, one of the basic steps of initial network entry is the process known as ranging. This procedure aims in having the two peers (BS and MS) acquire the correct timing offset and making the correct power adjustments so that their transmissions are aligned for the chosen physical method.

When a ranging transmission opportunity occurs (these are discrete time instances indicated in the UL-MAP message) for the first time, the MS shall send an RNG-REQ message. Once the BS receives a RNG-REQ message, it shall assign Basic and Primary Management CIDs for the MS and commit bandwidth. At the same time, the BS shall calculate the Radio-Frequency (RF) power level, frequency offset, and make timing offset adjustments necessary for optimal communication. Finally, it will construct an RNG-RSP message with all this information and transmit it using the Initial Ranging CID. These two messages are responsible for the ranging procedure and have similar format. In more detail the RNG-RSP message defines:

- The Basic and Primary Management CIDs for this SS.

- Information about RF power level adjustment.
- Information about offset frequency adjustment.
- Information about timing offset corrections.

If the status of the RNG-RSP message is "success", the initial ranging procedure shall terminate. On the contrary if the Ranging Status field is "continue" the Basic CID shall be used and MS and BS shall continue exchanging RNG-REQ and RNG-RSP messages for fine-tuning the parameters mentioned above. Once the RNG-REQ is within the tolerance threshold of the BS, the MS shall join data traffic in the UL. If the Ranging Status is "abort" then the MS repeats the cycle of initial network entry by scanning for DL frequency.

Besides initial network entry, ranging also occurs at predefined time intervals. Periodic ranging allows the MS to adjust its transmission parameters so that optimal UL communication levels are maintained. Periodic ranging may be also initiated by the BS. For this reason the MS should always be able to accept RNG-RSP messages in an unsolicited manner. These messages are not encrypted or integrity protected and they are stateless, i.e., an MS will proceed to actions dictated in an RNG-RSP message if that message is addressed to it and appears to be well-formed. Whatever the case, an attacker may manipulate the ranging messages in many ways to affect single users or the entire network. The relevant attacks found in literature are analysed below.

4.1.1 RNG-RSP DoS Attack

This attack is possible due to the fact that RNG-RSP messages can be transmitted in an unsolicited manner. The RNG-RSP attack may be addressed to a single target MS or multiple ones. In the first case, the attacker must also know the CID used by the victim MS. This information can easily be sniffed from any (unencrypted) management message exchanged between that specific MS and the BS. The she needs to forge and transmit to her victim an RNG-RSP message with the "Ranging Status" field set to "abort" [48], [49]. This will force the victim to disconnect from the network immediately. In the case where the attacker needs to create a wider impact, she will cycle through all 65,536 possible CIDs in a brute force manner, and forged RNG-RSP messages for each CID. After that the victim device will attempt to reconnect to the network by executing Initial Network entry. It is possible, by repeating this procedure against a significant number of users, to achieve DoS for an even larger number of users than the ones that immediately targeted. This is because each round forces the network to a series of heavy signalling procedures, including the Initial Network entry and several cryptographic procedures.

4.1.2 RNG-RSP Annoyance Attack

An RNG-RSP message can be altered in a number of different ways by an intruder aiming to disrupt the normal MS communication. For example, the attacker may alter the RNG-RSP "Frequency" field in order to force the victim MS to shift to another channel. Theoretically, the MS will have to rescan many frequencies (wasting 5 ms in each one) until it locates again the proper channel. As a consequence, this will cause annoyance to the users of network. Similar results can be achieved by shifting only the UL or the DL channel, or by altering other fields of the same message such as the Timing Adjust and Power Level Adjust [50]. In matters of execution and implementation cost the RNG-RSP Attack is similar to the RNG-RSP DoS one. Let us underline that with very little effort this mild attack may be transformed to a stepping stone for far more dangerous methodologies. For example, in a potentially more dangerous scenario, the attacker would have to shift the victim MS to a frequency where a rogue BS set by the attacker exists.

4.1.3 RNG-REQ Downgrading Attack

One of the purposes of the RNG-REQ message is to inform the BS about the preferred DL burst profile. By replacing the optimal burst profile with a less effective one, the attacker will achieve downgrading the service [50], [49]. The effectiveness of the attack depends on the selected choice of the burst profile. Generally, this kind of information cannot be deduced on the fly, for any given MS.

4.1.4 RNG-RSP Water Torture Attack

This is a modification to RNG-RSP Downgrading Attack but with totally different possible effects. An attacker might forge and send an RNG-RSP message with the "Power Level Adjust" field set to the maximum value. This will force the MS to operate in higher energy requirement state, thus causing a quicker drain of its battery resources [50]. The effects, which are higher battery depletion rate persist for a considerable time (i.e., until the next ranging/periodic ranging). On the other hand, the drain of energy is not expected to be low, and all the attacker can hope for, is a simple annoyance.

4.1.5 RNG-REQ DDoS Attack

In this case a group of collaborating attackers may produce a large number of fake RNG-REQ messages (with different values each time) and simultaneously transmit them to

the target-BS in order to have it function under heavy burden [51]. The construction and transmission of multiple RNG-REQ messages with random fields and fake MAC ID, in contention mode, is not a resource intensive process for an attacker. On the contrary, the response part in the BS side is a multi-step process which involves the allocation of Basic and Primary management CIDs, deciding whether the signal is good enough or any adjustments are necessary, constructing an RNG-RSP etc. A collaborative attack of this kind is expected to cause considerable burden in the BS which will possibly result in lower quality of service or even Distributed DoS (DDoS) for all legitimate MSs connected to the specific target-BS. Actually, when an attacker attempts such an attack she affects the system in many different levels. For example she a) artificially increases the number of collisions in the network, b) imposes burden on the BS by forcing it to conduct the ranging process for a large number of virtually non-existing MSs, and c) tricks the BS into ranging and then committing bandwidth and CIDs to fake MSs. The only real requirement in matters of implementation methodology is the attacker to have control over a small number of programmable MSs and synchronize their actions. The cost of this orchestrated attacks depends on its scale. Since at this phase addressing information has not been assigned to the MS (and MAC address field contained in the RNG-REQ can easily be spoofed) the BS has not any means of recognizing the attackers. Additionally, the small volume of traffic involved, makes it extremely difficult for external tools such as Intrusion Detection Tools to classify such behaviour as intrusive.

4.1.6 MOB ASC-REP DoS Attack

When association level 2 is used, during the ranging procedure, an MS may receive an Association Result Report (MOB ASC-REP) message instead of several RNG-RSP ones. In this case, the RNG-RSP information that is sent by each target BS is gathered to the serving BS over the backbone network. The BS then aggregates all the data from the RNG-RSP messages to a single MOB ASC-REP message, transmits it over the Primary Management CID. The MOB ASC-REP report messages are unprotected making it possible for an adversary to forge them stating that no services are available from all the target BSs [50], [52], [53]. A MOB ASC-REP DoS Attack will prevent the victim MS from associating with the optimal BS which translates to lower QoS for the target MS.

4.2 Power Saving Attacks

4.2.1 MOB TRF-IND Water Torture Attack

This attack takes advantage of the unprotected nature of the MOB TRF-IND messages. These messages are normally sent from a BS to a sleeping MS, when there is traffic pending for that MS. To systematically skip the sleep mode is expected to have a negative impact to the lifetime of the battery of the MS. If the attacker is able to forge valid MOB TRF-IND, and repeatedly transmit it to a sleeping MS in the vicinity she would be able to drain the energy resources of the victim on a higher pace. This attack was first described in [48] and is also mentioned in [53], [54].

4.2.2 BR and UL sleep control header Annoyance Attack

It is possible for an MS to request activation of sleep mode by issuing a BR and UL sleep control header instead of relying to the traditional mechanism of MOB SLP-REQ. The authors in [55] claim that it is feasible for an attacker to forge a BR and UL control header with the victim's identity (MAC Address) and send it to an MS to have it fall into sleep mode. As a result, the BS will stop transmitting messages to that MS and DoS will take place. In our opinion, this is actually an manufacturer-dependent issue. The specification indeed leaves room for such an invalid request for sleep to happen even though in careful implementations of the standard the BS is expected to reject or postpone any requests for Sleep mode if the BS has currently queued traffic pending for that MS. In our opinion, the attacker can hope for disturbance for the user that will include a brief lack of service for the MS to fall into sleep and the wake up after the first availability window. Moreover,the attacker must know whether the victim's equipment does support sleep functionality.

4.2.3 Secure LU DDoS Attack

Location Update (LU) is a process through which a BS stays informed about the current location of a given MS. This process may be initiated by the MS at will or when one of the following conditions apply:

- the MS detects a change in paging group,
- on a standard basis, e.g., prior to the expiration of the idle mode timer
- as part of its power down procedure

- when the MS MAC hash skip counter exceeds a threshold

There are two modes supported: secure LU or unsecure LU. In the first one, the MS is required to send an RNG-REQ message to the BS including an HMAC/CMAC tuple. Naturally, the BS will have to verify the HMAC/CMAC value. If the current BS does not share security context with the MS then it will have to acquire it from the backbone network via the LU Request message. The backbone network will generate and provide the keying material via an LU Reply message. The authors in [56] claim that this process may pose a considerable burden to the network when it is performed simultaneously by a large number of devices. Since any MS can request bandwidth for LU, the attacker will simply have to construct a valid RNG-REQ message but with wrong HMAC/CMAC. In principle, this attack is very similar to the RNG-REQ DDoS one but it involves some additional procedures (named above) by both the BS and the backbone network that may magnify the result and cause additional damage.

A misbehaving MS can generate a large number of such requests easily and without running the risk of getting discovered.

4.3 Handover Attacks

4.3.1 MOB NBR-ADV Downgrading Attack

MOB NBR-ADV messages are not integrity protected, giving a malicious user the capability of altering them at will. More specifically, by removing information about neighbour BS in the corresponding message fields, will prevent a valid handover procedure to take place as the victim MS will think it is isolated. While moving away from the serving BS the MS will have no other choice than to remain attached to it and the QoS will be reduced gradually until it will be out of service [48], [52], [50], [53]. In such attempts, the intruder must have already pre-established a tunnel between the MS and the BS, constantly eavesdropping for any MOB NBR-ADV messages and then transmitting a flood of spoofed messages almost simultaneously, but with stronger signal. Since this message is transmitted periodically, the attacker must follow its movement and always alter these messages, upon every broadcast. It is easy to realise that the attack focuses mostly on single target MSs.

4.3.2 MOB NBR-ADV DoS Attack

MOB NBR-ADV message can be manipulated in an alternative way that will allow an attacker to indicate the presence of a non-existing BS with better characteristics than the serving one [57], [50]. By transmitting messages crafted this way, a victim MS will disconnect from its currently serving BS and attempt to connect to a new one that does not actually exist.

4.4 Miscellaneous Control Message Attacks

4.4.1 SBC-REQ Security Downgrade Attack

The critical process of basic capability negotiation takes place during the initial network entry. During this step the MS informs the BS about the supported security capabilities of the device. This is carried out via a negotiation process that involves the exchanging of SBC-REQ message from the MS to BS and the SBC-RSP message from the BS to the MS. These messages are exchanged before the BS and MS start an encrypted session, so naturally no actions for securing the contents of SBC-REQ can be done. This vulnerability was first mentioned by [48] and later on in [49]. Also, the authors in [57] described a potential attack by exploiting this vulnerability. An attacker may attempt to transmit an altered an SBC-REQ simultaneously with the valid message sent from a legitimate MS to the serving BS during the network entry process, hoping that her fake but higher-power message will overshadow the valid one. The forged message should contain false information about the security capabilities of the legitimate SS, typically lower or no security capabilities. The authors claim that in the second case, the communication between the two parties will be conducted in a non-encrypted way, allowing any malicious entity to easily eavesdrop the communication.

4.4.2 FPC Downgrade Attack

The Fast Power Control (FPC) is an optional mechanism used for adjusting the power levels of multiple MS to an optimal level, simultaneously. It is much more efficient than the standard mechanism, namely the periodic ranging. FPC messages are always sent on the Broadcast CID and their format is rather simple: It contains the number of MSs to be affected, and for each MS its Basic CID as well as the necessary power correction. Once again, this management message is not integrity protected thus it may be altered to set the "Transmission Power" field of victim MSs to non optimal levels, in this case too low. In the first case, the aggressor by simply broadcasting a single

message after specific time intervals, will force the MS to go through the procedure of adjusting its power levels until the signal is strong enough. As collateral damage the accumulated power adjustment messages will possibly result in many uplink bandwidth requests. This generally causes collisions in uplink of the MS and stalls the procedure of acquiring correct transmission power. The second case, causes faster drain of battery of the victim MSs.

4.4.3 FPC Water Torture Attack

This is a slightly modified version of the FPC Downgrade Attack where the attacker sets the "Transmission Power" too high. This may lead to drain the batteries of MSs [50], [58], [52], [53] using a methodology and characteristics much similar to the ones of RNG-RSP Water Torture Attack.

4.4.4 RES-CMD DoS Attack

Reset Command (RES-CMD) is a message used to reset an MS that appears to be unresponsive to its serving BS, or in situation where there are persistent anomalies in the UL transmission. When this message is received by an MS then it shall reset itself, and repeat initial network entry procedure. Fortunately, this message is protected by HMAC/CMAC therefore it cannot be spoofed by an attacker. Nevertheless, it is possible to force the BS to transmit this message by itself. The trick here is to have the BS think, that the victim MS is malfunctioning. This can be achieve for example by systematically choosing to transmit at the exact same times as the victim SS. Provided that the two signals will arrive at the BS with similar power strength the final message the BS receives will appear as a single unintelligible message [52].

4.4.5 DBPC-REQ DoS Attack

Downlink Burst Profile Change Request (DBPC-REQ) is a message transmitted from the MS to the BS on the Basic CID to request a change of the least effective DL burst profile. Usually, this happens when channel conditions change with the DBPC-REQ message being utilized in such situation as a quicker alternative to the RNG-REQ message [53].

As expected, the DBPC-REQ message is also unauthenticated, an attacker can change the Burst profile (modulation, encoding etc.) with the intention of disrupting communication between the BS and MS by the misuse of the DBPC-REQ message.

4.5 Attacks Against WiMAX Security Mechanisms

4.5.1 Interleaving

This theoretic attack was originally mentioned in [59], [60]. It consists of two rounds: In the first round the attacker impersonates a valid MS and sends an Authentication Information message followed by an Authorization Request message (these messages were intercepted and stored during previous valid sessions). After receiving the Authorization Reply message the attacker must complete the protocol by providing a valid Authorization Acknowledgement response. The attacker is not in position to construct this message because she does not have knowledge of the valid MS's private key and cannot decrypt the Authorization Reply message. However, the attacker can start the second round (in parallel) aiming at using the valid MS as an oracle to construct an Authorization Acknowledgement message on her behalf. In this round the attacker will assume the role of a BS. By forcing the MS to start another protocol instance, it will use the Authorization Reply produced in the first round (it was received by the valid BS). The valid MS will provide the correct Authorization Acknowledgement message which the attacker will forward to the valid BS and finish the first round. In this way the attacker having acted as a Man-in-the-Middle will authenticate herself rather than the valid SS and trick the system into registering the wrong user.

4.5.2 AUTH-REQ Replay Theft of Service Attack

The authors in [50] noticed that the random number field contained in message Auth-Req, fails to protect against replay attacks. They implied that message can still be retransmitted by an attacker and the BS will have no means of knowing about its freshness, leaving room for a theft of service attack to occur. Actually, the random number field in the Auth-Req message is a mechanism introduced to associate each Auth-Rep message with one Auth-Req and not to protect Auth-Req from replay attacks. The MS will know for sure that the Auth-Rep is fresh, if the MS random number field matches the one originally sent in the Auth-Req message.

4.5.3 AUTH-REQ Replay DoS Attack

This attack is valid against the first version of the PKM protocol. Xu and Huang [61] described its methodology, stating that an ill motivated user is possible to store and replay an instance of a legit Auth-Req message sent by the SS in the past. If the BS has set a timer for rejecting duplicate Auth-REQs originating from the same

SS within a specific period, the attacker might be able to trick the BS into dropping even valid requests coming by the victim SS. Depending on the vendor it is possible for this attack to be feasible in the PKMv2 of the protocol. In this case there are two possibilities: (a) either the attack will take the course the authors described leading to a DoS against a small/moderate number of users, or (b) the BS will proceed normally with the authorization process giving room for a DDoS attack to grow. This possibility was also recognized in [62]. Considering the second case, for each Auth-Req message the BS will have to verify each of the messages signature, generate keying context, construct the Auth-Reply message and finally transmit it to the MS. It is obvious that this sequence of actions may be a burden to the BS especially if it is repeated many times and for a large number of simultaneous requests. The problem with this attack which differentiates it from other DDoS attacks, is that it has an upper bound. That is the collaborating attackers may reach a limit of simultaneous requests. This is due to the structure of the Auth-Req message which contains the SAID field. This value is validated and used for the construction of the Auth-Rsp. This practically limits the attacker to replay Auth-Req messages for only the MSs whose basic CID is active.

4.5.4 PKM-RSP Auth-Invalid DoS Attack

PKM-RSP are messages issued by the BS and sent to the SS. Generally, messages of this kind are comprised by the following fields:

- Management Message Type - for PKM-RSP messages the value of this field is 10.
- Code - this field identifies the type of PKM packet.
- PKM Identifier
- TLV Encoded Attributes - PKM attributes carry the specific authentication, authorization, and key management data exchanged between the MS and BS.

The Auth-invalid message is sent by BS to MS when (a) the AK shared between BS and MS expires or (b) the BS is unable to verify the HMAC/CMAC properly. This message has a great chance to be used as a DoS tool for shutting down legitimate users. First of all, the Auth-invalid message itself is not HMAC/CMAC protected. Furthermore, it can be sent in unsolicited manner from the BS (when one of the aforementioned conditions occur). The PKM identifier mechanism is not used in this case. Thus, the attacker can easily inject fake messages of this kind and have the MS pass to Reauth Wait state waiting further instructions from the BS. In the most likely event, that the MS's Reauth Wait timer expires without receiving any message, the MS will send a Reauth Request.

In the Reauth Wait state the device may accept messages such as an Auth Reject which will cause immediate break of all subscriber traffic[50], [52], [63].

4.5.5 DES CBC IV Attack

Cipher-Clock Chaining (CBC) [64] is a cipher mode in which the plaintext is broken into fixed size blocks and each one is XORed with the previous block before it is encrypted. In this way, each message is unique and each block is dependent on all preceding plaintext ones. The first block is a special case since no previous one exists, hence a random number, namely the IV, is be used instead. More specifically given a symmetric key K and a plaintext P:

$$C_i = E_k(P_i \oplus C_{i-1}), C_0 = IV$$

$$P_i = D_k(C_i \oplus C_{i-1}), C_0 = IV$$

Generally, it is important for the IV to be unique and unpredictable. If not unique, then the CBC mode is degraded to a simple Electronic Codebook (ECB) mode where the distribution of the sequences of characters is maintained thus allowing traditional cryptanalysis methods (such as statistical analysis) to succeed. If on the other hand, is not unpredictable, then it gives room to a chosen plaintext attack to succeed. It is true that while the IV for AES in CBC is produced in a secure way the same is not true for the IV that is used for DES in CBC mode. Actually, the specification states that the IV field in the keying information should be generated in a random way, and then be XORed with the frame number or the UL-MAP for the DL and UL respectively. The IV field is static for the entire TEK lifetime and is transmitted as an unencrypted field of the RSP: Key Reply message. On the other hand, the frame number is a simple counter which is exactly what makes the final IV material predictable. This vulnerability was first mentioned in [65]. Although an attack methodology was never given we assume that the author implies a known plaintext attack. In attacks of this type the aggressor typically, as a first step, captures a cipher block c_v. As a second step, she generates a plaintext block of information as:

$$p = IV_i \oplus IV_{i+1} \oplus P_{guess}$$

where IV_i is the IV used to construct the c_v, IV_{i+1} is an estimation of what the next IV is going to be and P_{guess} is a guess of the plaintext encrypted to produce cv. Next, the attacker sends and forces the victim to encrypt p as follows:

$$c_a = E_K(IV_i \oplus IV_{i+1} \oplus P_{guess} \oplus IV_{i+1})$$

and as a final step she compares the two ciphertexts. If $c_v = c_a$ then her original assumption about the plaintext block must be true. It is to be noted that, similar attack methodologies have been investigated for the IPsec realm in the past [66], [67].

4.5.6 DES CBC Insecurity Attack

This vulnerability was first revealed in [65]. According to the authors DES [68] in CBC mode looses its security after $2^{n/2}$ blocks encrypted with the same key, where n is the size of the blocks used by the cipher. Since DES utilizes 64-bit blocks, it is expected that after 2^{32} blocks of the respective size, the security of the system will be diminished. This is realistic as under real-life conditions WiMAX networks have data rates that exceed this security threshold before the end of the TEK's lifetime. Although an analytical methodology for this attack is never provided in the literature, it is certain that the first step on behalf of the attacker is to force the system to switch to PKMv1 and then instruct it to choose DES in CBC mode. This is necessary since in PKMv2 the Authorization Request (which is the message that informs the BS for the supported cipher suites) is protected by the signature of the MS. The attacker must first send a bogus SBC-REQ message and then transmit a fake Authorization Request message with the data encryption algorithm identifier field set to 0x01.

4.6 Multicast/Broadcast Attacks

4.6.1 GTEK Update Mode DoS Attack

GTEK is shared among all members of a multicast/broadcast group so that each member is able to decrypt the traffic it receives from the BS. GTEK is a symmetric key. This means that an MS cannot only decrypt data but also encrypt them using the same GTEK key. The members of the same group will be able to decrypt such messages but will not be able to distinguish if the message originates from the BS or an ill-motivated member of the group. As long as this message has a valid encryption and HMAC/CMAC the other MS will take for granted that the traffic is originated from the legitimate BS. An adversary MS, member of the group, can use this opportunity to send malicious traffic pretending to be the BS.

A possibly more harmful situation appears when the same scenario happens with the GTEKs. The GTEK is encrypted and transmitted to all group members using GKEK, which is also known to all group members. An adversary that is already member of the group can manipulate MBRA to distribute its own fake GTEKs using the GKEK she

rightfully owns. The messages will again be valid, and all the members will eventually replace their current keys with the fake one. After that, all the group members except the attacker will no longer be able to decrypt incoming traffic from the original BS [69]. This attack is straightforward in implementation and can affect all MSs within the same MBS group with a single alter/broadcast of a message (which is typically a moderate number of MSs). Moreover, the effects persist for as long as the current GTEK remains active. For prolonged period the attacker must actively and continuously alter/forge the Group Key Update Command with a fake key. The BS has no means of knowing that the MS of a given group have another (wrong) key.

4.6.2 GTEK Theft of Service Attack

Members joining a multicast/broadcast group are provided the active GTEK. With this key the members of the group are able to decrypt subsequent traffic, but also all traffic sent since the specific GTEK became active, even if the members hadn't join the group yet. Therefore, an attacker can passively store traffic and near the end of the GTEK lifetime join the network as a valid user [69], [61], [70]. The methodology of this attack is extremely simple and does not require any costly equipment. The actual service duration that the attacker will be able to intercept traffic is provider specific as the GTEK lifetime is not specified by the standard. Typical implementations set this counter anywhere from 30 minutes to 7 days which is adequate considering that MBS deals mostly with multimedia services. This attack highlights the issue of lack of backward secrecy of the MBRA.

4.6.3 MCA-REQ DoS Attack

The Multicast Assignment Request (MCA-REQ) message is sent from the BS to an MS that is requesting to join or leave a multicast polling group. Upon receiving this message the MS shall add the multicast CID to its transmission opportunities or remove it according to the Join/Remove command of the corresponding field. Subsequently, the MS will respond by sending an MCA-RSP message back to the BS. These messages are transmitted over the primary management connection. Also, since the MCA-REQ message is sent unprotected an attacker may remove an MS from a polling group at will. This attack can disturbance against a single user, but it is possible to be extended in a larger scale, causing overloading of the UL resulting in greater uplink delay [53].

Chapter 5

Mac Layer Attacks Against UMTS

In this chapter the major attacks against UMTS networks as documented in the literature so far are described. Note that the organisation of the attacks described in this chapter, follows a offended-subsystem approach. The majority of the attacks described here inflict DoS to the network.

5.1 Attacks Against Core UMTS

5.1.1 RRC connection Request Message

In Khan et al., [71] among other types of attacks, investigate the feasibility of a DoS attack by taking advantage of a particular flaw spotted in the UMTS security architecture. Their proposed attack involves the modification of the RRC connection Request Message and more specifically the field which defines the UE security capabilities. This message is not integrity protected since the AKA procedure takes place at a later stage and the MS and SGSN do not share a common IK yet. Any modification of this message will go unnoticed until eventually the AKA procedure completes and the Security Mode Command message is sent to the MS. This message includes the user's equipment security capabilities as received from the RCC Connection Request message in order to be verified by the UE. In case of mismatch the connection will terminate, but during the process sufficient resources will have been already consumed at both sides.

5.1.2 Signalling Attack

Lee et al. [72] introduce a novel DoS attack specific for the 3G wireless networks which they identify with the term "signaling attack". Unlike traditional DoS attacks that unfold in the data plane this one targets and attempt to overload the signaling plane. The signaling attack is implemented by sending low volume (for instance 40 byte packets) bursts at a specific time interval such that as soon as Radio Access Bearer (RAB) is torn down due to a period of inactivity a new packet burst that originates from the attacker forces for a new RAB establishment. This triggering of radio channel allocations/revocations is associated with a large number of signaling operations; more specifically 15 signaling messages are being processed by the RNC for the establishment of a synchronized RAB and 12 messages for its release. The results of this attack are:

- congestion of the RNC-BS with setup/release messages
- consumption of resources of the RNC processor
- potentially consumption of the battery of the MS

The attack can prove to be very dangerous since it does not require many resources from the attacker point of view (by using a cable modem with 1Mbps uplink bandwidth the attacker can simultaneously attack 160K MSs) and it can evade detection by traditional IDSs. In the same work the authors propose a technique for detecting and repelling this attack.

5.1.3 Dropping ACK Signal

The protection of IMSI is considered of paramount importance in UMTS. Therefore, it is transmitted and used as seldom as possible. To this end, temporary identities known as TMSIs are distributed to the users and used for all signalling communication. TMSIs are assigned to users, right after the initiation of ciphering, or every time a user roams to an area monitored by a different SGSN. Although, a TMSI is transmitted encrypted to the UE the SGSN does not associate the IMSI with the corresponding TMSI unless it receives a TMSI Allocation Complete message from the MS. If this message never reaches the intended SGSN then both the associations IMSI, TMSIold and IMSI, TMSInew are considered valid, by the SGSN in charge for uplink communication and the UE is free to use any of them. Contrariwise, for the downlink, the IMSI must be used because the network has no means to know which one of TMSInew or TMSIold is valid at the UE side at this particular moment. Upon such an event, the SGSN will instantly instruct

the MS to delete every available TMSI. The network may initiate the normal TMSI allocation procedure. Capitalizing on the aforementioned situation the aggressor might wish to position his equipment to a strategic location, for instance circumferential to a given network cell (where typically new TMSIs are assigned to subscribers entering the cell after a handoff). Then, he would monitor for TMSI Allocation Command messages and immediately drop any TMSI Allocation Complete message. This would cause new TMSIs to be created repeatedly, which would be expressed as DoS to all the users entering the particular routing area. Alternatively, this attack could be used to expose and create a database of IMSIs, which will be used for more persistent attacks.

5.1.4 Modification of Unprotected RRC Messages

The Radio Resource Control (RRC) messages are considered vital for the smooth operation of the UMTS network. It is without surprise that these signalling information messages are protected by integrity mechanisms. Unfortunately, either because these messages are exchanged during the early stages of a connection or for reasons of efficiency, many messages exist that are not integrity protected and therefore are vulnerable to manipulation. Some unprotected RRC messages might be:

- Handover to UTRAN Complete
- Paging Type 1
- Push Capacity Request
- Physical Shared Channel Allocation
- RRC Connection Request
- RRC Connection Setup
- RRC Connection Setup Complete
- RRC Connection Reject
- RRC Connection Release
- System Information (Broadcast Information)
- System Information Change Indication
- Transport Format Combination Control (TM DCCH only)

Modifying and issuing unprotected RRC messages is expected to cause general system instability, or at least commotion, which may lead DoS for the end-user. Let us consider the following example: an attacker would transmit an RRC Connection Release message during a valid ongoing session. Similarly, an attacker could transmit a forged RRC Connection Reject message, before a valid RRC Connection Setup Complete is transmitted.

5.1.5 Modification of the Initial Security Capabilities of MS

During this mode of assault, the attacker issues a forged a RRC Connection Request message with invalid classmark value for the classmark field, in order to cause the termination of the connection of a single user. This prospect was first proposed in [71]. However, if we take a closer look we may realize that a more interesting potential emerges, since the system under attack is forced to go through a sequence of heavyweight operations. More specifically, if the attacker has assembled a large database of stolen IMSIs she would be able to cause much more extensive damage. She would have to issue a very large number of simultaneous connection requests with bogus classmarks, triggering many simultaneous heavy operations (both bandwidth and computationally wise) to take place.

5.1.6 Modified Periodic Authentication Messages

Periodic local authentication in UMTS is a procedure for providing an additional security level to the network [25]. Potentially, it offers some sort of integrity protection in the U-plane.

In this procedure the volume of data transmitted during the RRC connection (i.e., the COUNT-C value [25]) is periodically checked by both the RNC and the UE. The system makes use of two variables to keep track of the user data transmitted from the MS towards the network. The first one, Count-CUE, tracks the volume of user data transmitted by the user equipment, while the other, known as Count-CRNC, stores the volume of user data actually received by the corresponding RNC. The value of these variables is cross-verified at regular intervals upon initiation by the RNC in charge. If a significant inconsistency is found then the RNC may decide to abruptly release the connection assuming that someone is injecting or dropping messages on the connection.

Assuming that the network provider supports this option, the RNC is constantly monitoring the COUNT-CRNC value associated to each radio bearer. If this threshold is reached, the RNC sends a Counter Check message which contains the most significant

bits of Count-C of each active radio bearer. The user equipment compares the Count-C value(s) received from the RNC with its local value, computes the difference, if any, and constructs a Counter Check Response message with the differences. If one or more of the values contained in the Counter Check Response message is greater than 0 (null) then the RNC may decide to send a Release Connection message. Otherwise the procedure completes with success. All the messages described above are integrity protected. However, if the (RNC) notices that the received message has been somehow tampered, then according to the specification[25], the RNC may release the connection. Although the behaviour of the system thereafter is actually up to the provider, without doubt, the specification leaves room for DoS situations to occur.

5.1.7 SQN Synchronization

Resynchronization is a procedure done for aligning the value of SQN in the MS and HLR. Since it involves the generation of new AVs (normally in batches) as well as their transmission from the HLR to the SGSN, it is considered a computational intense and communicational heavy procedure. Naturally, an attacker would wish this procedure to be executed simultaneously for large numbers of users, and if possible, repeatedly in order to mainly overstress the HLR. However, one cannot simply modify the SQN_{HN} value of the Authentication Request message, because that messages is protected. Every attempt to spoof the SQN would lead to MAC verification failure in the MS and probably the whole procedure would be abandoned.

Such situations of course, are DoS attacks from a user point of view but such execution limits to individual user level DoS. However, following an alternative methodology, there is a prospect for a rather massive exhaustion of the server's resources. Instead of attempting to modify the Authentication Request message and hope for an Authentication Reject as response, the attacker could eavesdrop on connections and build a database of MS_i, $AuthReq_1, AuthReq_2, ..., AuthReq_n$. After the elapse of a period the attacker would repeatedly replay these messages towards the corresponding MSs. Normally, these messages will pass the MAC verification process but the not the SQN one, thus triggering the Synchronization Failure message to be sent and resynchronization procedure to be initiated towards the HLR. The correct timing the attacker chooses to unleash his attack is important. He may wait in order for the SQN_HN to be considerably old or soon enough in order for the SQN_HN to be contained in the array of recently received SQNs in the USIM.

What is more, Extensible Authentication Protocol (EAP)-AKA [73] authentication method used for WLAN/UMTS interworking also makes use of AV. As usual, if the

received SQN_{HN} is in the incorrect range the MS should perform the SQN synchronization procedure. Here the situation is worse because the communication penalty in terms of network signaling is increased. This is because the cost for accessing HLR is expensive, especially when AAA, SGSN and HLR are located in different countries. That is, the AAA server in the visited domain must notify the HN and request fresh AV from the HN's HLR once more. Therefore, leaving aside the additional overhead caused to the involved entities, another penalty is the bandwidth consumption between the AAA server and the HLR.

The synchronization attack described in this subsection is feasible mostly due to weaknesses spotted in the UMTS specification itself. At its current form the UMTS architecture in order to protect against reply attacks in AKA procedure leaves room for DoS attacks. A counter value indicating the number of failed authentication attempts at the MS side can be a valuable tool for both the MS (for avoiding frequent resynchronization attempts) and the HLR (for the same reason and for extracting conclusions and taking appropriate measures).

5.2 Attacks Against WLAN/UMTS

In this section we consider some novel attacks that originate from the way Extensible Authentication Protocol (EAP)-AKA operates [73]. EAP-AKA [73] has been specified for achieving access control integration in hybrid UMTS/WLAN network realms.

Extensible Authentication Protocol (EAP) [74] provides a universal authentication framework that is frequently used in wireless LANs, MANs, and cellular networks. EAP is not an authentication mechanism per se, but it offers a series of generic functions and a negotiation process based on the preferred authentication mechanism between two parties.

5.2.1 EAP-Response/AKA-Client-Error Notification

Spoofing notification messages is a straightforward process which according to [73] involves changing the most significant bit of the notification code. This bit is a 16-bit number, which is called the Success bit (S bit) and specifies whether the notification designates failure. By definition, the peer uses the client error code 0, i.e., "unable to process packet", while the server employs one of the general failure codes ("General failure after authentication" (i.e., error code 0) or "General failure" (i.e., error code 16384) depending on the phase of the EAP-AKA exchange). When receiving a notification code

with these respective values, the server must issue an EAP-Failure packet. Analogous alternations must be held to the Phase bit (P bit) which is the second most significant bit of the notification code. This bit designates at which phase of the EAP-AKA exchange the notification is issued. For example, if the P bit is set to 1, the notification can only be issued prior to the EAP/AKA-Reauthentication round in re-authentication or before the EAP/AKA-Challenge round in full authentication indicating various failure cases.

If the peer detects any other error in a received EAP-AKA packet, it issues an EAP-Response/AKA-Client-Error message with error code 0. Specifically, this error code is used in various cases, e.g., "The peer encountered a malformed attribute", "unrecognised or unexpected EAP-AKA Subtype in the EAP request", "the peer is not able to parse the EAP request" etc. It is stressed that none of the aforementioned peer notifications/messages (EAP-Response/AKA-Authentication-Reject, EAP-Response/AKA-Synchronization-Failure, EAP-Response/AKA-Client-Error) is protected (authenticated) by an AT_MAC attribute. Therefore, these messages could be exploited by an attacker in several stages of the EAP-AKA process. For instance, the attacker could spoof an EAP-Response/AKA-Client-Error message and sent it to the EAP-Server in order to fool him into halting the protocol.

5.2.2 EAP-Response/AKA-Synchronization-Failure Resynchronisation

Also, it could spoof an EAP-Response/AKA-Synchronization-Failure notification into forcing the server to trigger the costly resynchronisation procedure.

5.2.3 EAP-Request/AKA-Notification Session Termination

Typically, in situations where the EAP-server detects an error when processing a received EAP-AKA response, it must respond using an EAP-Request/AKA-Notification packet with an AT_NOTIFICATION code that implies failure. Some of the error cases forcing the server to send an EAP-Request/AKA-Notification are: "The server is not able to parse the peer's EAP response", "The server encounters a malformed attribute, a non-recognized non-skippable attribute, or a duplicate attribute", "Unrecognised or unexpected EAP-AKA Subtype in the EAP Response" etc [73]. As with peer notifications EAP-Request/AKA-Notification packet is not protected and can be exploited by an attacker into fooling the client to tear down the protocol session.

5.2.4 EAP-AKA Request HLR Flooding

The EAP-AKA server acquires authentication vectors from the HLR residing in the HN, in a process which is generally both computationally and communicationally intensive. Thus an insider, i.e., a malicious peer, may produce a large number of concurrent and forged EAP-AKA requests to stress the resources HLR. In this case the burden to the network is much larger due to the nature of the processes involved.

5.3 Attacks Against GSM/UMTS

This family of attacks is feasible only if the victim MS is located in the GSM EDGE Radio Access Network (GERAN) coverage of a UMTS network. In such scenarios the attacker aims in inflicting damage to UMTS clients by taking advantage of the flawed GSM/UMTS interconnection processes.

5.3.1 Real Time Eavesdropping

This attack was originally described in [75, 76] and further analysed in [77]. If successful, the attacker will have complete access to the unencrypted traffic of the victim MS, until the next authentication or handover to the UTRAN takes place. Initially, the attacker sets up a BTS supposedly connected to a 3G VLR/MSC. The MS is lured to camp on its radio channels and it is tricked to transmit its IMSI by receiving a user identity request message by the attacker. After this phase the false BTS disconnects from the MS, but simultaneously the attacker sets up a BTS connected to a 2G VLR/MSC, hoping that the victim MS will then camp on these radio channels. In the meantime, the attacker also assumes the role of the MS towards the valid 3G network and by using the security capabilities and the IMSI of the victim MS, it acquires RAND, AUTN and immediately disconnects. In the process, the fake BTS initiates a 2G AKA using the same RAND received from the valid 3G VLR/MSC during the previous step. The fake BTS, accepts the MS but sends cipher mode command with the flawed encryption algorithm, A5/2, as the preferred one. The MS generates the 64 bit K_c but the attacker can easily break the algorithm and derive the key in the way [78] describes. Finally, the fake BTS disconnects the device which freely and without interruption starts authenticating with the valid 3G VLR/MSC. The VLR/MSC will construct the authentication request using the same quintet transmitted to the false BTS. In that way the new K_c will be the same as the one retrieved by the attacker.

5.3.2 Impersonation Attack

The goal of this attack is to steal the identity of an existing MS in a UMTS network and take advantage of its services. Closely resembling the attack described in 5.3.1 in matters of methodology, this attack is also feasible only if the MS is located in the GSM EDGE Radio Access Network (GERAN) coverage of a UMTS network. Initially, the attacker sets up a BTS, connected to a 3G VLR/MSC, and convinces the victim MS not only to connect to it, but also transit its IMSI. While still connected to the victim MS, the attacker poses as the victim MS and attempts to connect to the valid UMTS network. To that end, it initiates a 3G UMTS process to retrieve RAND and AUTN. Having this information the fake BTS proceeds to 3G AKA with the victim MS but during that step it declares that the preferred cipher will be A5/2. Finally, the attacker breaks this algorithm to derive the K_c which she can use to authenticate to the real network.

Chapter 6

Assessment of Wireless Attacks

In this chapter we attempt to evaluate some of the most important attacks described in chapters 3, 4 and 5 by providing a quantitative assessment. The attacks included in this chapter were chosen because: (a) it is possible to evaluate them against some quantitative characteristics, (b) their impact is highly bounded to these characteristics, and (c) they may be representative for a broader category of attacks. Due to the nature of the attacks, some of the evaluations were based on simulations while in other cases the evaluation results were extracted from experiments which were conducted under realistic conditions, using the appropriate equipment. In these cases, the devices employed were a Nokia Lumia 800, an iPhone 2, a Samsung Nexus smartphones, a Samsung Galaxy Tab tablet, as well as two desktop PCs with a Linksys WUSB54GC and D-Link DWA-125 wireless USB adapters running Ubuntu Linux 12.04 and Windows 7 respectively. Standard wireless penetration testing tools were employed such as the Aircrack suite [79] and the MDK3 [80] tool. Whenever specific attacks were not offered by any publicly available tool, custom scripts for launching these attacks were implemented. For example Probe Request Flooding attack was fired with File2air tool [81] (using the Lorcon-old library [82]) while the Fake Power Saving and the Disassociation attacks were unleashed by custom C programs implemented using the Lorcon2 library [83]. Hereunder, the most important of the conclusions are denoted.

A risk analysis for each one of the described attacks is also appended. Tables 6.1 and 6.4 gathers and presents the attacks that have been described in the corresponding sections. We analyse and evaluate the severity of each attack according to a modified version of the methodology presented in [84], [85] (which in turn is a stricter version of a methodology developed by ETSI [86]). Specifically, we classify attacks according to the risk they impose to the studied system as: Major, Moderate, Minor. This classification is done by taking into account two factors:

- *Likelihood of Occurrence* - This criterion indicates the possibility of an attack to be implemented by exploiting vulnerabilities of the system. The attack is considered unlikely if its implementation cost is high, major technical obstacles exist, or the risk of the attacker to be exposed is high. An attack is possible if the cost of the attack as well as the risk of exposing herself are moderate and the technical difficulties are solvable. An attack is likely if the associated costs and risks for the attacker are low and there are no technical difficulties associated with the attack.
- *Impact Upon the System* - This criterion is an indicator for the possible consequences to the system, provided that the attack succeeds. The attack is considered to have low impact if it affects small number of users, for a short amount of time and simply generates commotion to the system. An attack is considered of medium impact if it succeeds to afflict loss of service, affects a larger number of users but still its consequences are reversible. An attack is considered of high impact if it affects a large number of users for a significant amount of time and causes financial losses for the provider or loss of privacy/confidentiality for a user.

To formalize the aforementioned model we assume the following sets:

$C = \{E_x, Ma, In\}$ represents the Cost of the attacker with Ex being Expensive, Ma being Manageable, and In being Inexpensive.

$D = \{Ha, So, Ea\}$ represents the Difficulty to implement the attack with Ha being Hard, So being Solvable, and Ea being Easy.

$R = \{Hi, Mo, Lo\}$ represents the Risk for the attacker associated with this attack Hi being High, Mo being Moderate, and Lo being Low.

$T = \{Sh, Ln\}$ represents the Time span of the attack with Sh being Short, and Ln being Long.

$S = \{Sm, Me, La\}$ represents the Population of users affected by the attack with Sm being Small, Me being Medium, and La being Large.

$O = \{A, DoS, LoP, ToS\}$ represents the Outcome of the attack with An being Annoyance, DoS being Denial of Service, LoP Loss of Privacy and ToS being Theft of Service.

Table 6.1 contains an evaluation of all the attacks discussed in the process of this work according to this model. For all possible Threat, Likehood and Impact values with respect to the aforementioned characteristics the reader should consult appendix C.

6.1 Theoretical Evaluation

6.1.1 Energy Consumption of MOB-TRF-IND Water Torture Attack

In order to evaluate the amount of energy consumption caused by attack described in section 4.2.1 we have proceeded to an analysis which involves the 3 following scenarios:

- An MS does not support Sleep mode but does not send or receive any traffic for a given period of time.
- An MS does support Sleep mode and does not receive any traffic for the same period of time
- An MS which supports Sleep mode is under the attack described in section 4.2.1.

The energy consumption for each of the 3 scenarios respectively can be modelled in equations 6.1, 6.2 and 6.3.

$$E_{AVG} = \begin{cases} \frac{T_{T_x}(E_{T_x}+E_A)+T_{R_x}(E_{R_x}+E_A)+T_S^i E_s+T_L E_L+T_{LR_x}E_{R_x}}{T_{T_x}+T_{R_x}+T_S^{min}+T_L+T_{LR_x}} & \text{if } 0 \geq i \leq n \\ \frac{T_{T_x}(E_{T_x}+E_A)+T_{R_x}(E_{R_x}+E_A)+T_S^m axE_s+T_L E_L+T_{LR_x}E_{R_x}}{T_{T_x}+T_{R_x}+T_S^{min}+T_L+T_{LR_x}} & \text{if } i > n \end{cases} \tag{6.1}$$

$$E_{AVG} = \frac{T_{T_x}(E_{T_x}+E_A)+T_{R_x}(E_{R_x}+E_A)+T_S^m axE_s+T_L E_L+T_{LR_x}E_{R_x}}{T_{T_x}+T_{R_x}+T_S^{min}+T_L+T_{LR_x}} \tag{6.2}$$

$$E_{AVG} = E_A \tag{6.3}$$

where E_AVG is the average amount of energy consumed, T_{T_x} is the time required for transmitting a packet, E_A is the energy consumed in awake mode, E_{T_x} is the energy consumed for transmitting a packet, T_{R_x} is the time required for receiving a packet, E_{R_x} is the energy consumed for receiving a packet, T_S^{min} is the smallest possible time window of unavailability, T_S^{max} is the maximum window of availability, E_S is the energy consumed during unavailability interval, T_L is the time window of availability without performing any operation, E_L is the energy consumed during availability interval without performing any other operation, T_{LR_x} is the time required to receive a message during availability interval, and E_{LR_x} is the energy required for receiving a message during

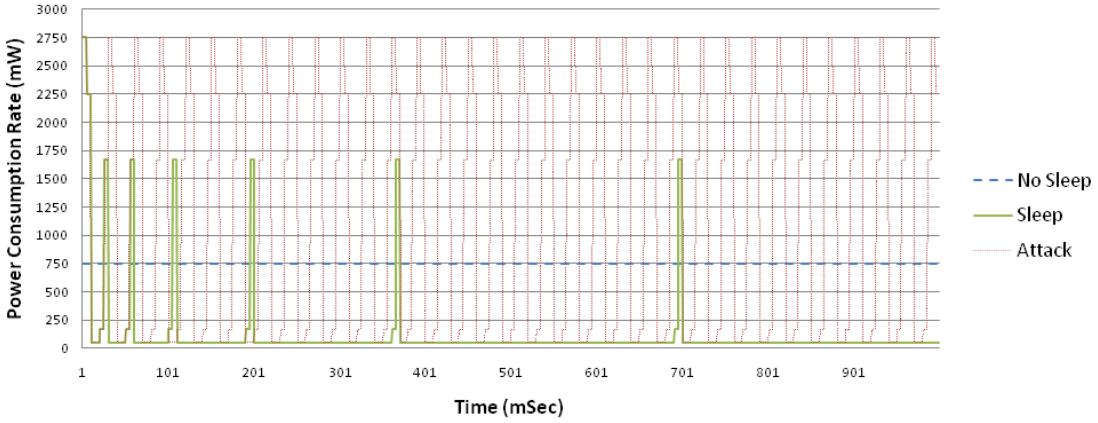


FIGURE 6.1: Snapshot Energy Consumption Under MOB-TRF-IND-Water-Torture Attack

availability interval. All time units are counted in msec, while all energy units are counted in mW.

Based on the energy values found in [87], the energy values for transmitting and receiving of a popular commercial device, as well as the values for time parameters retrieved from [23], we have proceeded to a simulation with the following: $T_{T_x} = 5$, $T_{R_x} = 5$, $T_S^{min} = 10$ (2 frames), $T_S^{max} = 5120$ (1024 frames), $T_L = 5$, $T_{LR_x} = 5$ and $E_A = 750$, $E_{T_x} = 2000$, $E_{R_x} = 1500$, $E_S = 50$, $E_L = 170$. In our experiments we assume that the network is operating in OFDMA/TDD with 10Mhz bandwidth. The frame duration is 5msec and for simplifying the calculations all three scenarios assume that packets are transmitted to the MS immediately and there is no delay. Also, the energy consumed for other operations of the MS (those relevant to the operating system for example) are neglected. Figure 6.1 presents a snapshot of the instantaneous current consumed for each of the three scenarios during the first second of operation, while figure 6.2 illustrates the average energy consumption.

The results of the analysis indicate that by unleashing a MOB TRF-IND Water Torture Attack, the attacker will be able to achieve an energy consumption rate which surpasses that of an MS with no Sleep mode support for over 54%. More specifically, the average power consumption is 750 mW for the case where the MS does not support Sleep mode (scenario a), only 51.7 mW for the case where Sleep mode is enabled (scenario b), and 1156 mW for this last case, i.e. the attack takes place. We can easily deduce that the energy consumption during the first scenario would deplete a battery of 1400 mAH 3.7 V (such as the ones equipped by contemporary smartphones) in 6.9 hours. For the second scenario the same battery would be drained in 100 hours under normal conditions, while under attack the battery would be depleted in about 4.48 hours. This can prove quite annoying for users of handheld devices for example, while it is doubtful if it will cause

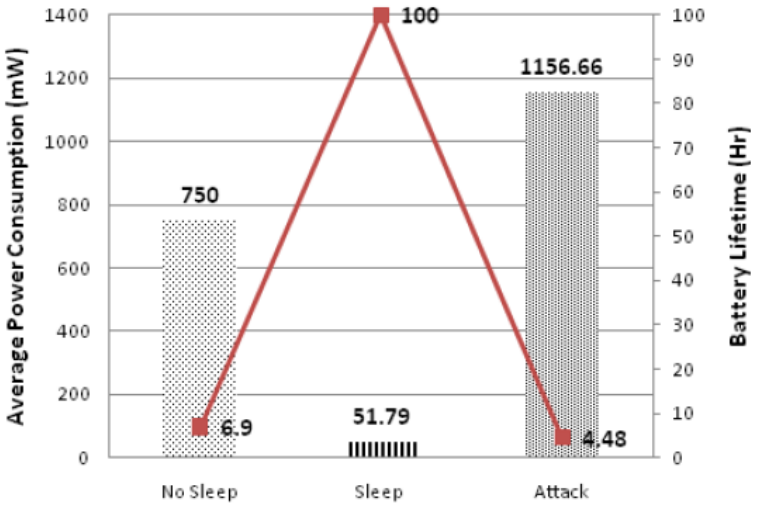


FIGURE 6.2: Average Energy Consumption Under MOB-TRF-IND-Water-Torture Attack

disturbance to users of larger energy capacity devices such as laptops. While these values may not reflect realistic discharge rates they are indicative of the impact of the attack.

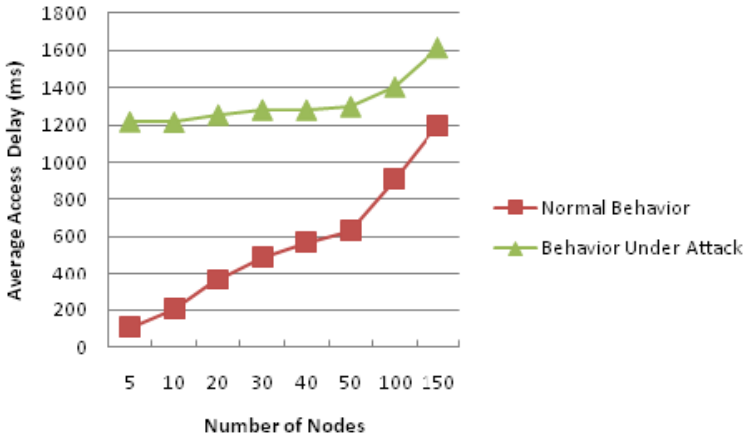
6.1.2 Degradation of Service from RNG-REQ DDoS Attack

For evaluating the impact of RNG-REQ DDoS Attack, the following scenario is considered: A number of MSs which has arrived since the last UCD transmission receives a new UCD message at instance 0 so all MSs are cleared to enter contention for initial ranging process. We consider this simulation for just a time frame as big as the UCD interval (5 sec) but the aggressor unleashes her attack only during the first second of the ranging process. This actually is an interval most likely be chosen in real attack conditions since the Back Off (BO) window size is still small and the collision probability is quite large. During this interval the attacker is transmitting an RNG-REQ message on every single transmission opportunity of every frame.

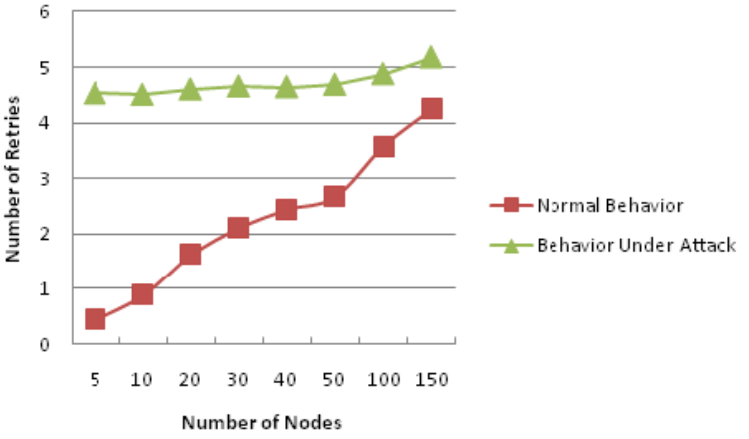
For this simulation scenario we evaluated the initial ranging process in normal operation as well as under attack. More specifically, the behaviour of the network in matters of access delay and number of retries is considered under different number of contenting mobile nodes. For the simulation experiment the following assumptions have been made: frame duration of 5 msec, initial BO window 8, final (maximum) BO window 1024, UCD interval 5 sec, T3 200 msec, simulation duration 5 sec.

The attack causes all contending MSs to collide and as a result to progressively set their backoff window to a very high interval. This has an immediate effect in the access delay thereafter. Still, the total number of RNG-REQ messages transmitted by the attacker

in the 1 sec period of attack is not more than 600 messages with total traffic about 96 Kbps (assuming that the RNG-REQ message is 20 bytes). If there is a number of collaborating attackers this value per user can become even smaller. This makes it even harder for deployed defence mechanisms in the BS (such as Intrusion Detection Systems) to become alerted of this abnormality. Figures 6.3a and 6.3 illustrate the delay and number of retries an MS has to make in both scenarios.



(A) Average Access Delay



(B) Ad-Hoc Mode

FIGURE 6.3: Average Connection Retries

One can notice that when the attack is unleashed against 5 contending nodes (this corresponds to an arrival rate of 1 node per second) becomes comparable to that of 150 contending nodes (arrival rate of 30 nodes per second) in normal conditions which justifies our classification of this attack as major. At this point the reader should notice that the TBEB algorithm is also part of the bandwidth request mechanism. Therefore, attacks such as RNG-RSP DoS, Signalling DoS (in the unlikely event of success), PKM-RSP: Auth Invalid DoS, Secure LU DDoS as well as MCA-REQ DoS cause very similar results as the one investigated in this section. In most cases such attacks will force many MSs to disconnect simultaneously. Naturally, after that this large number of MSs will

attempt to reconnect performing Initial Network Entry. Eventually, this will result in a large number of MSs contending for a small number of TO in the Initial Ranging step, which is actually the bottleneck of the whole Initial Network Entry process.

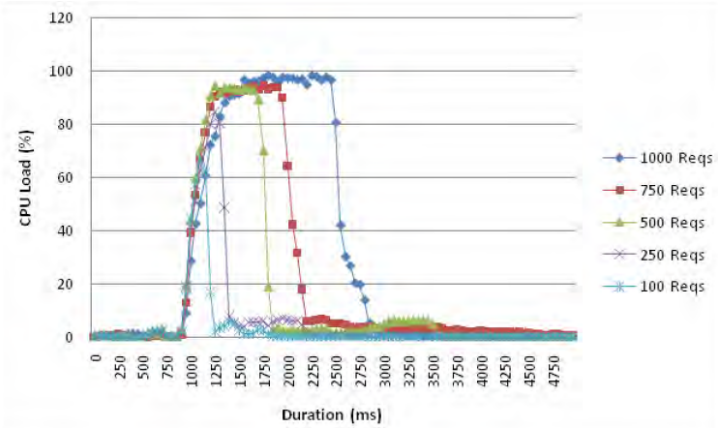
[88] conducts similar experiments on the ns-2 simulator and concludes that parameters of the Initial Ranging step should be considered critical for system security as a possible inaccurate setting may lead to serious DoS attacks or poor system performance.

6.1.3 Computational Burden of AUTH-REQ Replay DoS Attack

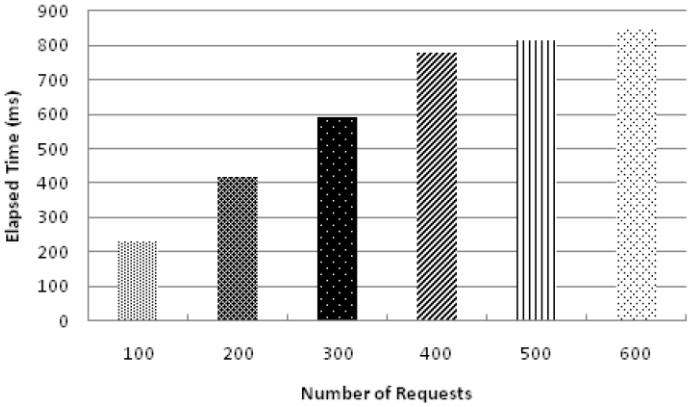
To evaluate the AUTH-REQ Replay DoS Attack we considered the situation where different number of nodes perform the attack described in section 4.5.3 against a specific BS. We monitored the amount of CPU load imposed to the system as well as the total amount of time that is required from the BS to serve all the requests. Our purpose is to evaluate the computation burden of this attack and attempt to estimate the amount of client requests needed to (over)stress the BS. The number of Auth-Req considered starts at 100 and scales up to 1000 messages. The maximum amount of Auth-Req messages (1000) reflects the value of 10% of the number of maximum simultaneous connections supported by the stateof-the-art BS equipments today [89]. This, apparently small, percentage is a rather realistic attack condition as the attacker must first eavesdrop and create a database of valid Auth-Req and then make sure that the corresponding CIDs are still active.

The experiments were conducted in a custom made simulation environment written in C++ and tested on a Windows 7 (64 bits) Intel Core i7 2.80 Ghz machine incorporating 4 GB of RAM memory. Modern BS equipment is expected to have similar computational power and have analogous performance. Figures 6.4a and 6.4b present the CPU load and delay (in terms of service times) respectively.

From the experimental results it is obvious that a significant penalization to the system for a considerable amount of time happens only for more than 500 simultaneous requests. In this case, the CPU load peaked at 94.24% and remained at high levels of 69.3% average for 814 msec. In the case of 1000 simultaneous requests the CPU load peaked at 98.4% and remained at high levels of 77% average for about 2 seconds (2050 msec). Typically, BSs that support a large amount of simultaneous connections are expected to incorporate a stronger CPU than those that support a smaller number of connections. Generalising this empirical study we could conclude that the AUTH-REQ Replay DoS Attack can be fruitful for the attacker only if she is willing to invest time and effort to eavesdrop over a number of Auth-Req messages of at least 5% of the simultaneous connections the victim BS can support.



(A) System CPU load during an Auth-Req Attack



(B) Total Amount of Time Required to Serve All the Auth-Req Messages

FIGURE 6.4: CPU Load and Delay Caused by Auth-Req Messages

6.1.4 IV's Required in WEP Cracking Attacks

Most of the documented WEP cracking attacks are based on some kind of statistical observations of a network's traffic, however the amount of traffic needed to actually crack the key is non deterministic. The basic characteristic of all these methods is that they require a large number of IVs which may be obtained by monitoring the traffic for encrypted data frames (such as ARP or IP packets). Such attacks can be absolutely passive and in this way totally untraceable. However, in practise, this is rarely the case as attackers inject traffic to the network (usually ARP packets) to trigger responses (enforcing the generation of new IVs), thus speeding up the process and making such attacks practical. Actually, several techniques including ARP amplification or double ARP amplification exist to even further boost the generation of IVs. Table 6.2 summarises the estimated amount of IVs required for successful cracking by popular attacking methods, based on statistical observations.

TABLE 6.2: Average IVs required for WEP cracking by various attacks

Attack	IVs (average)	Success	Year
FMS	5,000,000	50%	2001
KoreK	700,000-2,000,000	50%	2004
PTW	40,000-500,000	50%-95%	2007
VX	32,700	50%-95%	2007
Modified PTW	24,200	50%-95%	2008

In any case, to offer a clearer view of the amount of IVs required versus the amount of IVs generated through everyday traffic we conducted several experiments on different use case scenarios. More specifically, we calculated the average amount of IVs per minute generated by applications such as video streaming, moderate web page browsing, file downloading, as well as intrusive scenarios such ARP injection attacks. All scenarios assume having one client connected to the examined network. Figure 6.5 summarises the results obtained per application.

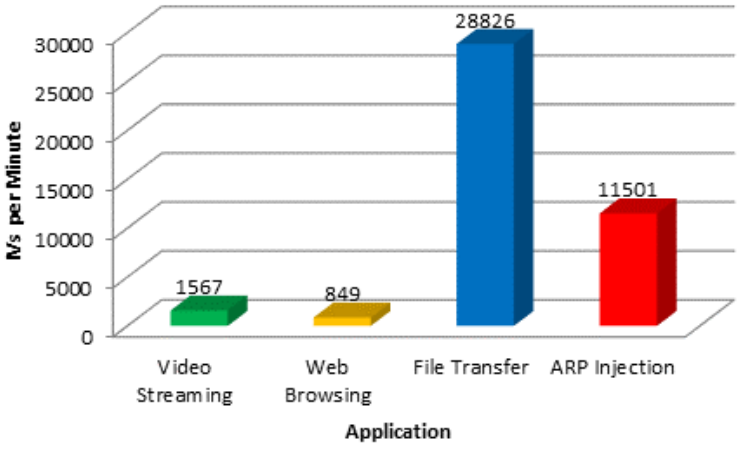


FIGURE 6.5: Average amount of IVs per minute generated by various applications

From the figure it becomes obvious that in networks with low to moderate load the attacker must wait several hours (for the case of FMS) to several minutes (for the case of PTW) to gather the appropriate amount of IVs that will allow her to unveil the key.

6.2 Practical Evaluation

6.2.1 Loss of Connection with Deauthentication and Disassociation Flooding

As already mentioned in section 3.0.5.1 Deauthentication flooding attack is the most popular DoS attack in 802.11 networks.

In our experiments we used a range of different devices and measured the elapsed time from the moment a Deauthentication frame is sent by an aggressor to a victim STA until the STA gets fully re-associated to the AP. We noticed that in most of the cases, these cycles are non-neglectible (e.g., greater than one second). This dictates that a relatively small number of packets per minute is enough to significantly disrupt a victim's communication if not cause a complete DoS. On the one hand, this conclusion is contradictory to the common practice of the most popular modern wireless injection tools (Aircrack suite, MDK3) that aggressively transmit hundreds of Deauthentication frames per second. On the other hand, judging by the experimental results in [90] (which was published in 2003) we can assume that manufacturers tend to construct NIC cards which complete the re-authentication cycle faster.

For the Deauthentication attack we relied on the Aircrack suite but due to the lack of support of a pure Disassociation attack by any of the existing penetration tools the attack was launched by a separate self implemented tool. By comparing the deauthentication cycles with the disassociation ones we noticed that the latter are noticeably greater. This conclusion contradicts to our initial hypothesis that the disassociation cycle is briefer because of the less actions involved. In practice, such cycles are longer due to the fact that upon receiving a disassociation frame the STAs will first issue a deauthentication frame to the AP and then go through a complete authentication and association/re-association cycle. This behaviour is not according to the standard but has been observed for all of our test subjects. In this way, it is made easy for the attacker to substantially disrupt operations such as web browsing, app downloading, voip calling, video streaming even for the most potent of the devices with as few as 100 frames per minute. Figure 6.6 compares the cycles for a variety of wireless devices.

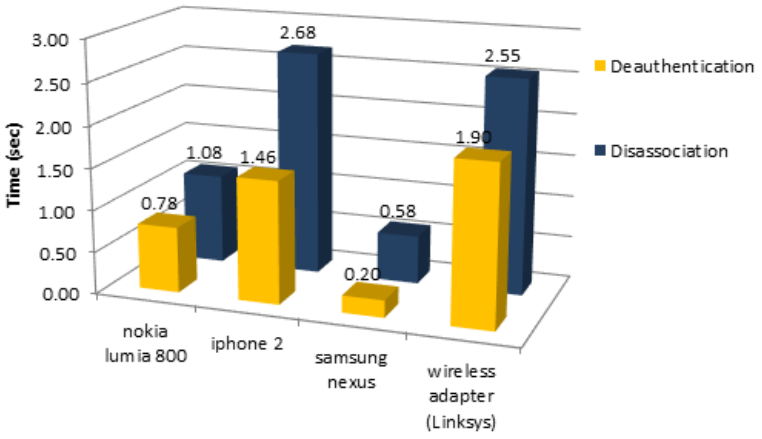


FIGURE 6.6: Deauthentication vs. Disassociation Cycles For Several Devices

It is worth mentioning that the use of WPA over WEP didn't have any substantial impact on the re authentication/reassociation cycle.

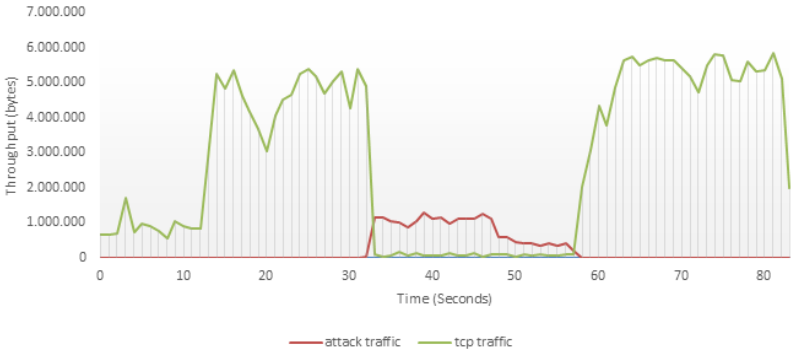
6.2.2 Reduction of Throughput with Probe Request Flooding

During all the experiments considering this attack we did not notice actual DoS against any number of users of the network. However, what was apparent was annoyance in the form of reduced throughput. While the theoretic ground of this attack is based on the goal of exhausting the physical resources of an AP, according to our experiments the main cause of commotion comes from the signalling overhead imposed on the wireless medium. It must be made clear that a single probe request frame triggers multiple responses from AP's in the vicinity simultaneously. Thus, it is easily understandable that the more APs exist in the neighbourhood the more effective the attack gets.

We believe that it is much more realistic for an attacker to cause havoc to a network in this way rather hoping for driving a contemporary AP (even a low-end home device) to its physical limitations and to force it to drop clients. Our experiments were conducted with a custom-tailored version of the File2air tool (using the Lorcon-old library) that allowed us to send 5,000 Probe Request packets in total with variable MAC address fields (all corresponding to existing manufacturers) at variable rates. We evaluated the results in both TCP (FTP file transfer) and UDP (Skype call) application scenarios. More specifically, in the UDP scenario we noticed that the throughput dropped from 145Kbps to 68Kbps and in the TCP scenario from 2Mbps to 269Kbps. This is translated to a loss of 53% and 87% respectively. Figure **??** shows the effects of probing attack to throughput under TCP and UDP scenarios.

6.2.3 Denial of Network Entry with Beacon Flooding

As we already mentioned, this attack comes in two flavors: (a) transmitting beacons that advertise non existing ESSIDs, and (b) transmitting beacons that all advertise an existing ESSID, but correspond to different (non existing) BSSIDs. The first case does not cause real DoS but may prove a factor of commotion. Actually, the eminence of nuisance depends upon the patience of a user locating the network of interest in a (unusually large) list of ESSIDs, most of which have random (thus meaningless and unusual) names. For the second variation of this attack we executed our experiments with the use of MDK3 by injecting Beacon frames that advertise the same ESSID as the legit AP but with different (random but corresponding to existing manufacturers) BSSID. This attack successfully prevented the entry of new clients to the network for all hand-held devices except Samsung Nexus. For the laptop machines the ones equipped



(A) Drop of Throughput in TCP Scenario

(B) Drop of Throughput in UDP Scenario

FIGURE 6.7: Effect of Probe Request Flooding Attack in Throughput

with Windows 7 OS seemed to be immune to this attack. Still, the attack was successful against the Linux equipped machine. As expected, this type of Beacon flooding attack had no success with the already connected devices

6.2.4 Stressing the AP Resources with Authentication Flooding

The theoretic foundation of this attack lays in an attempt to exhaust the physical resources of the APs and then possibly force it to collapse. Results from real life experiments however, indicate that even low-end contemporary devices can effectively cope with this threat. Actually, even after 8 million authentication attempts there wasn't any noticeable deviation from the AP's normal behaviour (i.e., freeze or reset). However, we noticed that during the course of this attack even in its the early stages (i.e., the first two seconds) the client was unable to perform authentication and enter the network. More specifically, all devices presented such behaviour with the sole exception that of Samsung Nexus which was able to connect but with a noticeable delay. This attack may pose as a more effective equivalent of the Beacon Flooding attack. The above mentioned experiments were conducted with the use of the MDK3 tool with an average injection rate of 900 authentication frames per second.

6.2.5 Packets Replayed with ChopChop

We conducted our experiments with the Aireplay-ng tool (of the Aircrack suite). In the course of the attack we replayed packets of different sizes. We came to the conclusion that the amount of time required for the ChopChop method to fully analyse a given packet depends on the actual size of the packet. Some examples of various packet sizes and the corresponding requirements in number of packets to be injected and amounts of time are given in table 6.3.

TABLE 6.3: Requirements in Number of Frames and Time for ChopChop Attack

Size	Frames Injected	Total Time
70	6550	131
80	9445	187
122	13255	264

From the table it is clear that a significant amount of packets needs to be replayed back to the AP for the ChopChop to complete successfully. However, traffic even of such magnitude can be camouflaged in busy networks if the packet size is the only criterion of detection. On the other hand, the replayed packets will have several fields identical, including the IV one. While fields such as IV are randomly selected and as such they are subject to possible repetitions it is highly unlikely that the IV field of numerous packets in a short amount of time, say 1 sec, will be identical. The following table 6.4 presents a summarizing overview and evaluation of the attacks against 802.11 networks.

TABLE 6.1: Evaluation of WiMAX Attacks

Attack	Threat	Cost	Difficulty	Risk	Duration	Size	Outcome	Protocol
RNG-RSP DoS Attack	Major	Expensive	Easy	High	Long	Large	DoS	802.16-2009
RNG-RSP Downgrading Attack	Minor	Expensive	Solvable	High	Long	Medium	Annoyance	802.16-2009
RNG-RSP Water Torture Attack	Minor	Expensive	Easy	Moderate	Long	Medium	Annoyance	802.16-2009
RNG-REQ Downgrading Attack	Minor	Expensive	Hard	Moderate	Long	Medium	Annoyance	802.16-2009
RNG-REQ DDoS Attack	Major	Inexpensive	Easy	Low	Long	Large	DoS	802.16-2009
MOB ASC-REP DoS Attack	Minor	Expensive	Solvable	Moderate	Long	Medium	Annoyance	802.16-2009
Signaling DoS Attack	-	-	-	-	-	-	-	-
MOB TRF-IND Water Torture Attack	Minor	Expensive	Easy	High	Short	Medium	Annoyance	802.16-2009
BR and UL Sleep DoS Attack	Minor	Inexpensive	Hard	Low	Short	Large	Annoyance	802.16-2009
Secure LU DDoS Attack	Major	Inexpensive	Easy	Low	Long	Large	DoS	802.16-2009
MOB NBR-ADV Downgrading Attack	Minor	Expensive	Hard	High	Long	Small	Annoyance	802.16-2009
MOB NBR-ADV DoS Attack	Minor	Expensive	Hard	High	Long	Small	Annoyance	802.16-2009
SBC-REQ Security Downgrade Attack	Minor	Expensive	Hard	Low	Long	Small	Loss of Privacy	802.16-2004
FPC Downgrade Attack	Moderate	Expensive	Solvable	Moderate	Long	Large	Annoyance	802.16-2009
FPC Water Torture Attack	Minor	Expensive	Easy	Moderate	Long	Medium	Annoyance	802.16-2009
RES-CMD DoS Attack	Minor	Expensive	Hard	Moderate	Short	Small	Annoyance	802.16-2009
DBPC-REQ DoS Attack	Minor	Expensive	Hard	Moderate	Long	Medium	Annoyance 8	02.16-2009
Interleaving Attack	-	-	-	-	-	-	-	-
AUTH-REQ Replay Theft of Service Attack	-	-	-	-	-	-	-	-
AUTH-REQ Replay DoS Attack	Moderate	Manageable	Solvable	Low	Long	Medium	DoS	802.16-2009
PKM-RSP: Auth-Invalid DoS Attack	Major	Expensive	Easy	Moderate	Long	Large	DoS	802.16-2009
TEK Reuse Attack	-	-	-	-	-	-	-	-
DES CBC IV Attack	Minor	Manageable	Hard	Low	Short	Small	Loss of Privacy	802.16-2009
DES CBC Insecurity Attack	Minor	Manageable	Hard	Low	Short	Small	Loss of Privacy	802.16-2009
GTEK Update Mode DoS Attack	Moderate	Inexpensive	Easy	Low	Long	Medium	DoS	802.16j
GTEK Theft of Service Attack	Major	Manageable	Easy	Low	Long	Medium	Theft of Service	802.16-2009
MCA-REQ DoS Attack	Major	Expensive	Easy	Moderate	Long	Large	DoS	802.16-2009
Malicious Sponsor Node Attacks	Minor	Inexpensive	Hard	Low	Long	Large	Annoyance	802.16e
PKM-REQ: Auth Request Replay Attack	Major	Inexpensive	Easy	Low	Long	Small	Theft of Service	802.16e
PKM-RSP Replay Attack	Major	Expensive	Easy	Moderate	Long	Moderate	Loss of Privacy	802.16e
OSS Distribution Attacks	Major	Inexpensive	Easy	Low	Long	Large	Theft of Service	802.16e
PKM-REQ: Key Request DoS Attack	Major	Expensive	Easy	Moderate	Long	Large	Dos	802.16e

TABLE 6.4: Summary and Evaluation of Attacks

Attack	Effect	Traffic Injected	Version	Difficulty	Comments	Threat
FMS	Secret Key Cracking	$> 2,000,000$	WEP	Easy	Slow	Moderate
Korek	Secret Key Cracking	$> 700,000$	WEP	Easy	Slow	Moderate
PTW	Secret Key Cracking	$> 50,000$	WEP	Easy	Fast Requires resources	Major
Dictionary	Secret Key Cracking	1	WPA/WPA2	Easy	depends on weak passwords	Moderate
	Keystream Retrieval					
Chopchop	Packet Decryption Keystream Retrieval	i=256*m	WEP	Moderate	-	Moderate
Fragmentation	Packet Decryption	i=16	WEP	Moderate	Reveals up to 64 Slow	Moderate
Cafe Latte	Secret Key Cracking without AP	i=65280	WEP	Easy	not possible against all OSs	Minor
Hitre	Secret Key Cracking without AP	1	WEP	Easy	Fast	Major
Deauth	Loss of Connectivity	High	All	Easy	Can Target Client	Major
Disassociation	Loss of Connectivity	High	All	Easy	Can Target Client	Major
Deauth Broadcast	Loss of Connectivity	High	All	Easy	Affects All	Major
Disassociation Broadcast	Loss of Connectivity	High	All	Easy	Affects All	Major
Block Ack	Annoyance	Low	802.11n	High	Requires Accuracy	Minor
Authentication Request	Inability to join the network	High	All	Low	Ineffective Against Most Devices	Minor
Fake PS	Annoyance	High	All	High	Requires Accuracy	Minor
CTS Flooding	Annoyance	High	All	Low	Can Target Client	Minor
RTS Flooding	Annoyance	High	All	Low	Can Target Client	Minor
Beacon Flooding	Inability to join the network	High	All	Low	Effective Against Limited Devices	Minor
Probe Request	Annoyance	High	All	Low	Affects All	Moderate
Probe Response	Annoyance	High	All	Low	Can Target Client	Moderate
Honeypot	Loss of Privacy	None in the Network	All	Moderate	Relies on Naive Users	Major
Evil Twin	Loss of Privacy		All	Moderate	Requires Knowledge of Secret Key	Major
Rogue AP	Loss of Privacy	None in the Network	All	Moderate	Requires Access to the Wired Network	Major

Chapter 7

Anomaly Detection

7.1 Detecting Anomalies in Data

The term anomaly, originates from the greek word "$o\mu\alpha\lambda o\varsigma$" (omalos) and the prefix "$\alpha\nu$" (an), literally meaning abnormality. The process of identifying anomalies in a gathering of related data is commonly referred to as anomaly detection. This process involves the definition of the portions of the data that represent normal behaviour and the declaration of any observation which does not belong to those portions as an anomaly.

Generally, when analysing data, the exceptional observations that correspond to abnormalities are of equal importance, if not more significant, than the regularities that obey a well-defined notion of normality. Note that anomalies in datasets must not be confused with noise, which can be defined as observations in data which hinder data analysis, since they are of no particular interest to the analysis. Noise removal [91] deletes the unwanted objects before any data analysis is performed on the data so that this process can conclude with higher success rate. Figure 7.1 presents data arranged in a 2D plot. In this example, the data instances form 4 separate regions one of which (C1) is dominant over the others, hence it is considered as normal. Regions C2 and C4 are far away from C1 so they are considered anomalous and they are easily distinguishable. Note that region C3 is fairly close to the normal area C1, so smart techniques must be employed in order to identify the actual membership of candidate data instances.

7.1.1 Basic Aspects

Anomaly detection is a complex process involving multiple parameters. Most of them are problem depended, however a common denominator can be inferred, with respect to its most important aspects. More specifically we can identify:

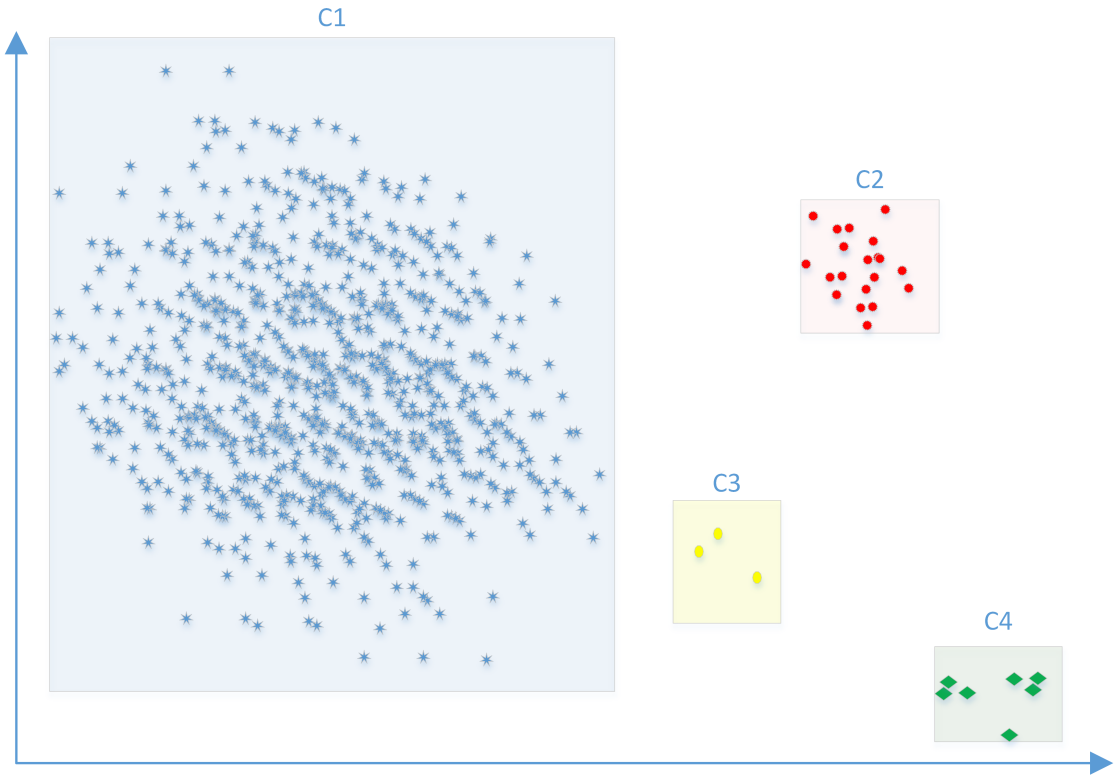


FIGURE 7.1: Normal VS. Anomalous Classes in a Sample Dataset

7.1.1.1 Type of Input Data

Typically, input data are presented to anomaly detectors as a set of data instances. Often described as records, vectors, samples, or observations, these data instances are usually comprised of several attributes. In turn the attributes (frequently referred to as variables, features, or dimensions) can be of different types such as binary (being able to take only one of two possible values), categorical (being able to take only one of a set of predefined values), arithmetic (discrete or continuous), text or date. Data instances may also by semantically connected or develop relationships among them [92]. More specifically:

- Instances may be ordered with respect to time in the dataset. For example, frames received from a monitor node in a network.
- Instances may be sorted with respect to a similarity measure when this relation depends on distance we speak of spatial data and when both distance and time is involved we speak of spatio-temporal.
- Data may be organized as a graph. In a graph data are represented as vertices connected to other vertices with edges. The type of relation among instances is usually specified with the definition of what the edges represent.

Some anomaly detection techniques are able to work with certain types of data. Thus, one of the most important parameters in anomaly detection is the nature of the data that a given technique is required to work upon.

7.1.1.2 Data Labels

Ideally, input data should come with a tag denoting whether they are normal or anomalous. The existence of labelled data instances usually boost the detection accuracy. Realistically, data labelling is not always possible, either due to the high cost of the labelling process, or the sensitive nature of the data. The extent to which the labels are available, influences the choice of anomaly detection techniques to be utilized. Generally, anomaly detection techniques are organized into three big families with respect to whether they are able cope with the absence of pre-labelled data instances or not. More specifically:

- *Supervised* anomaly detection methods build a predictive model based on a set of pre-labelled data, which includes both normal and anomalous instances. By using the resulting model, the detector is able to determine the class of unseen instance.
- *Semi-supervised* anomaly detection techniques are able to detect anomalies after being trained by a dataset that has only the normal instances labelled. Generally, it is much easier to construct a dataset with normal instances as in most situations normality is the rule. On the one hand, this resolves the issue of manually labelling a dataset, but on the other, it is hard to guarantee that there are not abnormal cases included.
- *Unsupervised* anomaly detection techniques do not require labelled data at all. They are able to organize data instances into a (usually predefined) number of groups/clusters with similar characteristics. Some methods go one step further as they implicitly assume that the anomalous instances are fewer than the normal ones, so they label these minority groups as anomalous ones.

7.1.1.3 Nature of Anomalous Data

A critical factor that influences the choice of a respective anomaly detection technique is the nature of the anomalies under consideration. Generally, anomalies are organized into three main categories, which are elaborated below:

- *Point anomalies* - These are typically, individual data instances significantly diverse than those considered normal by the standards of the dataset. This is the simplest type of anomalies.
- *Contextual anomalies* - This type of anomalies is heavily influenced by the values of specific attributes within a specific context.
- *Collective anomalies* - These anomalies consist of instances that are considered normal when they are located in isolation, but are anomalous when they occur together as a collection. Such anomalies may occur when data instances develop some kind of relation with each other. A typical example of collective anomaly is web traffic datasets of normal and DoS attacks. DoS are produced by flooding the server with an extremely large amount of a specific type of packet, which can be tracked sporadically under normal conditions.

7.1.1.4 Result Presentation

In some problems the presentation method is as vital as the accuracy of the detection itself. To this end, methods that incorporate complex data visualisation techniques to describe the results have been developed. More traditional approaches present their results as dots scattered in a 2D or 3D space, as a set of instances with a tag, or in some cases, a tag and a confidence score.

7.1.2 Challenges

The nature of the application domain introduces problem-specific challenges to the anomaly detection process. These challenges may drastically differ from one problem to another however, there exist some which are common to most formulations of the anomaly detection problem. These can be enumerated as:

- Distinguishing between excessive normal and anomalous data - The outliers of normal and anomalous observations may confuse the detection process, especially when these instances have characteristics that are very close or overlap.
- Noise elimination - Typically, noise has a negative impact on the speed of the detection, but in some occasions noise and anomalies share common characteristics which makes distinguishing between the two a tedious process.
- Adapting to variable conditions - The boundaries of normality may shift through time and what is considered normal in a given period may prove anomalous in another.

- Existence of labelled dataset - Labelled data can be used for effectively guiding the construction of prediction models (often referred to as training) or for validating the efficiency of generated models. However, labelling of data is not a trivial task as it often evolves manual labour.

7.2 Basic Anomaly Detection Techniques

The various approaches proposed in literature, have attempted to provide a solution to the anomaly detection problem from different perspectives. Nevertheless, as the context and the parameters of problems may differ drastically, not a single one has been found to act as a panacea. One can speak about techniques that may have the advantage over others in certain types of anomalies. The most important types of anomaly detection methods are described in the following sections. These approaches have been successfully applied in fields like biology and medicine ([93], [94], [95], [96]), engineering ([97]), image processing ([98]), speech recognition ([99]), text processing([100], [101]), as well as intrusion detection ([102], [103], [104]). For a holistic review of anomaly detection methods and applications the reader should refer to [105].

7.2.1 Classification

Classification refers to the problem of categorising new data instances into one of the possible classes with respect to a training set that contains pre-labelled data instances. The classification consists of two phases:

- *Training phase* - during which a classifier develops a model by learning from the available labelled training data
- *Testing phase* - which classifies an instance of unknown class as normal or anomalous using the model produced in the previous step.

A training set is a necessity for classification techniques which in some problems may be impossible to acquire. Moreover, the training set must be adequate in size and accurately labelled for a precise model to be built. Generally, training is much more computationally expensive than testing, since the latter simply uses a pre-constructed model for classification.

7.2.1.1 Neural Networks

Artificial neural networks (ANNs) are computational models inspired by the structure of the central nervous systems of animals, with the goal of mimicking their learning capacity and pattern recognition efficiency. ANNs are generally represented as graphs of interconnected nodes, called "neurons" which are able to compute output values from a set of input values.

ANNs are comprised of several layers of neurons that are interconnected. Thus, in an example ANN with three layers, the first layer has input neurons which simply receive data via its synapses and forwards them to the second layer of neurons, and then via more synapses to the third layer of output neurons. The synapses consist of numerical parameters that are adjusted by a learning algorithm, namely the adaptive weights. Complex problems may require layers of more neurons and/or larger number of layers.

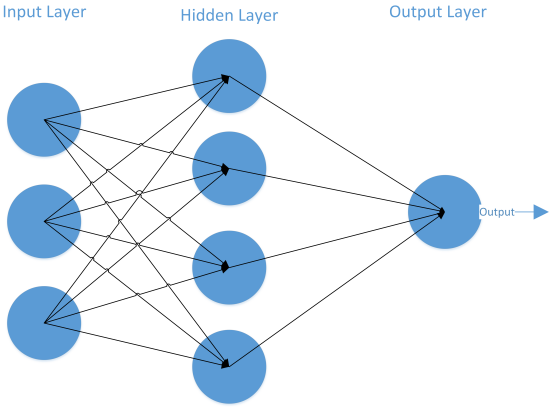


FIGURE 7.2: Example of the Structure of a Neural Network

The basic advantage of ANNs is that they are able to conduct their processing in a non-linear, distributed, parallel and local way and at the same time, have high level of adaptation. ANN are considered one of the best classification techniques and have been used in most anomaly detection areas (including intrusion detection) as the main or in conjunction with other algorithms and techniques [102].

7.2.1.2 Bayesian Networks

Bayesian networks are Directed Acyclic Graph (DAG) whose nodes represent random variables and the edges represent conditional dependencies among them. Unconnected nodes correspond to variables that are conditionally independent of each other. The nodes are associated with a probability function. The input of the probability function is a particular set of values for the node's parent variables. The output is the probability of the variable represented by the node. Bayesian networks and derivatives have been

applied to intrusion detection early on [106], [107], [108], and is considered as one of the dominant techniques in this area.

7.2.1.3 Support Vector Machines

Techniques employing Support Vector Machines (SVMs) [109] can be applied on binary classification problems only. An SVM model represents data instances as points in a 2D space, and maps these points in such a way, that the distance of the nearest points of any class is maximum. Data instances on the margins are called support vectors. New instances are labelled as normal or anomalous based on which side of the gap they fall on [110].

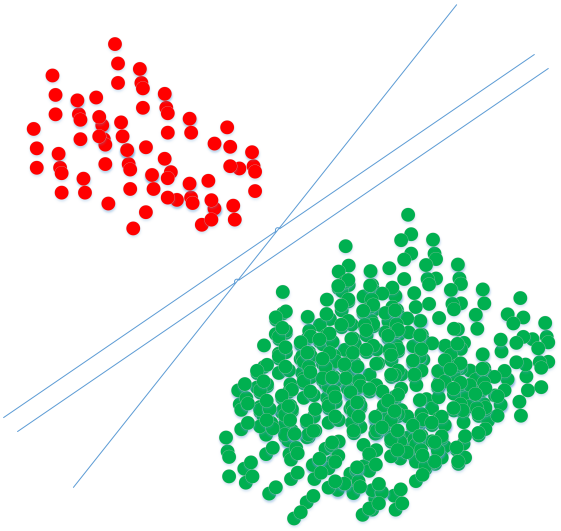


FIGURE 7.3: Example of SVM

7.2.1.4 Decision Trees

Decision Trees is a method which progressively designs tree structures that correlate observations about the attributes of data instances to conclusions about its class. During the learning phase the construction of the model from class-labelled training data takes place, resulting to a flow-chart-like structure. The internal nodes refer to an attribute, the branches represent the outcome of a test, and the leaves correspond to the class. A special family of such techniques, namely the ensemble methods is able to construct more than one decision tree at a time, for increased accuracy. Popular algorithms of this type are ID3 [111] and its extension C.45 [112]. The majority of decision trees algorithms require the attributes of the data instances to take nominal values only. Decision trees methods are popular in intrusion detection [113], but are usually used in conjunction with some other classification method for improved performance.

7.2.2 Nearest Neighbour

The nearest neighbour attempts to locate and retrieve the data instances that are the most similar to a query instance, in given a collection of data. Nearest Neighbour algorithms for anomaly detection capitalise on the observation that normal instances exist in dense neighbourhoods, while anomalies tend to be isolated. A form of distance measure such as Euclidean distance is used as metric of dissimilarity of the instances. One of the benefits of nearest neighbour based techniques is their unsupervised nature. However, nearest neighbour based detectors may perform poorly if normal instances do not have enough close neighbours or if anomalies have enough close neighbours. This is the reason why semi-supervised techniques have been applied to nearest neighbour techniques.

Nearest neighbour based anomaly techniques can be grouped into two main categories according to the adopted measure of dissimilarity:

- Distance to neighbours based - In this category belong approaches that use the distance between a given data instance to the *kth* nearest neighbour, as their anomaly measure. If the instance surpasses a predefined threshold, then it is labelled as anomaly. Alternatively, anomalies may be considered as the *top-n* instances with the largest anomaly scores.
- Density of neighbourhood. In this category belong approaches that use the distance between a given data instance and the *kth* nearest neighbour in the dataset as their anomaly measure. If the instance surpasses a predefined threshold, then it is labelled as anomaly. Alternatively, anomalies may be considered as the *top-n* instances with the largest anomaly scores.

The application of nearest neighbour algorithms is not new in the intrusion detection ecosystem [114], [115].

7.2.3 Clustering

Clustering aims at organizing similar data instances in compact groups. Since a separation occurs with respect to the attributes of data, it is possible to utilize clustering techniques for anomaly detection, if the anomalies are grouped in a different cluster than the normal data. Usually, clustering methods achieve the separation of dissimilar data by calculating the distance of a data instance from another. Once more, distance here refers to a metric such as the Euclidean distance, which scales the dissimilarity between

vectors. Clustering was adopted early on in the field of intrusion detection [116], [104], [117].

Usually, such methods perceive anomalies as the instances which have the following characteristics:

- are not embodied to any formed cluster - Such anomaly detection methods rely on clustering techniques that do not necessarily incorporate all given instances to a cluster.
- are embodied to very small or sparse clusters - The assumption here is that the number of anomalous data instances is much less than the one of the normal ones. Thus, when the size and/or density of a given cluster is below a threshold then all the instances contained it are anomalous.
- are placed in the outer layer of a cluster or outside it - In such cases the anomalies must not form independent clusters.

7.2.4 Statistical Methods

Statistical methods employ statistical analysis for constructing detection models. Some representative examples of both categories are further presented in the following subsections.

7.2.4.1 Gaussian Techniques

Gaussian techniques assume that the underlying data follow a Gaussian distribution. As a first step, these methods estimate the mean of normal instances with regard to one or more of its aspects. The anomalies are determined in a second step by calculating the distance of a new instance from the mean and by checking if that distance exceeds a threshold.

A popular approach relies on the use of box plots, which graphically depicts a summary of the data characteristics including, the minimum and maximum instances, the lower and upper quartile (Q1 and Q3 respectively), as well as the median. Anomalous instances are those whose distance from the box exceeds a threshold. Usually, the lower bound is $Q1 - 1.5 * IQR$ and the upper bound is $Q3 + 1.5IQR$, where IQR is the Interquartile Range (IQR).

7.2.4.2 Regression Techniques

Techniques that fall in this category initially fit a regression model to the dataset and then for each instance the part of the instance which is not explained by the regression model, known as residual, is used to assign an anomaly score to the instance.

7.2.4.3 Hybrid

Such techniques use a mixture of multiple parametric statistical methods to model the data. Normally, they model the normal instances and anomalies as separate parametric distributions. For example, [118] assumes that the normal data is generated from a Gaussian distribution with variance ($N(0, \sigma 2)$). On the other hand, anomalies are generated by the same distribution but with larger variance, $N(0, k2\sigma 2)$. The testing phase involves, determining the distribution in which the test instance falls.

Chapter 8

Nature Inspired Approaches for Network Intrusion Detection

The methods of nature have always been a source of inspiration to humans for problem solving. Yet, it is only until recently, that researchers managed to successfully model and emulate the natural processes in a variety of research fields, ranging from engineering, computer science, economics, medicine and social science. Therefore, it was only a matter of time until such techniques were tested against the intrusion detection sector. The advantages that biology inspired approaches impose to the field of intrusion detection is the basic topic of this chapter.

Initially, we describe the limitations of traditional intrusion detection approaches that led the research community to a quest for novel, unconventional, approaches. Next, we introduce the foundational principles of 3 of the most important bio-inspired approaches. In the process, we conduct a survey of IDS that fall into these categories.

8.1 Swarm Intelligence

The term Swarm Intelligence (SI) was introduced by Beni in the context of robotics system [119]. The methods and algorithms that this research field embraces draw inspiration from the behaviour of insects, birds and fishes, (or more generally, swarm formations of animals in general) and their unique ability to solve complex tasks collectively, although the same thing would seem impossible in individual level. Indeed, single ants, bees, birds and fishes appear to have very limited intelligence as individuals, but when they socially interact with each other and with their environment they seem to be able to accomplish harder tasks such as finding the shortest path to a food source,

organizing their nest, synchronizing their movement and travel as a single coherent entity in high velocities etc. This achievement becomes even more commendable if one takes into account such accomplishment is done without the presence of a centralized authority (e.g., the queen of the hive) enforcing the right actions. Applications of this can be found in NP-hard optimization problems such as the travelling salesman, the quadratic assignment, scheduling, vehicle routing etc.

The unique characteristics of SI establish it as one of the better options amongst the existing ones for intrusion detection. This is due to the fact that SI techniques aim at solving complex problems by employing numerous, yet simple agents, without requiring any kind of a central coordinating authority. Such agents collaborate with each other towards finding an optimal solution for the given task. Their organization occurs naturally via direct or more frequently indirect communication (e.g., by marking their environment). In this respect, agents can be used for carrying out several hard tasks, like finding classification rules, organizing traffic into clusters, keep track of intruder trails etc. In intrusion detection in particular, the self-organizing and distributed nature of these systems is highly desirable as it offers the means to break down a difficult IDS problem into multiple simpler ones and delegate it to parallel-functioning, autonomous units. This potentially makes the IDS autonomous, highly adaptive, parallel, self-organizing and cost efficient.

The systems surveyed in this section are organized primarily according to the adopted SI technique. The three main categories that accrue are: (a) IDS that make use of Ant Colony Optimization, (b) IDS that employ Particle Swarm Optimization and (c) IDS that utilize Ant Colony Clustering. Each class may further be broken down into smaller subcategories leading to the following taxonomy scheme:

- ACO Oriented IDS Approaches
  - ACO for Induction of Classification Rules
- PSO Oriented IDS Approaches
  - PSO & Neural Network Hybrid Approaches
  - PSO & SVM Approaches
  - PSO & K-Means Approaches
  - PSO for Induction of Classification Rules
- ACC Oriented IDS Approaches
  - ACC & SOM Hybrid Approaches
  - ACC & SVM Hybrid Approaches

8.1.1 Ant Colony Optimization

The foraging behaviour of ants and more specifically their unique ability to find the shortest path from their nests to food sources, has inspired the conception of perhaps the most prominent algorithmic model of this kind, namely the Ant Colony Optimization (ACO). Most ant species have very limited or no vision and simultaneously are deprived of speech or any other means of conventional communication. Nevertheless, ants seem to act in a strictly organized manner, which indicates that some sort of camouflaged communication takes place. Indeed, experiments conducted to certain ant species prove that this communication occurs by depositing a substance called pheromone along the path they follow. In more detail, ants initially move randomly in order to locate a food source. As soon as they do so, ants carry food to their nest and deposit pheromone traces along the trail. Subsequently, ants decide which of the available paths they shall follow based on the pheromone concentration deposited on each particular path. As was anticipated, paths with greater pheromone concentration have higher probability of being selected. The insects that follow the shortest path return to their nests earlier and pheromone on that path is reinforced with an additional amount sooner than the one in the longer path. Therefore, the selection among the paths is biased toward the shortest path.

Deneubourg et al. presented the double bridge experiment in which nest and food source were separated by a bridge of two branches of equal lengths [120]. In fact, the authors noticed that the majority of ants will eventually follow only one of the paths but which one is randomly decided. Goss et al. extended the experiment by using paths of unequal lengths [121] showing that in all experiments the majority of the ants will, ultimately, choose the shortest one as shown in 8.1.

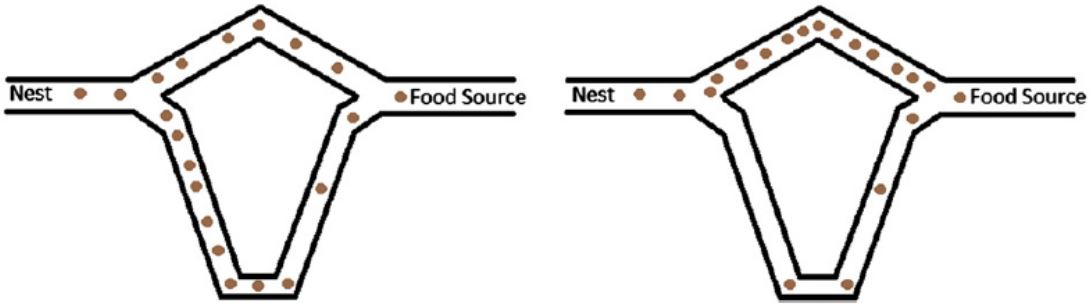


FIGURE 8.1: The Extended Double Bridge Experiment

Dorigo et al. introduced an algorithmic model of the described behaviour for solving minimum cost path problems on graphs known as Simple Ant Colony Optimization (SACO) [122], [123]. In this model ants begin from a source node of a graph $G = (N, A)$ and try to reach a destination node following the shortest path. To each arc (i, j) of a

graph an amount of artificial pheromone is deposited $\tau_i j$. This information can be read and written by the ants to govern their movement to the next node. Specifically, the probability of an ant k located at a node i of choosing j as the next node to be visited is calculated as:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^a}{\sum_j \tau_i^a j} & \text{if } j \in N_i^k \\ 0 & \text{if } j \notin N_i^k \end{cases}$$

Where N_i^k of ant k when in node i contains all the nodes directly connected to i, except the predecessor of i. a is a parameter for controlling convergence speed. When the ant reaches its destination it has to return to the source. In this backward mode the ants deposit pheromone along the trail. Normally, the ant will attempt to follow the same route but if that route contains loops then it must eliminate them first, in order to avoid the problem of self- reinforcing loops. The new amount of pheromone in the arc (i, j) after ant k has traversed it in backward mode is calculated as:

$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau^k$

Pheromone trails evaporate over time. This mechanism can be seen as a way to avoid the problem of convergence to suboptimal paths, or a way to adapt to dynamic graph changes if they ever occur. Pheromone evaporation is simulated by applying the following equation to all arcs:

$\tau_{ij} \leftarrow (1-p)\tau_{ij}, \forall(i,j) \in A$

where $p \in (0, 1]$ is a constant.

8.1.1.1 ACO for Deduction of Classification Rules

Soroush et al. presented one of the first works employing ACO as an efficient method for intrusion detection and more specifically the inference of classification rules [124]. Their proposed system was based on the novel rule extracting algorithm Ant-Miner [125]. The authors adjusted the Ant-Miner algorithm to cope with high dimensional, high volume data, such as the ones usually involved in intrusion detection. Ant-Miner itself is inspired by the foraging behaviour of ants in order to classify numerical data to one of some predefined classes. In particular, this algorithm relies to ants to construct a set of candidate rules in parallel. The rules are of the type:

$IF(term_1 term_2 ... term n_t) THEN class_c$

In this case $term_i$ is formed by three parts, (a) an attribute of the dataset, (b) an operator, and (c) a value. The quality of these randomly generated candidate rules is evaluated against the training set by considering the confusion matrix of real and predicted instances with respect to the training set. During this process the pheromone trails increase for the terms used in the generated rules, in a proportional way to the fitness of that rule. At the same time they decrease for all the rest of the terms (evaporation). Among the generated rules in this step the best one is selected and added to the discovered rules set. This is done iteratively until an adequately large base of rules is constructed. These rules are used later on in test sets as criteria for discriminating network connections into intrusive or normal.

Similarly Junbing et al. propose an Ant-Miner based classification system [126]. Its main differentiation is the introduction of multiple ant colonies. The authors noticed that the algorithm's efficiency might be pushed back, in the case where ants searching for best rules of a class A, have been mislead by the pheromone trails deposited at a prior time, by ants searching for rules of a different class. In this case, each class is handled by different ant types organized into colonies. In this way, ants that belong to a colony deposits a distinct type of pheromone which applies only to ants belonging to the same colony. Colonies are searched in parallel to finally discover one rule per colony. As a final step, the rule with the best quality is selected and added to the rule set.

Fork [127] is an IDS optimized for the special requirements of ad-hoc networks. Due to the resource constrained nature of these terrains, the nodes may produce intrusion detection requests if are incapable of meeting the intrusion detection requirements at a specific time. Thus, the capable nodes are allowed to compete according to an auctioning system for satisfying these requests. The engine that powers Fork is also based on Ant-Miner. The most important modifications include: (a) The priority assignment strategy: a method for rewarding the creation of good quality solutions. (b) Use of modularity: a method of forming clusters of similar pathways in the solution graph. (c) Use of attack thresholds: a method for improving the processing time for the formation of more accurate rules.

Works of Abadeh et al. ([128]; [129]) and Alipour et al. [130] were among the first that combined genetic algorithms and ACO for the induction of accurate fuzzy classification rules. Fuzzy set theory [131] has been applied successfully in the past in the field of intrusion detection [132] and has proven to provide very competitive DR and FAR percentages. In this case, fuzzy if-then rules are coded as strings, with 5 linguistic values being represented by the following symbols: small (A_1), medium small (A_2), medium (A_3), medium large (A_4) and large (A_5). For instance, a rule which is coded as follows: (A_3, A_2, A_5, A_1), C_j, CF_j can be translated as: if x_1 is medium and x_2 is medium

small and x_3 is large and x_4 is small then the class is C_j with certainty $CF = CF_j$. For the most part their algorithm follows the flow of the Michigan algorithm [133], thus an initial population of fuzzy if-then rules is randomly generated. This population is then evaluated and genetic operations take place so that a new population can be produced by generating new rules. At this point, the ant colony algorithm takes a fuzzy rule and modifies it by performing a number of predefined changes so that an improved version of the same rule is produced. The algorithm then continues as normal by replacing a prespecified number of if-then rules with newly generated ones and finally stops according to some termination rules. In other words, the authors added a local search step based on ACO to the Michigan algorithm. By doing so, the entire (global) search capability of the algorithm is enhanced.

Agravat et al. [134] on the other hand, decided to modify the algorithm so that it will store all the generated high quality rules by the entire ant colony, instead of just the best one produced by each ant. In the end, all rules are first sorted with respect to their predictive accuracy, and then sorted again with respect to false positives this time.

8.1.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) seeks inspiration in the coordinated movement dynamics of swarms of animals such as the birds or the fishes. Reynolds' studies in the bird flocking behaviour [135] indicate that the transpositions of the entire flock is a result of the individual effort of birds, each revolving around 3 basic laws: (i) collision avoidance, which dictates individuals to avoid neighbour mates by readjusting their physical position, (ii) velocity matching, which dictates individuals to synchronize their speed with neighbour mates, and (iii) position centring, which dictates individuals to stay close to flockmates. Reynolds applied this model to simulate the aesthetics of the flock choreography in a three dimensional computer generated environment. Before that, the sociologist Wilson, noticed that individual members of a swarm may profit from the discoveries and previous experiences of other members during tasks such as food discovery for instance [136]. Putting it in another way, a larger number of swarm members, increases the chances of locating a rich food source and the social information sharing among the swarm members offers an additional advantage. It was not until later however, that Kennedy and Eberhart introduced the term of Particle Swarm Optimization and their work was the main influence of the basic PSO model [137]. According to this model, a fitness function exists $f : \mathbb{R}^n \to \mathbb{R}$ which measures the quality of the current solution. A number S of particles (solutions) is placed randomly inside the hyperspace in the position $x \in \mathbb{R}^n$ each having a random velocity $v_i \in \mathbb{R}^n$ The particles move in the hyperspace and at each step evaluate their position according to the fitness function.

Each particle in the swarm represents a possible solution. The basic update rule for the speed is:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1(p_i - x_i) + c_2 r_2(g - x_i)$$

Where ω is the inertia weight constant, c_1 and c_2 are the acceleration constants, r_1 and r_2 are random numbers, p_i is the personal best position of particle i, g is the global best position among all particles in the swarm, and x_i is the current position of particle i. Moreover, the update rule for the position is:

$$x_i(t+1) = x_i + v_i(t+1)$$

Two key features of this model are that (a) the speed (and therefore the next position) of each particle is calculated according to the findings of both that particle and the findings of the rest of the swarm and that (b) the global best solution is communicated among all particles of the swarm. The reader may notice the obvious similarities PSO portray to Genetic Algorithms as described in section 8.3. Indeed, they both consider a fitness function that acts as a criterion for population reproduction and update their population using randomness. However, PSO does not incorporate genetic operators such as mutation and gene crossover. Furthermore, PSO retain a kind of memory, which is essential toward the convergence to an optimal solution.

8.1.2.1 PSO & Neural Network Hybrid Approaches

PSO has been extensively used in combination with various types of ANN for improving the performance of the resulting system.

Michailidis et al. were the first who managed to successfully merge these two techniques to create an improved system for intrusion detection [138]. During the training phase the PSO is executed recursively to train the ANN with each particle in the PSO corresponding to the synaptic weights of the ANN. The optimal synaptic weights are fed to ANN, which is actually responsible for the main part of the classification. The input layer of the ANN is constructed by the m features that constitute a record in the dataset. The output layer is comprised of the normal and abnormal classes. The particle with the optimum adaptation values is searched globally.

A Wavelet Neural Network (WNN) [139] is a feedforward type of ANN. Systems of this type use a wavelet function on the hidden layer instead of the sigmoid one. The resulting systems may achieve higher learning speed and avoid the creation of local minima, therefore this type of NN has been used frequently in intrusion detection. Liu and Liu ([140], [141]) proposed the utilization of PSO instead of the typical methods

of weight adjustment (such as the Gradient Descent (GD) algorithm [142]), as it makes it possible for the solution not to get trapped in local minima. The authors used two variations of PSO, namely Quantum Particle Swarm Optimization (QPSO) [143] and Modified Quantum Particle Swarm Optimization (MQPSO) respectively, to train the WNN.

Ma et al. [144] use both the Conjugate Gradient (CG) algorithm [145] and QPSO for parameter optimization. The QPSO has a better global searching ability compared to the CG, thus it is employed in the initial steps of the training to quickly cover a larger portion of the search space. As the iterations of the algorithm proceed, and before the solution gets trapped in local minima, CG is utilized to help QPSO escape this unwanted situation. Moreover, Ma and Liu [146] adopt principles of fuzzy set theory and integrate them on a WNN based IDS. The hybrid ANN is able to "fuzzily" describe fault characteristics of a state classified as "abnormal".

Radial Basis Function Neural Networks (RBF) [147] are probabilistic Neural Network frequently adopted by classifiers in the field of intrusion detection. An RBF may achieve classification faster because that process is based on a simpler measure, that of the distance of the centres of the neurons from the inputs fed to it.

On the negative side, RBF is highly sensitive to certain parameters such as the number of center and the variance of the RBF, which are generally chosen manually. If the parameters are not optimal this will have a negative impact on the accuracy of the resulting classification. Systems such as [148] use PSO for RBF parameters optimization and achieve better performance than standard RBF. Tian and Liu [149] build upon the same theme to create a hybrid PSO-ANN system but also introduce an evolutionary mutation algorithm as an extra step in order to (a) protect PSO from trapping into local minima, (b) increase the diversity of the population, and (c) expand the scope of the search.

8.1.2.2 PSO & SVM Approaches

Another technique frequently used in combination with PSO is Support Vector Machines (SVM) [150], [151]. SVM is based on structural risk minimization of statistical learning theory and showcases good learning ability and generalization skills when applied to high dimensional or noisy datasets. These attributes are highly desirable in intrusion detection. However, one of the major shortcomings of this technique is the difficulty to determine certain parameters so that the performance of the algorithm becomes optimal. Wang et al. were among the first who combined PSO and SVM [152]. They used two different flavours of PSO the Standard Particle Swarm Optimization (SPSO) and Binary

Particle Swarm Optimization (BPSO) [153] for seeking optimal SVM parameters and for extracting a subset of the most relevant features, respectively. In a similar way, Ma et al. [154] propose a combinatorial BPSO-SVM technique where dataset features and the crucial SVM parameters are represented by each particle position. The choice of SVM parameters and feature reduction happens simultaneously in one step. The classification process, which based on SVM, is given the inputs from the previous step, thus it becomes much more accurate. Hybrid PSO-SVM systems are common in literature [155], [156].

8.1.2.3 PSO & K-Means Approaches

Xiao et al. [157] combined the simplicity of the K-Means algorithm [158] with the PSO to create a hybrid detection algorithm. According to their algorithm, the data points are first assigned to K clusters in a random manner, then the centroids are calculated and the position of each particle is deduced. For each particle, the fitness function evaluates the position, and if necessary, the P_{best} and G_{best} values are updated along with the velocity and position. Finally, the K-Means algorithm runs with the aim to optimize the new generation of particles. The advantages of this approach is that the algorithm converges to local optimum with very low probability and has high convergence speed.

8.1.2.4 PSO for Induction of Classification Rules

Guolong et al. explored the efficiency of a novel rule-based IDS based on PSO [159]. In their approach, each particle represents a rule, and recursively an entire population of particles is created based on the training dataset. For each of the particles, its fitness is calculated and the values of P_{best} and G_{best}, i.e., the velocity and the position values of that particle are updated. When some criteria are met, the G_{best} particle (the most accurate classifying rule) is inserted into the rule sets and at the same time the training data covered by this rule is deleted. The authors had to overcome the fact that PSO cannot be directly applied to network intrusion datasets because the attributes take distinct values. To overcome this limitation they also proposed a new coding scheme. Chang et al. [160] followed a similar approach to achieve better detection rates by incorporating a more accurate fitness function to the system described above.

8.1.3 Ant Colony Clustering

Many ant species exhibit an interesting behaviour concerning the organization of their nests. By taking a look on the inside of their nests, one may notice that eggs, brood and food are not randomly scattered. On the contrary, they follow a strict organization

into piles of homogeneous objects. Moreover, if for example an external force would cause turmoil to the nest, then the sum of the ants would start reconstructing these piles rapidly. This behaviour is achieved while each ant appears to work autonomously without receiving any orders by ants placed higher in the hierarchy. Based on these observations mathematical models have been constructed to simulate the clustering and sorting behaviour of real-life ants. Deneubourg et al. constructed the basic model to describe this behaviour and applied the result in robotics [161]. According to their model, ant-like robots without communication abilities, hierarchical organization or any global mapping of their environment, move randomly on a two dimensional space and pick up the objects that are located in the less dense areas. Being able to carry them they dispose them in locations where a large number of similar type of object exists. Thus, the probability of picking up or dropping objects is relevant to two factors: (a) the density of objects in the immediate neighbourhood, and (b) the similarity of objects. More specifically, the probability for an unloaded ant-like robot to pick up an object o_j is calculated as:

$$p_{pick}(o_i) = (\frac{k^+}{k^+ + f})^2$$

Where f is an estimation of the spaces in the neighbourhood that are occupied by objects of the same type, and k is a constant. If there is a small number of objects in the neighbourhood, then $f << k^+$ and p_{pickup} tends to 1, and as a result, the objects will likely be picked up. On the other hand, the probability for a loaded ant-like robot to drop the object if that robot is located on an empty cell is calculated as:

$$p_{drop}(o_i) = (\frac{f}{k^- + f})^2$$

In case where many objects are observed in the immediate neighbourhood then $f >> k^-$ and p_{drop} tends to 1, which in turn means that the object will most likely be dropped. The model assumes that each ant-like robot has a short term memory of m steps that records what is met in each of the last m time steps. Since the robot moves randomly in the search space, this sampling provides an estimation of the type of objects that exist in the immediate neighbourhood. For example, for a memory of 5 steps at time t the memory string could have been "_AA_B" indicating that the robot met 2 objects of type A and 1 object of type B. Thus $f_A = \frac{2}{5}$ and $f_B = \frac{1}{5}$. Lumer and Faieta generalized the aforementioned model for clustering multidimensional datasets [162]. The algorithm scatters the multidimensional records of the dataset in a theoretical two dimensional grid. At each iteration of the algorithm the elements are rearranged in such a way so that similar elements are grouped together to form compact clusters (ideally one for each class in the dataset). According to the LF model, the probability of picking an element i, is defined as:

$P_{pick} = (\frac{k^p}{k^p + f(i)})^2$

Where k_p is a constant and f_i is the local estimation of the density of elements in a small surrounding area defined as a square of d nodes. Likewise, the probability of dropping a carried item is calculated by:

$$P_{drop} = \begin{cases} 2f(i) & \text{if } f(i) < k_d \\ 1 & \text{otherwise} \end{cases}$$

The density dependent function $f(i)$ for an element i, at a particular grid location, is defined as:

$$f(i) = \begin{cases} \frac{1}{d^2} \sum_j \frac{1 - d(i,j)}{a} & \text{if } f(i) < k_d \\ 0 & \text{otherwise} \end{cases}$$

In the expression above, $d(i, j)$ measures the dissimilarity between all elements in the local area that surrounds node i and a scales the dissimilarities. Since the elements are vectors, d measures dissimilarities by calculating the Euclidean distance between the elements in nodes i and j. The normalizing term d^2 equals the total number of sites in the local area of interest, thus $f(i)$ may only take its maximum value if all the neighbourhood is occupied by identical elements. The algorithm described above can lead to the construction of clusters of similar objects from an initial randomly scattered state. Figure 8.2 visually depicts this process. This achievement is of paramount importance for any IDS. Based on the assumption that intrusive activity happens scarcely than legitimate one, datasets that contain low level network traffic can be analysed in order to form clusters that represent different types of attacks or normal activity respectively. The LF algorithm and subsequent variations of it were also utilized with great success in a number of other applications such as text document classification [163] to name one. Recall that this section focuses solely on the application of this family of algorithms in intrusion detection, neglecting the rest of the potential applications.

8.1.3.1 ACC Approaches

Ramos and Abraham were two of the first researchers who attempted to introduce the LF algorithm described above into the intrusion detection realm [164]. In this case, instead of having agents exploring the terrain randomly, they suggested on relying on pheromone traces to guide the agents in the grid. Moreover, the computation of average object similarities, which is dictated by the LF algorithm, is avoided since it is blind

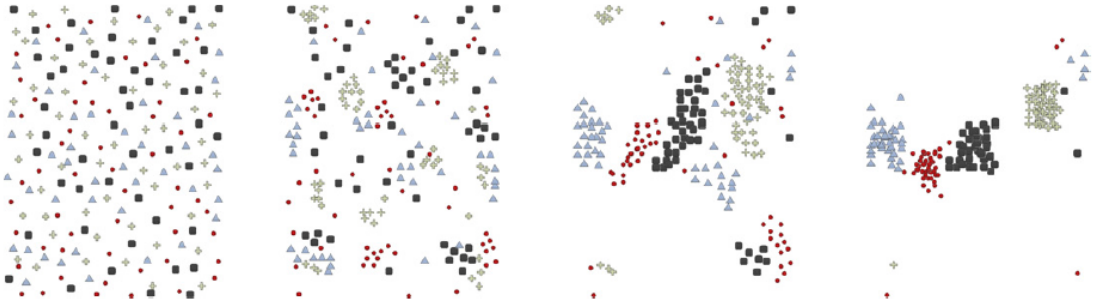


FIGURE 8.2: The Arrangement of Data into Four Classes After (a)0 (b)10,000 (c)50,000 and (d)130,000 Iterations

to the actual number of objects present in a given neighbourhood. According to the authors, this strategy (a) allows ants to find clusters of objects in an adaptive way, (b) eliminates the need of short term memory of the agents, thus making the algorithm less resource demanding, (c) it accelerates the algorithm into finding optimal solutions since the ants tend to focus on areas of higher interest.

Tsang and Kwong noticed that data used in the cases of intrusion detection analysis, typically, have large volume and high number of dimensions, [165],[166]. The original LF algorithm suffers from the fact that (a) many homogeneous clusters are formed and thus it is difficult to be merged when they are spatially separated into a large search space, (b) the density of similarity measures, favours cluster formation in locally dense regions but discriminates dissimilar objects intensively. In other words, elements of type A that are located near compact clusters of elements of type B will likely remain isolated. Under the light of these facts, the authors proposed a variation of the LF algorithm which combines measurement of local regional entropy and average similarity. Furthermore, they relied on two different types of pheromones for guiding the ant-like agents toward clusters (for object deposition) and toward isolated objects (for object pick up) respectively. Based on this classification algorithm, the authors proposed an integrated multiagent IDS architecture for industrial control systems later on [167].

8.1.3.2 ACC & SOM Approaches

Feng et al. followed a similar approach to the LF model although in their case the neighbourhood is perceived as circular area around the ant, and the pick and drop probabilities are calculated based on non-linear functions [168], [169]. After the initial clustering step, the labelling of the clusters begins. Finally, live detection is possible by calculating the a-posteriori probability with the help of the Bayes theorem. This makes the detection procedure more accurate since it is independent of cluster centres. Later

on, Feng et al. [170], fused the algorithm described above with a variation of the Self Organizing Maps (SOM) [171] neural network model.

Dynamic Self-Organizing Maps (DSOM) [172] was added as an extra step before the main ant colony clustering process. Rather than placing the input data randomly on a 2-D grid, DSOM is used to represent the input data. As an extra step the ants move the objects in the output layer of DSOM and normally form clusters. This additional step increases the efficiency of cluster formation process.

8.1.3.3 ACC & SVM Approaches

Zhang and Feng presented a hybrid framework [173] which combines SVM [151] and ant colony clustering for increasing the performance of IDS. Typical SVM techniques when used for clustering in intrusion detection, map the data as points in a multidimensional space. SVMs create hyperplanes between two classes of objects with the best one being that with the maximum distance between marginal points of the two classes. The authors incorporate, active training [174], a technique for decreasing the necessary amount of training data used to train the SVM. This is a multistep process where, for each iteration, only some of the training data are chosen and the hyperplane is modified gradually. During this step, the authors use ant colony clustering as a selection technique of the data points. The active training algorithm is extended by adding an extra step of cluster creation around marginal points, and then the selection is made from the data points of these clusters.

An intrusion detection model based on the combination of SVMs and ant colony clustering can also be found in [175]. In this approach, ACC, has more active role which is to refine the clusters initially produced with SVMs. The utilized ACC algorithm in this case is based on the fuzzy ants concept [176]. This algorithm is a variation of the original LF algorithm, but in this case, ants are allowed to pick up and drop objects to initially form heaps, each positioned on a single cell. As a second stage of this algorithm the ants pick up and drop the entire heaps formed in the first stage to construct clusters.

All the approaches discussed here are arranged in chronological order and presented in figure 8.3. IDS are organized according the SI technique adopted by the system. Arrows indicate other ML methods that possibly influenced each IDS.

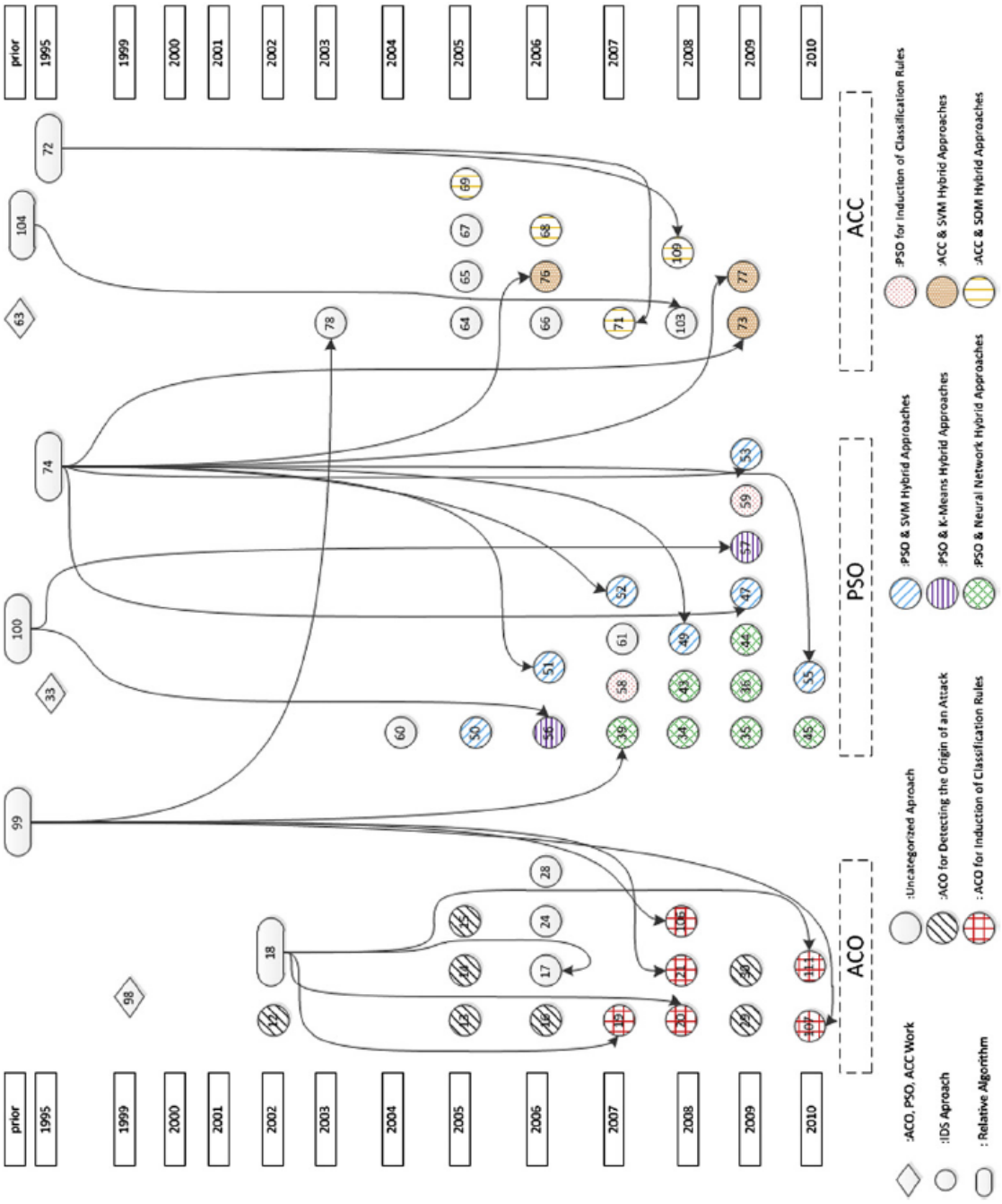


FIGURE 8.3: Major SI-based IDS Approaches in Chronological Order

8.2 Artificial Immune Systems

The Human Immune System (HIS) is a self-organizing, distributed but also lightweight system that has undergone millennia of evolution to render itself an effective defensive system against various harmful pathogens, such as bacteria, viruses, and parasites. The admirable attribute of the HIT is that it is capable of distinguishing pathogens (even ones that has not encountered before) from self tissues, and then proceed to the appropriate actions against the intruders. The features of the HIS, by definition, fulfil the design principles of an IDS, so it is without surprise, that this paradigm has gathered the attention of the research community. Eventually, the accumulated interest has lead to the generation of a new research field, namely the Artificial Immune System (AIS).

8.2.1 The Human Immune System

The HIS follows a multilayered, complex structure which includes the adaptive immune subsystem among others. The way the adaptive (also refereed to as "aquired") immune system functions, has been proved the primary source of inspiration for the creation of AIS. This system is able to adaptively alter itself in such a way that it may distinguish between a large variety of alien, non-self, elements even in the presence of native, self, ones. More importantly, it is capable of developing memory to their signature structure for faster responses in the future.

Among the large variety of the cells in the adaptive immune system, the lymphocytes and more specifically the T and B cells play a more central role. T lymphocytes are involved in cell-mediated functions, while B cells are important in the humoral operations of the immune system. The two coordinate their actions to distinguish self from non-self cells.

Specialized cells, namely the Antigen Presenting Cells (APC) engulf and fragment antigens to smaller, inactive portions, the peptides. Then, they expose these peptides to their surface, so that T cells (which have receptors) may bind with them. A binding of this kind is actually of chemical nature and when it occurs a positive match is said to have taken place. The more compatible the structure of receptors and peptides is, the larger their affinity is, a metric that dictates the strength of the bond.

Lymphocytes pass through several phases in their lifetime, perhaps the most significant of which is the negative selection stage. During this process, T cells and B cells that bind with self cells are killed, so that autoimmunity is prevented. T cells and B cells which survive the negative selection become mature, and only then are granted access to the blood stream to start patrolling for pathogens. At this point, the lymphocytes may be considered mature but still are naive, since on the one hand, they don't have

encountered antigens and on the other, they aren't exposed to certain uncommon self proteins. The chance for false detection is still high, thus these cells need to be activated first from multiple sources. Even after activation, the lymphocytes still pass through several processes of mutation until they become fully capable. One of these processes for example, is somatic hypermutation, during which, B cells mutate with great pace to produce new versions of themselves that possess differentiated receptor structures. The cells become able to match to a large variety of antigens. In this way, the immune system is able to protect against unseen or dynamically changing pathogens (like viruses). When B cells and T cells are activated a portion of that population will mutate to become memory cells. These cells play the role of a database of effective B and T lymphocytes. Upon every new encounter with a previously seen antigen, the appropriate memory cells are activated. In this way, subsequent exposures to a known antigen produce a timely response.

8.2.2 Artificial Immune System Models for Intrusion Detection

As already pointed out, the HIS is a highly complex system that involves a plethora of different functions. Judging by the works in literature we can conclude that it is only specific functions of the entire immune response have been adopted and led the researchers in creating a digital analogy. The most important of such can be organized into:

- Idiotypic Network Theory
- Negative Selection
- Clonal Selection
- Danger Theory

8.2.3 Idiotypic Network Theory

This theory assumes that antigens are primarily detected from an idiotypic network of interconnected B cells. These cells both stimulate and suppress each other in certain ways that lead to the stabilization of the network.

8.2.4 Negative Selection

Models of this type are inspired by the negative selection process that undergoes in the thymus during the maturization of the lemphocytes. The receptors of T cells are

constructed through a random rearrangement of their structure, during a process which may result in new receptors matching both antigens and self proteins. With negative selection however only the T cells that do not bind to self are permited to exit the thymus and enter the bloodstream. In essence, negative selection is the mechanism that gives the immune system the ability to detect antigens, while maintaining tolerance to the body's own proteins.

The first attempt to construct an artificial version of the HIS was proposed by [177]. The authors chose to model the negative selection mechanism and used it as the core of an IDS specialized in the detection of invalid system call sequences and suspicious file changes. The proposed system required a dataset of normal/self records given as strings. Then, a process of hypermutation, would initiate during which a new set of random strings would be generated from recombining parts of the existing ones, on a binary level. That step produced a large population strings to be added to a set of detector candidates. The detector strings matching one in the existing normal/self strings would immediately be removed from the detector pool. In the detection phase, if a new record matching one of the detector strings, would appear, an alert would be raised. Essentially, Forrest [177] algorithm consists from 3 discrete steps: (a) recognizing what is perceived as normal, (b) generating possible instances of anomalous activity, and (c) monitoring for anomalies by trying to match the instances of the previous step with the test instances.

The same system was later on extended with emulation of the possible states or conditions, a lymphocyte may go thorough in its lifetime, i.e., immature, naive, activated, memory or dead. This contributed to a faster elimination of self-matching detector strings and rendered the system more adaptive to slight changes on the structure of normal data.

Finally, Williams et al. [178] added a maturation mechanism in an effort to expand the space of the non-self antibodies, thus achieving more efficient detection rates.

8.2.5 Clonal Selection

The Clonal Selection Theory provisions that only the lemphocytes which systematically recognize antigens are reproduced and proliferate. Being close to the principles of that theory as well as the foundations of evolution and natural selection, the basic axes of such models are: (a) new detectors are clones of their parents (thus maintaining their basic characteristics) but are also subjects to mutation (thus incorporating new detection abilities), (b) the detectors that act against self lympocytes are banned, and in that way,

are not able to expand their population, (c) the detectors that are matched with antigens (detect threats) proliferate.

Capitalizing on this theory, Kim and Bentley [179] proposed a double layered IDS. The first subsystem generates detectors from a library of genes, and forwards them to the secondary, which executes the actual detection and simultaneously perpetuates the successful detectors through clonal selection.

8.2.6 Danger Theory

Danger Theory suggests that cells which die abnormally (a condition known as necrosis), or get injured, release biological alarm signals. A type of cells known as Dendritic Cells (DC) forward the alarm signal to the adaptive immune system for specialised response.

The DCA algorithm, proposed by [180], places the emulated DCs at the core of their architecture. In their model, input data correspond to the antigens and different kinds of signals (pathogen associated molecular, patterns, safe signals, danger signals and inflammatory cytokines), describe different danger states of the monitored system. The DCs are able to coordinate the functionality of the immune system by correlating these signals to the antigens.

Twycross et al. [181] proposed an alternative approach to the negative selection algorithm, that is based on danger theory. Once more, entities described as DCs have a vital role in this model. During the training phase DCs are fed with normal data and with randomly generated T-cells. In the case of successful match, the corresponding T cells are removed. During the testing phase, an alarm is raised when T cells are activated by antigens.

Dasgupta et al. [182] simulated the functions of T and B cells in further extend. In their model, T cells were specialised to perform low-level continuous bitwise match, in an analogy with the real T cells which are capable of recognizing peptides extracted from real proteins. On the other hand, the B cells performed a high-level match, similarly to the real B cells which recognize epitopes on the surface of antigens. This model further simulated suppression of false alarms and negative selection, maturing/activating of the cells, clonal selection and somatic hypermutation of mature T cells and B cells.

8.3 Evolutionary Computation

Evolution is the natural process of alternation in the inherited characteristics of organizations over the course of several generations, so that these organisms become better

adapted to their environments. Charles Darwin [183] formulated the theory of evolution through the process of natural selection. Evolution by natural selection makes three assumptions about its population:

- More offspring are produced than can possibly survive.
- The characteristics of individuals vary in a population, leading to different rates of survival and reproduction.
- Differences among individuals are inheritable.

The least fit members are statistically the ones that become extinct the first, thus gradually leaving a population dominated by the members which are better adapted to their environment and posses those characteristics that can ensure their survival. By reproduction these characteristics are inherited and possibly combined, leading to descendant with even more of these desirable characteristics. Natural selection, perpetuates and increases desirable features over a population and is the sole driving force of adaptation. However, it is not the only cause of evolution with two others being the random mutation and genetic drift.

Approaches that fall into the family of Evolutionary Computation (EC) are Evolutionary Programming (EP), Genetic Algorithms (GA), Genetic Programming (GP). While some differences, regarding implementation details may exist, conceptually they are nearly identical. Such techniques rely on procedures that simulate the natural processes of reproduction, mutation, recombination, natural selection and survival of the fittest. Typically, candidate solutions, play the role of individuals in a population, and a fitness function dictates whether these individuals will act as a basis for the creation of better solutions.

The partial solutions are usually represented as tree structures and with each generation applying mainly two actions on them:

- *Crossover* - is applied on individuals by switching one of its nodes with another node from another individual in the population.
- *Mutation* - is applied on individuals by altering a node.

Except for these operators, many other variations have been proposed for improving detection rate specifically in the intrusion detection realm. Among them, seeding and deletion are two emerging operators that have been applied with success.

- *Seeding* [184] is a method for dynamically generating new individuals fast, and usually used to initialize the first population.
- *Deletion* [184] helps in maintaining the size of a population constant by removing the correct individuals after the generation of new ones. In traditional EC, the less fit individuals are replaced blindly. However, this approach may not be appropriate for multimodal, unequally distributed datasets such as the ones met in intrusion detection problems. In such datasets, rules that cover normal class will have a higher fitness than others, thus rules for the normal class have much lower chances to be deleted compared to ones for other classes. By integrating class distribution into the deletion operator contributes to the better handling of minority classes.

8.3.1 GA & Neural Networks

Evolutionary computation has been applied to ANN for automatic design of the network. Typically, deciding an optimal network structure is one of the bottlenecks of ANN. The experimental results usually indicate that the hybrid solutions outperform the conventional approaches. In schemes such as [185], GA were employed for discovering an optimal feature set and to learn the RBF network parameters such as the basis function, the hidden neurons, and the training epochs.

The IDS presented in [186], [187] uses ANN and Fuzzy logic to detect threats, but utilizes GA to generate all nodes for the network.

8.3.2 GA & Clustering

The authors in [188] capitalise on the Unsupervised Niche Clustering (UNC) a clustering algorithm that combines GA with a niching strategy. More specifically, the GA locates the clusters using a robust density fitness function, while the niching technique creates and maintains the candidate clusters. UNC is less prone to finding suboptimal solutions than traditional techniques.

Similarly, [189] used the K-means clusterer to locate possible cluster centres, and in a second step, GA is used to optimize the result by refining the cluster centres.

The authors in [190] employed GA to find the value of the most important parameter of K-means algorithm which is the number of clusters.

8.4 Conclusions

According to our study, the majority of the IDS that make use of ACO, utilize this mechanism for post intrusion procedures. More specifically, most approaches employ ACO for tracing the source of an intrusion and in some cases even responding to that intrusion at its source [191], [192], [193], [194], [195]. However, these systems do not fall within the scope of this work because they use the ACO paradigm for response rather than detection.

Very few proposals recognized the potential of ACO in the improvement of the classification process. Most of these apply ACO for classification rules extraction and all of them rely on a modification of the Ant-Miner algorithm, which uses the pheromone concept for extracting good quality classification rules. Generally, the Ant-Miner algorithm was originally tested on datasets with distinct record values and it is not optimized for datasets with continuous attributes such as the ones used for intrusion detection.

PSO in intrusion detection is rarely used as the exclusive method for classification. The majority of the relative research treats this technique as a supplementary step to some other machine learning classifier which in turn conducts the main part of the detection. One basic point to be taken into account is that the use of PSO has significantly boosted the performance of all the machine learning techniques in which it was applied. It is safe to say that the use of PSO into an existing machine learning based IDS is expected to enhance the system's DR accuracy by a factor of at least 3%. Unfortunately, with the incorporation of multiple techniques the computation requirements are expected to increase.

Interestingly, most ACC methods, rely solely on an ACC algorithm for the classification process, and to the best of our understanding the true potential of ACC methods is yet unexplored. From the presented works one can notice ACC based detectors achieve some of the best results for the R2L class among all machine learning approaches. Additionally, it performs extremely well for Normal class (99.1% on average).

A great amount of work uses GA in combination with other approaches (more frequently ANN) as an auxiliary method for calculating sensitive parameters.

Most researchers have confirmed the positive role, mutation has played in the process for searching the optimal. However, different opinions about crossover in multimodal problems where the population contains niches.

Conclusively, the advantages nature inspired approaches possess over traditional machine learning and other soft computing techniques render them good candidates for IDS. The strong points of nature inspired approaches can be summarised as:

- They can achieve high degree of parallelism with little effort and changes to their basic concepts.
- They have the potential of achieving better detection rates as they are able to avoid falling into local minima/maxima.
- They can adapt fast, to changes in conditions regarding both the normal and anomalous data instances.

Despite the considerable adoption of nature inspired computation tactics in the intrusion detection problem, some open issues remain. Evolutionary approaches favour the frequently occurring instances. In particular, individuals that represent the frequently occurring data instances are more likely to survive during the evolution steps, even if they are less fit than others [184]. Unfortunately, in intrusion detection datasets, the data distribution is expected to be highly unbalanced, with the attack data being the minority, and with some attack classes being an order of magnitude smaller than the rest.

The efficiency of the majority of the evolutionary algorithms depends upon the manual setting of numerous parameters including the termination after a maximum number of generations. Such parameters can only be decided empirically, something which renders the entire process error prone. The absence of an automatic stopping criterion, in particular, may undermine the process of discovery of the global optimum. An even worse situation occurs with the ACC algorithms which have even more parameters requiring custom fine-tuning.

Although the appealing metaphor of AIS has ignited a significant amount of work by the research community, there still seem to be hurdles preventing such approaches from being applied to real world problems. The most important one is its efficiency on the high dimensional data. In further detail, the majority of NS approaches rely on the greedy, random generation of non-self patterns, to be used as signatures for matching intrusions. Obviously, this scales bad especially, in high volume, high dimensional data such as the ones usually handled in intrusion detection problems.

Finally, it has been found that PSO besides its simplicity and speed it has a tendency to converge to a suboptimal solution on the early stages of its execution and naturally it performs poorly when used as the core detection method on an IDS. Further research is required to alleviate this behaviour.

Chapter 9

AWID: A Dataset for Wireless Intrusion Detection

In the field of network intrusion detection, the development of anomaly based IDS is constantly in the scope of researchers due to its promising characteristics, with the recognition of unknown threats being viewed as the "golden goose". However, the application of such systems as part of a real-life protection mechanisms is hindered, as the majority of such systems struggle with inaccurate predictions percentages and unsatisfactory performance.

The existence of trustworthy datasets that will not only act as a type of virtual instructor for the IDS, but also a reliable benchmark is considered of paramount importance, especially during the early stages of IDS development. Yet, in practice the available choices are not only limited but also of poor quality. Due to the sensitivity of their content many datasets created for industrial needs are never made available to the public, instead, they remain as a corporate secret. As for the publicly available ones, in some cases, their contents are heavily tampered by their creators (e.g., in order to become anonymized), while in other cases, they are outdated or may even contain data that do not correspond to realistic conditions.

In this chapter, we briefly describe the most common datasets used in the field of intrusion detection. We attempt to highlight the inefficiencies of the most popular one, namely the KDD'99, as well as the reasons why this particular dataset falls short when being used as a benchmark for wireless intrusion detection. In the process, we introduce our custom-tailored dataset, specifically targeting the wireless realm, namely the Aegean Wireless Intrusion Dataset (AWID) and provide details about several of its aspects. To the best of our knowledge this is the first attempt to release a security relevant, dataset specifically targeting any kind of wireless technology.

9.1 The Importance of Datasets in Anomaly Intrusion Detection

9.1.1 What is a Dataset

A dataset is typically a collection of information instances, frequently refereed to as records, vectors, observations, samples, events, or entities. Each of these instances contains a set of attributes which can be of different types such as nominal, numerical, textual or even binary. Attributes may be of the same type or more frequently of different types. Typically, each one of the instances is associated with a label to denote its class.

In intrusion detection the label indicates whether that record is a normal or an anomalous one and in most cases the attributes of that record correspond to a certain aspect of a network connection, a field of a packet header, a column of a log file generated by the OS or some part of a user executed sequence of commands.

9.1.2 Datasets in Supervised Anomaly Detection

As explained in 7 anomaly based intrusion detection approaches may fall into either one of the two large families of intrusion detection techniques, namely the supervised and the unsupervised one. Detectors of the first category rely on the existence of pre-labelled data to build their predictive models for normal and anomalous classes during an early training phase. Newly fed data instances are evaluated against this pre-constructed model, in deployment phase, so that they are categorized in normal or one of the intrusive classes.

It is straightforward, that supervised learning algorithms not only require the existence of such a dataset but more importantly their efficiency depends on the quality of that dataset. In other words, algorithms of this kind, will perform poorly if trained wrongly. In ideal conditions, the terms described in the dataset during the training step should be as close as possible to the ones faced in the deployment environment. However, with the intrusive traffic being unpredictable, a subject to fast evolution and altering, one can hope that trained cases are both abstract and variate.

9.1.3 Datasets in Unsupervised Anomaly Detection

Unsupervised approaches do not require a training phase so a dataset may seem as a redundant asset. Yet, even this kind of systems is imperative to be evaluated against

some sort of pre-labelled set of data to estimate their performance (both prediction and time wise) before their deployment.

9.1.4 Evaluation Metrics Used on Datasets

Usually, a dataset is used as a benchmark for estimating the effectiveness of an IDS according to its ability to provide accurate classification results. The four possible outcomes of the detection process are:

- True negatives signify the correct classifications of the IDS regarding instances of the normal class
- True positives refer to the amount of records successfully recognized as attacks
- False positives refer to false alarm events, i.e., normal records which are wrongly perceived as attacks
- False negatives refers to the amount of intrusive events that went undetected by the IDS

Based on the results described above, 6 more complex metrics can describe the performance of an IDSs from different angles. The definition of each of most commonly used evaluation metrics is as follows:

- True Negative Rate (TNR) - $\frac{TN}{TN+FP}$
- True Positive Rate (TPR) or Detection Rate (DR) - $\frac{TP}{TP+FN}$
- False Positive Rate (FPR) or False Alarm Rate (FAR) - $\frac{FP}{TN+FP}$
- False Negative Rate (FNR)- $\frac{FN}{TP+FN}$
- Accuracy - $\frac{TN+TP}{TN+TP+FN+FP}$
- Precision - $\frac{TP}{TP+FP}$

Another important performance metric is the Receiver Operating Characteristic (ROC) which can be defined as $\frac{DR_{AVG}}{FAR}$. It is usually used to compare performance of of the same systems when different parameters are applied to it or the performance of different systems.

9.2 Datasets for Intrusion Detection

9.2.1 DARPA 2000

A DARPA evaluation project [196] contributed a scenario specific dataset in the year 2000. This dataset consists of 3 subsets that correspond to specific attack scenarios, namely, the LLDoS 1.0, the LLDoS 2.0, and the Windows NT attack scenario. More specifically:

- LLDoS 1.0 includes a distributed denial of service attack unleashed by hypothetical a novice user. The scenario is executed over multiple sessions, grouped into 5 attack stages, over the course of which the attacker (a) probes the victim network, (b) gains access to a host by exploiting a specific vulnerability, (c) installs a trojan (mstream DDoS), and finally (d) launches the DDoS attack against an off site server from the compromised host.
- The LLDoS 2.0 scenario is similar to the LLDoS 1, yet more challenging from intrusion detection perspective, since the execution methodology of the attack is somehow more stealthy.
- Windows NT Attack scenario contains traces of data from that run of one day's mixed normal and intrusive traffic on an NT machine.

9.2.2 CAIDA DDoS Attack 2007

The CAIDA dataset [197] contains approximately one hour of traffic traces from a real DDoS attack that occurred in 2007. The entire set of records is given as a set of multiple files, each 5-minute in duration. All files are in pcap format, which can be read and analysed by tools such as Wireshark. The total size of the dataset is 21 GB (or 5.3 GB compressed). Non-attack traffic has been removed as much as possible leaving just the attack traffic, as well as the corresponding responses to the attack. The traces in this dataset are anonymized and the payload has been removed from all packets.

9.2.3 UNIBS-2009

The UNIBS-2009 [198] dataset contains traces which were collected from the network of the University of Brescia over three consecutive business days. The traffic was captured from the faculty's router with the use of the Tcpdump tool. The resulting raw data spans in 27GB disk size approximately, and it is constituted of primarily TCP and

UDP traffic. More specifically the traffic flows include Web (HTTP and HTTPS), Mail (POP3, IMAP4, SMTP and their SSL variants), Skype, traffic generated by Peer-to-Peer applications, such as BitTorrent and Edonkey, as well as other protocols (FTP, SSH, and MSN). The records were anonymized and stripped from their payload, in a second step for preserving the privacy of the users. The resulting traces come with a log file which indicates for each flow (a) the transport port numbers, (b) the outcome of the DPI analysis (by considering 200B of data for each packet, and signature patterns as provided by l7 filter), the application name that generated the flow, (as returned by GT).

9.2.4 CCTF-DefCon10

The CCTF-DefCon10 [199] dataset was prepared by the Shmoo Group. It was gathered during a capture-the-flag style penetration testing competition during which teams of hackers assume the the roles of the attacker and the defender interchangeably. The dataset has been stripped off its normal traffic traces leaving only the intrusive ones.

9.2.5 ISCX Datasets

The ISCX dataset [200] is built around the concept of intrusions specific profiles. A profile is fundamentally an abstract representation of certain features and events on the network. Its purpose is to facilitate the simulation and reproduction of certain behaviour as monitored through the operation of a real-life network. The profiles are subsequently used by agents to generate analogous events on the network which include anomalous as well as normal HTTP, SMTP, SSH, IMAP, POP3 and FTP traffic.

9.2.6 Android Genome Project Dataset

This dataset is a product of the Android Genome Project [201, 202] and contains of more than 1,200 malware applications, collected over a more than a year period of time (2010-2011). The contained samples cover the majority of the existing (at that time) Android malware classes. The creators of the dataset also characterized each sample with respect to their installation method, activation mechanism, the type of payload as well other aspects.

9.2.7 The Case of KDD99

The KDD'99 [203] dataset is considered as the golden standard in intrusion detection. Partially due to its transparent nature and in part due to the fact that it has been openly offered, it has become the most widely used benchmark in the field of network intrusion detection. The KDD'99 dataset was initially compiled for the needs of The Third International Knowledge Discovery and Data Mining Tools Competition, which was held along with the The Fifth International Conference on Knowledge Discovery and Data Mining. The objective of the competition was to develop a network intrusion classifier, capable of discriminating between intrusive and normal connections.

The dataset includes a variety of traces taken from a simulated a military network environment consisting of 3 victim machines running different operating systems. Another 3 machines were used inject background normal traffic that approximates the traffic profile of a real military network. A final node that acted as a monitor was placed next to the network's router and stored all traffic using the TCP dump format. The total simulated duration of the experiment was seven weeks.

9.2.7.1 Characteristics

Each record in KDD'99 corresponds to a connection between two hosts in the network and is described by 41 attributes (38 of which take continuous or discrete numerical values and 3 are categorical attributes). The dataset is comprised of two subsets, namely the training and testing one. The training set contains a total of 22 attack types and an additional number of 15 attack are solely contained in the test subset and consists of approximately 5,000,000 records. Although the records are labelled per attack type, each one of these can be further organised into of 4 broader attack classes, namely, denial-of-service, remote-to-local, user-to-root, and probe. In more detail:

- Denial of Service (DoS): In this class of attacks, the aggressor attempts to suppress the normal operation of system thus preventing valid clients from using the service. Examples of attacks included in the KDD'99 that fall into this category are the Smurf and SYN Flooding attacks.
- Remote to Local (R2L): In this case, the attackers attempt to gain access to a remote host without having a valid user account. An example of such attack is password cracking by brute force.
- User to Root (U2R): This class includes attacks where a misbehaving user has initially valid access to a low-privileged account but attempts to gain access to

a higher-privileged account frequently the superuser. Buffer overflow attacks fall into this category.

- Probe: This class contains attacks that first expose then collect information about a specific client. An example of probe attacks is port scanning.

The number of records of each of these classes in full KDD'99 and the reduced versions of the datasets respectfully, is given in table 9.1.

TABLE 9.1: Number of Records Per Class in Various Types of the KDD'99 Dataset

Dataset	DoS	Probe	U2R	R2L	Normal	Total
Full KDD'99	3883370	41102	52	1126	972780	4898430
10% Reduced	391458	4107	52	1126	97277	494020

TABLE 9.2: Number of Records per Attack and the Corresponding Classes in the 10% Reduced KDD'99 Dataset

Attack	Samples	Category
smurf	280790	dos
neptune	107201	dos
back	2203	dos
teardrop	979	dos
pod	264	dos
land	21	dos
satan	1589	probe
ipsweep	1247	probe
portsweep	1040	probe
nmap	231	probe
warezclient	1020	r2l
guess_passwd	53	r2l
warezmaster	20	r2l
imap	12	r2l
ftp_write	8	r2l
multihop	7	r2l
phf	4	r2l
spy	2	r2l
buffer_overflow	30	u2r
rootkit	10	u2r
loadmodule	9	u2r
perl	3	u2r
normal	97277	normal

9.2.7.2 Critique

Due to its wide adoption by the research community the KDD'99 has been analysed extensively [204], [205], [206] but has been the center of critique [207], [208].

It is noticed that the vast majority of machine learning algorithms trained with the KDD'99 performed poorly against the test version of the dataset, especially for the U2R and R2L classes. Actually, even after 15 years from the dataset's release, there is a surprisingly low number of approaches in literature, exhibiting satisfactory performance levels.

The organization of attacks included in KDD'99 follows an effect-centric approach which is primarily adopted by the corresponding taxonomies found in literature. While such approaches are good for academic purposes they are proven to be inefficient when it comes to intrusion detection. Hence, 2 attacks that fall into the same class may have so drastically diverse execution methodologies, that it might be impossible to identify the similarities in their traffic patterns. For example, the DoS class, includes attacks against the protocol stack, against applications as well as the system process table, all of which have practically nothing in common in matters of execution besides their effect. It is apparent that this taxonomy does not serve the intrusion detection process in the best way possible.

Another point that has raised objections is the unrealistic distribution of normal versus intrusive records in the test set, with the later covering over 80% of that particular scheme. However, unrealistic distributions expand even within the attack classes themselves, with the DoS class being the dominant, even though, experience has shown that in real life Probe attacks are by far the most common attack in computer networks.

What is more troubling is that, in their majority, U2R and R2L classes contained in the test subset, are comprised primarily by attacks not seen before in the training version of the dataset. This in combination with the fact that these two attack classes are contained is smaller percentages in the dataset gives a sound justification of their poor prediction levels.

Another discussed deficiency is the large number of noise (or repeated) records it contains. Theoretically, this forces the learning algorithms to be biased towards the more frequent records, thus preventing them from efficiently modelling the infrequent ones.

As a final word of notice we should add, that as the nature of attacks and even the normal traffic changes dynamically over time so should the dataset. The systems that are responsible for generating the background traffic in KDD'99 are by today's standards obsolete and by all means the contained attacks are no longer effective.

9.3 The Need for a Contemporary Wireless Intrusion Detection Testbed

The inefficiencies of the existing datasets for intrusion detection led us to formulate 5 major principles that a dataset in the field of intrusion detection should follow. In this way the dataset may not only pose as a trustworthy benchmark but also manage not to impede the intrusion detection process.

- Area Specific - Different domains may have drastically diverse behavioural profiles. Distinct evaluational territories, impose new challenges to classifiers. Naturally, the effectiveness of these systems is heavily influenced by the particularities of the domain it is applied to. It is straightforward that a dataset must clearly state the sector it refers to.
- Realistic Traffic - Both normal and anomalous traffic should be represented as realistically as possible. The dataset should effectively describe the normal conditions on a studied system, including the traces of the attacks, the effect these attacks have on the system, as well as the valid reactions of the system's entities to such situations. In this perspective, artificial traces are expected to have a negative effect on the consistency of the data and therefore the overall evaluation accuracy in real life conditions.
- Labelled - A labelled set of records is a mandatory asset during the evaluation of the detection systems. However, the process of labelling records is a tedious one and frequently error prone if done manually. Coupling records to one of the available classes should be done in an automated fashion.
- Integrity - Each record should be complete. The deletion of attributes from records may result in unexpected implications leading to lower detection rates, particularly for unknown attacks.
- Satisfactory Size - The amount of information available influences the accuracy of the detection mechanisms. It is also not uncommon for naive dataset reductions to happen, introducing that way an additional factor of abnormality.
- Variation - As attacks methods continuously evolve and intruders purposely alter their methodologies or even invent new penetrations schemes a large number of different attack classes, attack types and variants exists for virtually all environments. A dataset for intrusion detection should always strive to be as complete as possible.

- Freshness - This complements the aspect of variation. A dataset should be the subject of continuous update.

9.4 Introducing AWID

9.4.1 Setup & Method of Data Collection

The goal was to create a dataset that would contain realistic normal and attack traffic. To that end we choose not emulate traffic using the respective tools, nor reconstruct an artificial network but to rely on an existing SOHO infrastructure. This network already included a variety of mobile and stationary STAs, ranging from smartphones and tablets to laptops and desktops PCs. Following pre-compiled scenarios we introduced, at discrete times and for pre-specified durations, a single attacking node that was unleashing a set of attacks.

In further detail, the valid network, with ESSID "AegeanSecLab", consisted of 1 desktop machine, 2 laptops, 2 smartphones, 1 tablet and 1 smart TV. The position of the desktop machine and smart TV remained static throughout the course of all the experiments, for all scenarios. The smartphone devices displayed high mobility, i.e., they frequently changed position inside the facilities of the lab and joined/left the network numerous times throughout the course of the experiments. Finally, the laptop machines were semi-static, i.e., they rarely changed their position. Figure 9.1 illustrates the blueprints of the lab and the relative positions of the nodes inside the its facilities throughout the course of the dataset collection.

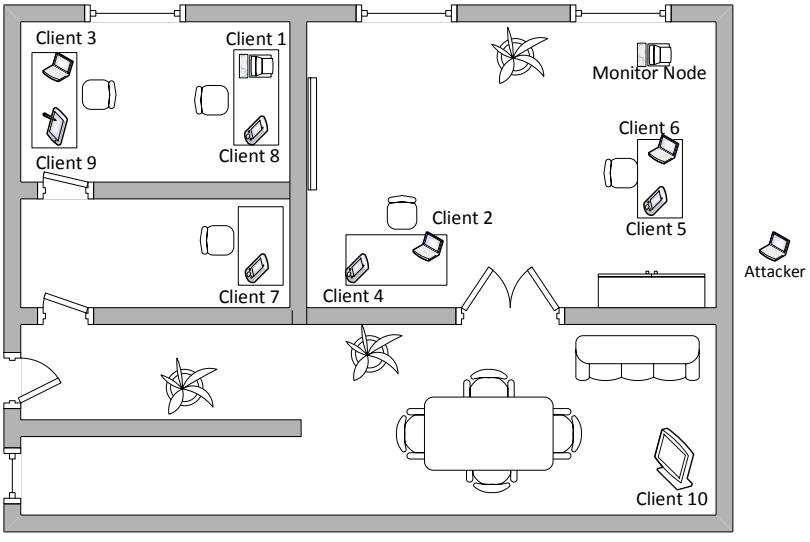


FIGURE 9.1: Lab Blueprints

The network was covered by a single AP, which was a Netgear N150 WNR1000 v3 device (Firmware Version V1.0.2.54_60.0.82), protected by WEP encryption (which nowadays is considered obsolete), supporting up to 54Mbps transfer rates. The services running on the client that were responsible for the normal traffic generation were: web browsing, e-mail messaging, music and video streaming, video conferencing, and file downloading.

A single node was responsible for introducing attack traffic to the network. The attacking node, had an Acer Aspire 5750G laptop, running Kali Linux 1.0.6 64 bit, at her disposal. The laptop was equipped with an D-Link DWA-125 card set in promiscuous mode for injecting packets, as well as a Linksys WUSB54GC card for connecting to the network. Depending on the attack scenario, the aggressor had to change her MAC address. For the attack execution various tools were used, including the Aircrack-ng suite, the MDK3 tool, the Metasploit framework, as well as custom made ones developed in C language (with the aid of the Lorcon2 library). The intruder was acting outside the perimeter of the facilities of the lab, changing her position in a random fashion.

A separate device played the role of the network monitor. This node was placed within the network premises, but was never associated with it or any other network in the vicinity (and therefore it was not probing). The monitor node was a desktop machine, running Linux Debian 7.3, equipped with a Samsung 840 series SSD hard drive capable of writing 130 MB/s and an Alpha AWUS036H card, set in promiscuous mode. The Tshark application (which is the terminal version of the Wireshark) was installed on that node and was used for dumping the captured traffic into several pcap files of smaller size. Each one of those files contained all (not limited to the network of interest) traffic captured during 1 hour. Hence, each capture file has different, unpredictable size.

Note, that the introduction of a wireless monitor node is not 100% reliable since packet drops and losses are expected to occur naturally. Yet, we highlight the simplicity and cost efficiency of this method and argue that it is the optimal solution for data capturing especially in resource constrained environments such as the SOHO ones.

After the completion of the monitoring phase an intermediate step took place, during which the generated binary files were transformed into text format (CSV files) and later on the produced ones were subjected to a process of normalization (e.g., to transcode the hexadecimal values into decimal ones).

Table 9.3 contains a detailed description of the equipment used throughout the course of the scenarios.

9.4.2 Types

The AWID dataset is constituted of 2 equal basic sets spread throughout 8 files. The two sets namely, AWID-CLS, AWID-ATK defer merely on the labelling method. The first one is labelled according a method-centric classification (4 classes), while the latter follows a more detailed attack-centric classification (16 classes).

Each of the two sets is comprised of two extended version subsets (namely, AWID-ATK-F and AWID-CLS-F) and two compact ones (namely, AWID-ATK-R and AWID-CLS-R). The compact versions are not derivatives of the full ones but were produced with the same capturing method, over two different, briefer, capturing sessions. The compact subsets are better suited for the early stages of experimentation, due to their smaller size which is permissive for analysis by a single machine. On the other hand, the extended ones have significantly larger size and possibly require big data analysis techniques.

Finally, each of the described subsets has two versions; a training (AWID-ATK-F-Trn, AWID-CLS-F-Trn, AWID-ATK-R-Trn, AWID-CLS-R-Trn) and a testing one (AWID-ATK-F-Tst, AWID-CLS-F-Tst, AWID-ATK-R-Tst, AWID-CLS-R-Tst). The training versions may be used for building models of "normal" and "abnormal" traffic during the learning phase while the testing ones serve for evaluating the trustworthiness of the constructed models.

The brief subsets (AWID-ATK-R-Trn and AWID-CLS-R-Trn) contain 1,795,575 records. Out of that volume 1,633,190 records refer to normal traffic and the rest are records classified as intrusive (162,385 records). These were generated by monitoring the test network for 1 hour, with the attack free traffic spanning 35 minutes and the traffic that contains attacks lasting for 25 minutes. The respective raw (pcap) file occupies 948 MB on the disk while the corresponding extracted dataset CSV file of 935 MB (or merely 68 MB if compressed with gzip). Likewise, the AWID-ATK-F-Trn and AWID-CLS-F-Trn contain 37,817,835 records, 1,085,372 of which correspond to some kind of attack. The records are spread over 96 files each of 1 hour of network monitoring. The accumulated size of the dataset in text format is approximately 15 GB and it was produced from over 16.3 GB of raw data.

Table 9.4 summarises the main characteristics and file structure of all the AWID sets.

9.4.3 Labelling

The AWID-CLS and AWID-ATK are virtually the same datasets with the sole differentiation on the way these records are labelled. While the AWID-ATK sets have the records

labelled as either normal or one of the 15 total kinds of attacks, in case of AWID-CLS set an execution-centric organization of attacks into classes is adopted.

In further detail the AWID-ATK may have records labelled as Normal, Fragmentation, ARP Injection, Deauthentication, Amok, Authorisation Request, Beacon, Probe Response, Evil Twin, Cafe Latte, ChopChop, CTS, RTS, Disassociation, Power Saving, Probe Request or finally Hirte. On the contrary, the AWID-CLS labels its records as Normal, Flooding, Injection and Impersonation classes.

Table 9.5 presents the correspondence of each attack type to an attack class.

TABLE 9.5: Correspondence of Categories and Attacks Contained in the AWID-CLS and AWID-ATK Versions of Sets

Attack	Category
Normal	Normal
Fragmentation	Injection
ARP Injection	Injection
ChopChop	Injection
Deauthentication	Flooding
Amok	Flooding
Authorisation Request	Flooding
Beacon	Flooding
Probe Response	Flooding
CTS	Flooding
RTS	Flooding
Disassociation	Flooding
Power Saving	Flooding
Probe Request	Flooding
Evil Twin	Impersonation
Cafe Latte	Impersonation
Hirte	Impersonation

9.4.4 Composition

Both AWID-CLS-R-Trn and AWID-ATK-R-Trn subsets were generated by 1 hour utilization of the test network. In our experiments the attack-free traffic lasted 35 minutes (60% of the time) while the rest 25 minutes (40% of the time) were dedicated to exploiting vulnerabilities of the test network. During the attack free traffic the users of the network were conducting ordinary activities such as file transfers, web browsing and video streaming but for some periods of time the network was dormant. During the

attack window a single node unleashed a set of 15 unique attacks and multiple variations of them in a non random way. The attacks were sequentially executed with the necessary order to achieve a specific task e.g., cracking the network key. In more detail, in the AWID-CLS-R-Trn and AWID-ATK-R-Trn subsets 54.5% of the attacking period was dedicated to injection attacks, 18.5% to flooding attacks and 26.8% to impersonation attacks. The AWID-CLS-R-Trn and AWID-ATK-R-Trn subsets include: Fragmentation and ARP Injection attacks, Deauthentication, Authorisation Request, Beacon, Probe Response flooding attacks as well as Evil Twin and Cafe Latte impersonation attacks. On the other hand, the AWID-CLS-R-Tst and AWID-ATK-R-Tst contains additional ones not met in the traiing sets. More specifically the ChopChop, CTS, RTS, Disassociation, Power Saving, Probe Request and Hirte attacks.

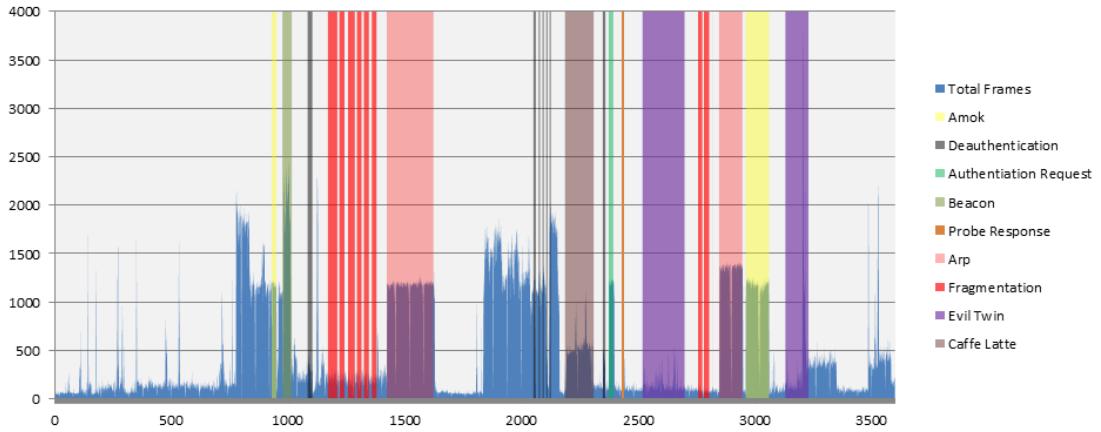
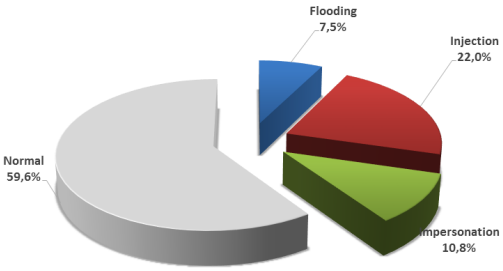


FIGURE 9.2: Sequence of Attacks in the Compact Training Set

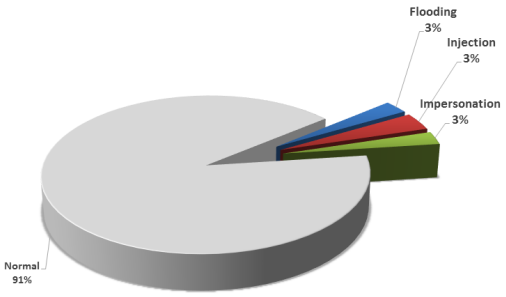
Note that the percentages above refer to time durations and do not correspond to actual number of records. For instance while flooding attacks occupied 18.5% of the attack time (or 7.5% of the total experiment time) in the AWID-CLS-R-Trn set, but during this time 48,484 malicious packets were introduced which may be as high as 29.8% of the entire attack traffic but is also just 2.7% of the sum of traffic in that set. The type of attacks included in the training and test versions of the AWID dataset along with the corresponding normal to attack traffic ratio with respect to time as well as traffic, are illustrated in figure 9.3 and 9.4. Moreover, the complete sequence of intrusive events, and their duration through time is illustrated in figure 9.2.

9.4.5 Record Scheme

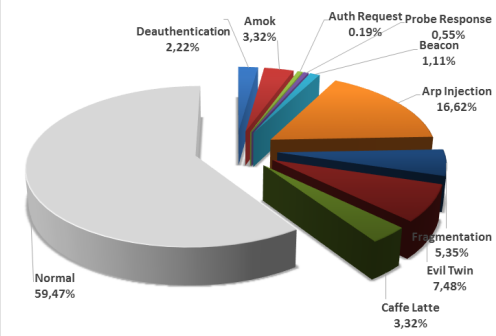
Packets in AWID are described as vectors of 156 attributes, with the last one always representing the attack category (for AWID-CLS-R-Trn, AWID-CLS-R-Tst) or the attack type (for AWID-ATK-R-Trn and AWID-ATK-R-Tst). The set of attributes is static



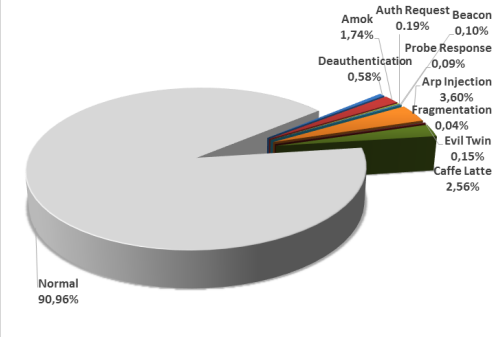
(A) Time Based Composition of the Training Subset by Class



(B) Traffic Based Composition of the Training Subset by Class



(C) Time Based Composition of the Training Subset by Attack



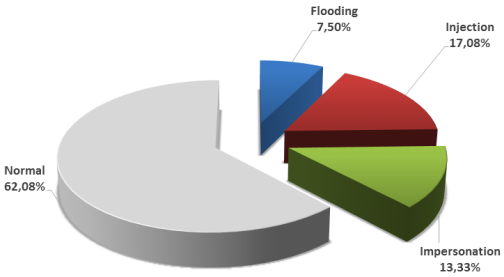
(D) Traffic Based Composition of the Training Subset by Attack

FIGURE 9.3: Attack vs. Normal Traffic in the Reduced Training sets

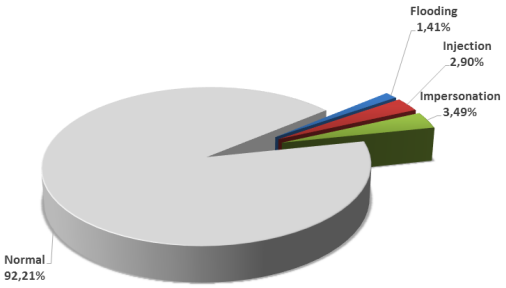
which means that a packet is described by the same number of attributes independently of its type and subtype. For this reason, a verbose representation of a 802.11 frame bearing (almost) all possible 802.11 fields was constructed. The values of -1 or “” were assigned to the fields that do not apply to a specific header type. Note that the actual data field was considered irrelevant, as its encrypted nature would not lead to the extraction of valuable conclusion and by all means it would effect negatively the detection process. Therefore, it and was not included in the dataset. Moreover, extremely rare fields (such as vendor depended ones) or custom tags were also filtered out beforehand.

Each attribute in the records was adopted either by (a) the MAC layer header, e.g., the Source Address (wlan_sa), Initialization Vector (wlan_wep_iv), the ESSID (wlan_mgt_ssid), (b) the Radiotap header e.g., Signal Strength (radiotap_dbm_antsignal), or finally, (c) from general frame information such as Packet Number (frame_number).

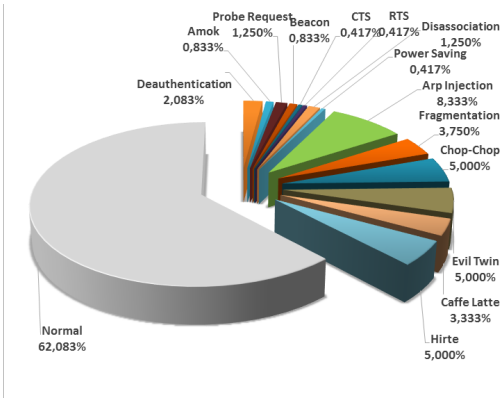
All attributes in the dataset have numeric or nominal values except for the SSID value which takes string values. Hexadecimal values or fields that represent MAC addresses were transformed to their corresponding integer values on a preprocessing step. For example, a typical MAC address corresponds to an integer value of 82468889197.



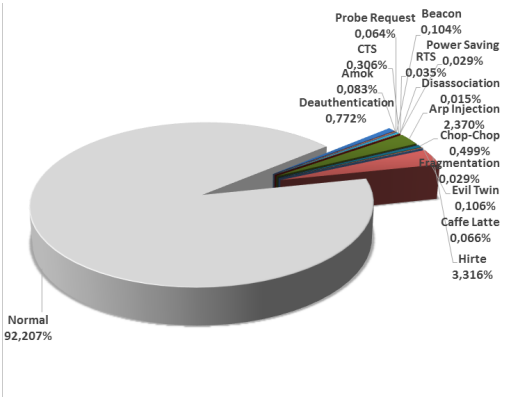
(A) Time Based Composition of the Test Subset by Class



(B) Traffic Based Composition of the Test Subset by Class



(C) Time Based Composition of the Test Subset by Attack



(D) Traffic Based Composition of the Test Subset by Attack

FIGURE 9.4: Attack vs. Normal Traffic in the Reduced Testing Sets

9.5 Evaluating ML Algorithms Against AWID

We evaluated the compact version of the AWID dataset against several soft computing algorithms with a twofold objective: on one hand we were interested to ascertain whether the crafted dataset possesses the qualitative attributes that may benefit the detection process, and on the other hand we wanted to give pointers towards the approaches that behave best with the AWID dataset.

9.5.1 Machine Learning Classification

Our experiments were conducted with the Weka framework on an 8-core, Ubuntu 12.04 server, virtual machine with 56 GB RAM, located on the Azure cloud service. The chosen datasets were AWID-CLS-R-Trn and AWID-CLS-R-Tst for training and testing purposes respectively. A summary of the results is shown in table 9.6.

One can notice that the J48 algorithm achieved the best FP, TP rates with the cost of training speed. In particular, J48 required 3921.68 seconds to construct its model from the training data, which is significantly longer than any other method. The Random Forest and OneR achieved the second best TP and FP rate respectively, both an order of magnitude faster.

What is more interesting is that algorithms such as Adaboost, Hyperpipes, ZeroR which present TP rate of 0.922 may initially seem satisfactory, yet these number are misleading. In truth, this illusion is generated by the fact that the portion of normal traffic greatly outnumbers the abnormal one. Actually, these algorithms do nothing more than assigning every record to the normal class, thus achieving TP rates equal to the records of that class. Nevertheless, a closer look at the comparative confusion matrices in 9.7 can verify that these algorithms misclassify all intrusive records.

Regarding the effectiveness of algorithms with respect to separate classes, the normal class was almost infallibly predicted by the OneR algorithm with 99.99% rate. Naive Bayes managed to correctly recognize 72.69% of the flooding records. The injection class was very accurately predicted by the J48 algorithm (99.98%) but records of impersonation class were the toughest ones to recognize with the top performer for the class, namely Random Tree, being able to correctly classify only 7.5% of these records.

9.6 Comparison

Although, the existing datasets have been proved valuable to the research community, all of them seem to have failed to meet the standard requirements as outlined in section 9.3. We argue that under this prism, the evaluation results presented by the majority of the proposed works in literature may exhibit a significant deviation, with respect to the achieved detection rates, when applied in real world.

For example, the CAIDA dataset is comprised by a limited number of events or attacks. Some of the traces contained in that dataset are anonymized ones with their payload maintained, while other traces have their protocol information, destination, and so forth completely removed. This dataset does not fulfil the requirement for integrity, satisfactory size,

The DEFCON dataset is harvested in the course of a Capture The Flag (CTF) competition which states that the traffic contained does not correspond to real-world network traffic.

The DARPA datasets fail to emulate real traffic and several irregularities among the attack instances are introduced due to the synthetic approach this dataset was generated.

The gathering of the AWID dataset was motivated by the evident absence of a satisfactory, publicly available dataset that could be used as a trustworthy benchmark. This lack is even more conspicuous in the wireless realm where not a single dataset specifically targeting this sector ever existed.

On the contrary, AWID dataset aims for intrusion detection in wireless environments at the MAC layer solely. It contains attack and normal traffic from the utilization of a real network. The attacks were unleashed by field experts and were not simulated. The harvesting of the test network lasted 4 days producing more than 30,000,000 records. The AWID includes the majority of the fields of a pcap file except for the payload of each packet is removed along with some other rarely occurring fields. Nevertheless, we claim that the actual data of the payload is of small significance for MAC layer intrusion detection. Moreover, the data contained in the payload are normally encrypted and the inclusion of that field would add tremendous overhead in the dataset. All the versions of the dataset contain labelled records. The tagging was done off-line with the aid of automated tools that made use of signatures defined by field experts. An effort was made to include as many as possible attacks of known and in some cases unimplemented attacks. In many cases variations of these attacks where included. The AWID is an ongoing project committed to correct, update and enrich its datasets with the bleeding edge threats against multiple networks.

A comparison of Datasets frequently used in intrusion detection with respect to the requirements defined in section 9.3 can be seen in table 9.8

Dataset	Area Specific	Realistic	Labelled	Integrity	Satisfactory Size	Variation	Freshness
Darpa	✓	✓	✓	✓	-	-	-
Caida	✓	-	✓	-	✓	-	-
Unibs	-	✓	✓	-	✓	✓	✓
Defcon	✓	-	✓	-	✓	✓	✓
ISCX	-	-	-	✓	✓	✓	✓
Android Genome	✓	✓	✓	✓	✓	✓	✓
KDD99	-	-	✓	-	-	✓	-
AWID	✓	✓	✓	✓	✓	✓	✓

TABLE 9.8: Comparison of Datasets Used in Intrusion Detection

TABLE 9.3: Specifications of the Equipment Used in the Experiments

Node	Type	Brand	OS	Network Card	CPU
Client1	Desktop	Custom	Ubuntu Linux 12.04 LTS	Netgear WNA3100 N300	Intel Core i7 3.2GHz
Client2	Laptop	Fujitsu-Siemens	Ubuntu Linux 12.04 LTS	Intel 3945ABG	Intel Core Duo T2050 1.6GHz
Client3	Laptop	Acer	Ubuntu Linux 12.04 LTS	Qualcomm Atheros AR9462	Intel Core i5 1.7GHz
Client4	Smartphone	iPhone 3G	iOS 4.2	NA	Samsung 32-bit RISC ARM 620MHz
Client5	Other	iPod Touch	iOS 3.1	NA	Samsung 32-bit RISC ARM 533MHz
Client6	Laptop	Acer Aspire 5750G	Windows 7	Broadcom BCM943227HM4L	Intel Core i5 2.8GHz
Client7	Smartphone	HTC Diamond	Windows Phone 6.1	NA	528 MHz ARM 11
Client8	Smartphone	Samsung Nexus	Android 4.2	NA	dual-core ARM Cortex-A9 1.2 GHz
Client9	Tablet	Samsung Galaxy Tab	Android 2.2	NA	Cortex-A8 1 GHz
Client10	Smart TV	LG 42LM7600S	Linux	NA	NA
Attacker	Laptop	Acer Aspire 5750G	Kali Linux 1.0.6	D-Link DWA-125/Linksys WUSB54GC	Intel Core i5 2.8GHz
Monitor Node	Desktop	Custom	Linux Debian 7.3	Alpha AWUS036H	Core i7 2.4Ghz

TABLE 9.4: File Structure of the AWID Collection

Filename refers to the code name of the CSV dataset file, *Classes* refers to the number of classes contained in that particular version of the dataset, *Size* refers to whether the version of the dataset is full or reduced, *Type* refers to whether the version of the dataset is meant for training or testing purposes, *Hours* refers to the number of hours invested in monitoring the network to produce the dataset, *Total Recs* refers to the number of records in the file, *Normal Recs* refers to the normal records in the file, *Attack Recs* refers to the attack records in the file, *Ratio* refers to time analogy of normal to attack traffic in the file, *Filesize* refers to the size of the resulting CSV file in MB, *Compressed* refers to the size of the CSV file compressed with gzip in MB, *Raw* refers to the size of the source PCAP file in MB.

Filename	Classes	Size	Type	Hours	Total Recs	Normal Recs	Attack Recs	Ratio	Filesize	No Files	Raw
AWID-CLS-F-Trn	4	Extended	Training	96	37817835	36732463	1085372	9:1	15100	96	16300
AWID-CLS-F-Tst	4	Extended	Test	12	4570463	4373934	196529	9:1	1700	12	1900
AWID-CLS-R-Trn	4	Compact	Training	1	1795575	1633190	162385	3:2	935	1	948
AWID-CLS-R-Tst	4	Compact	Test	1/3	575643	530785	44858	3:2	297	1	318
AWID-ATK-F-Trn	16	Extended	Training	96	37817835	36732463	1085372	9:1	15100	96	16300
AWID-ATK-F-Tst	16	Extended	Test	12	4570463	4373934	196529	9:1	1700	12	1900
AWID-ATK-R-Trn	16	Compact	Training	1	1795575	1633190	162385	3:2	935	1	948
AWID-ATK-R-Tst	16	Compact	Test	1/3	575643	530785	44858	3:2	297	1	318

TABLE 9.6: Evaluation of Various Classification Algorithms on the 156 Feature Set. Best performer in red.

Algorithm	Correctly Classified%	Incorrectly Classified%	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Time
AdaBoost	92.2073	7.7927	0.922	0.922	0.85	0.922	0.885	0.806	1513
Hyperpipes	92.2073	7.7927	0.922	0.922	0.85	0.922	0.885	0.952	7.79
J48	96.1982	3.801	0.962	0.437	0.954	0.962	0.948	0.759	3921.68
Naive Bayes	89.4323	10.5677	0.894	0.768	0.891	0.894	0.877	0.594	188.21
OneR	94.5758	5.4242	0.946	0.642	0.9	0.946	0.922	0.652	156.98
Random Forest	95.5891	4.4109	0.956	0.52	0.958	0.956	0.941	0.955	828.95
Random Tree	91.4379	8.5621	0.914	0.449	0.914	0.914	0.91	0.733	88.43
ZeroR	92.2073	7.7927	0.922	0.922	0.85	0.922	0.885	0.5	0.63

Normal	Flooding	Injection	Impersonation	Classified As
530785	0	0	0	Normal
8097	0	0	0	Flooding
16682	0	0	0	Injection
20079	0	0	0	Impersonation

(A) Adaboost

Normal	Flooding	Injection	Impersonation	Classified As
530785	0	0	0	Normal
8097	0	0	0	Flooding
16682	0	0	0	Injection
20079	0	0	0	Impersonation

(B) Hyperpipes

Normal	Flooding	Injection	Impersonation	Classified As
530771	8	0	6	Normal
2641	4857	0	599	Flooding
2	0	16680	0	Injection
18629	0	0	1450	Impersonation

(C) J48

Normal	Flooding	Injection	Impersonation	Classified As
508621	22164	0	0	Normal
2189	5908	0	0	Flooding
16400	0	282	0	Injection
18750	1329	0	0	Impersonation

(D) Naive Bayes

Normal	Flooding	Injection	Impersonation	Classified As
530775	0	7	3	Normal
8097	0	0	0	Flooding
3038	0	13644	0	Injection
20079	0	0	0	Impersonation

(E) OneR

Normal	Flooding	Injection	Impersonation	Classified As
530729	1	54	1	Normal
4077	4020	0	0	Flooding
2470	0	14212	0	Injection
18760	0	28	1291	Impersonation

(F) Random Forest

Normal	Flooding	Injection	Impersonation	Classified As
518657	906	716	10506	Normal
3854	4243	0	0	Flooding
338	0	1930	14414	Injection
17550	0	1003	1526	Impersonation

(G) Random Tree

Normal	Flooding	Injection	Impersonation	Classified As
530785	0	0	0	Normal
8097	0	0	0	Flooding
16682	0	0	0	Injection
20079	0	0	0	Impersonation

(H) ZeroR

TABLE 9.7: Confusion Matrices of Various Classification Algorithms on the 156 Feature Set. Best performer in red.

Chapter 10

Extracting Wireless Attack Signatures

Being in possession of attack signatures is not critical solely in anomaly-based intrusion detection but also creates a better understanding of the methods ill-motivated entities follow. This, indirectly, may prove beneficial even for misuse-laden intrusion detection.

The AWID dataset described in chapter 9 contains traffic from the utilization of a real life wireless 802.11 WEP protected network, and is primarily intended to be used as an evaluation tool for ML-based IDS. However, the realistic nature of this traffic database can also function as credible source for extracting valuable conclusions about wireless attack patterns and to document normal traffic of typical small to medium scale 802.11 wireless networks.

In this chapter, we first attempt to fingerprint the normal traffic contained in the compact version of the AWID dataset. In the process, we formulate pattern signatures for some of the most common wireless attacks based on observations on the AWID dataset. To the best of our knowledge this is the first documented attempt to describe the signature of 802.11 attacks in such depth. Based on these revelations and in combination with theoretical hypotheses we apply the extracted conclusions to achieve manual feature reduction on the AWID dataset. Finally, we repeat the experiments discussed in chapter 9 for validating the legibility of our assumptions.

10.1 Formulating Attack Signatures

As a fact, most attacks have inherent characteristics that attest their presence. The evidences may be the result of the theoretic foundations of the attack itself or the digital

footprints left by the tools that were used to execute it. A characteristic example is the the Authentication Request attack: In theory, such attacks can be recognized if one is monitoring for sudden spikes in the number of Authentication Request frames per time unit. In practise, this attack is implemented by the MDK3 tool exclusively. During this mode of execution the tool generates a flood of frames of subtype 0x0b, that all happen to have their Sequence Number field set with 0. This signature is not correlated with the attack on a theoretical level but when used in conjunction with the first it may give a stronger inddication of a Authentication Request flooding attack.

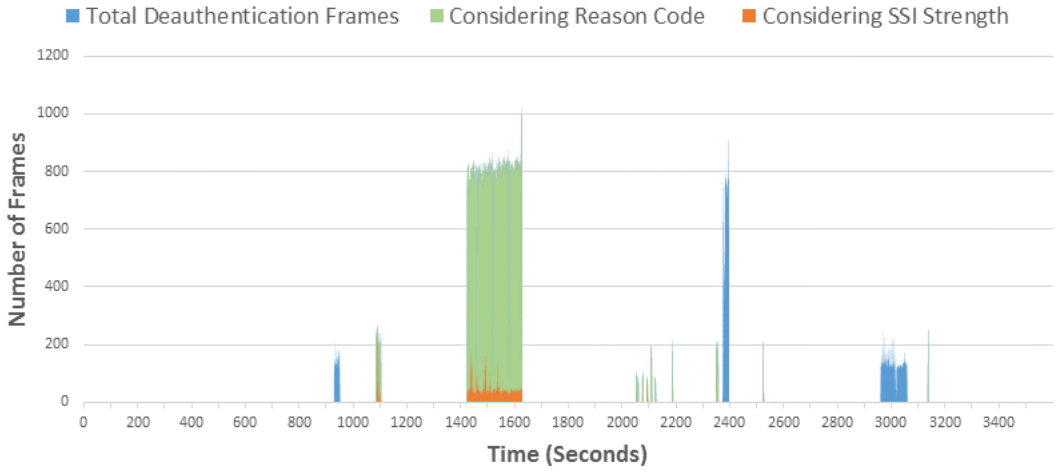
10.1.1 Flooding Attacks

Flooding attacks create a sudden spike of the number of management frames (in their majority) per time unit. Although this observation it easy to discern some type of flooding attack from the normal traffic it is not always as straightforward to distinguish it from other attack types as some may also cause a temporary increase of certain management frames. As will be made clear in the process, additional hints and traces usually exist although they are contingent on the specific tools used during each attack. Generally, there may be certain actions an aggressor can make to masquerade the latter (i.e., the tool specific traces) but there is very little she can do about the proportional increase of management frames. This process is imperative for the effectiveness of all flooding attacks.

Deauthentication Flooding attack is considered one of the most potent DoS attacks in the wireless realm, yet it is also one of the hardest to accurately identify. During its course, a burst of Deauthentication frames is generated, however, elevated levels of such frames may also be tracked in Amok, Dissassociation, Power Saving, Authentication Request Flooding attacks, as well as in the ARP Injection one, in the case where that attack fails (for example, when such frames are transmitted by intruders that possess or impersonate non authenticated MAC addresses). It is important to keep in mind that only in the case of an actual Deauthentication Flooding attack the corresponding frames are forged and transmitted by the adversary. In the rest of the cases they are products of the AP itself as part of a valid response to the attack. As far as the practical aspect of this attack, Aireplay, which is the de-facto tool for launching Deauthentication flooding attacks, transmits management frames that have the same Reason Code (0x0007) and their Sequence Number are out of order.

Figure 10.1a shows the total amount of Deauthentication frames throughout the entire duration of the reduced training set. Notice that only few timeslots contain Deauthentication frames that have the Reason Code field set to 0x0007. Even so, there are cases

where Deauthentication frames with the exact same Reason Code do not correspond to a Deauthentication Flooding attack. For example, the timeframe between seconds 1050 to 1150 contain frames originating from an actual Deauthentication Flooding attack, while the timeframe between seconds 1400 to 1600 contain Deauthentication frames that are produced by the AP as a valid response to a ineffectual ARP Injection attempt. Figure 10.1b focuses in this time zone. Notice that the Signal Strength criterion that is applied as a last resort reveals of the actual Deauthentication Flooding attack. As observed from the figure, in the first time slot there is a significant percentage of packets that deviate from a certain threshold of Signal Strength, while in the second time slot this percentage is kept low.



(A) Deauthentication Management Frames

(B) Zoom on Deauthentication Management Frames During Seconds 950 to 160

FIGURE 10.1: Patterns of Traffic During Deauthentication Flooding Attack

During an Authentication Request Flooding attack the Authentication frames are expected to show a significant increase. Naturally, increased numbers of Authentication

Requests can also be noticed in the Amok as well as the Deauthentication Flooding attacks, but in the case of the Authentication Request Flooding the accumulated volume is much higher. This attack is mainly launched via the MDK3 tool which always transmits Authentication frames with a static Listen Interval field (value 0x0000) and the Tagged Parameters field contain static, fewer in number parameters than usual. Additionally, the sequence number has always the constant value of 0.

A Beacon Flooding attack causes a vigorous increase in the quantity of Beacon frames. Typically, the advertised ESSIDs are new and short-lived (i.e., not many Beacon frames with the same SSID are transmitted) while frequently they have uncanny, randomly generated, names. An increase in Beacon frames occurs naturally in all impersonation attacks too, but in such cases the ESSID has the value of a network that already exists in the vicinity. The MDK3 tool is the only one that offers an implementation of this attack. Similarly to the Authentication Request Flooding attack, the generated frames have a Timestamp field of static value (0x0000000000000000). Secondly, the Sequence Number does not increase and remains 0 for all frames. Finally, the Short Preamble and Short Slot Time flags are simultaneously set to 0. After observing the beacon frames during the attack free periods, the frames that possess all the aforementioned characteristics are practically non existent. Figure 10.2 displays the total number of Beacon frames in the training set, as well as those Beacon frames in that set that meet the MDK3 signature attributes. Note that even with the use of the first filter alone (blue area) it is easy to identify the time frame in which a Beacon attack unfolds with high accuracy, however the use of the second filter (orange area) achieves optimal results.

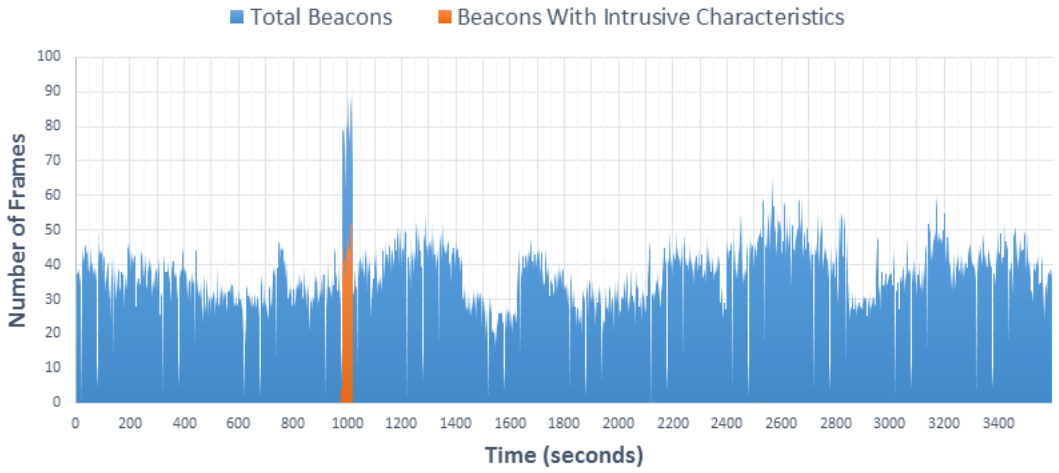


FIGURE 10.2: Traffic Pattern During Beacon Flooding Attack

Probe Response Flooding attack results in an outburst of Probe Response frames. An increase of such frames is also observed during the impersonation assaults but it is generally much milder. The Metasploit tool has a mode of attack (payload), which allows

an aggressor to discharge such attacks. Probe response frames crafted with Metasploit have a totally random sender address i.e., it may not have a valid Organizationally Unique Identifier (OUI), the Beacon Interval field does not have the usual value (which is 0.102400) but rather a random one, and the Sequence Number follows an out-of-order increment.

10.1.2 Injection Attacks

Injection attacks usually cause a deluge of validly encrypted data frames of smaller size.

In ARP Injection attacks the aggressor is inclined to transmit a large number of small data frames for a significant amount of time, hoping to evoke the appropriate response from the network. Currently, Aireplay is the preferred tool of hackers for unleashing attacks of this kind and by analysing the structure of the frames this tool generates, it is obvious that they have identical IV values, something which is statistically impossible to occur in such brief timeframes under normal conditions. Additionally, the DS Status flag is set to 1 which is another indication of an ARP Injection attack.

Figure 10.3 highlights the fact that small sized Data frames may occur under various conditions not necessarily solely on ARP Injection attacks. However, when seeking for small sized Data frames that have repeating IVs one may identify ARP Injection attacks with satisfactory accuracy. The reader should notice time durations between the second 1,400 to the 1,600 and 2,800 to the 3,000 second which refer to ARP Injection attacks. The first case represents a failed attempt since the amount of Data frames that have identical IVs is the same as the total amount of Data frames. On the other hand, the second timeframe corresponds to a successful attack as the number of total small sized Data frames (i.e., ARP Requests plus ARP Responses) is about three times the amount of the small sized Data frames with repeating IVs (i.e., ARP Requests injected by the attacker).

During a Fragmentation Attack the intruder injects a sequence of short, fragmented, data frames. If successful this process usually does not consume more than one second, however if not successful the same procedure will be repeated. The Aireplay tool contains an implementation of this attack and by examining the packets it produces, we notice that all have a static, invalid value in the Destination Address (ff:ff:ff:ff:ff:ed) field, the DS status flag is set to 1, the length of the frame is small (but not fixed) and finally the sequence number is out-of-order. Not surprisingly, the More-Fragments flag is set to 1 and the fragment number field is greater than zero in all (but one) of the fragments in the chain.

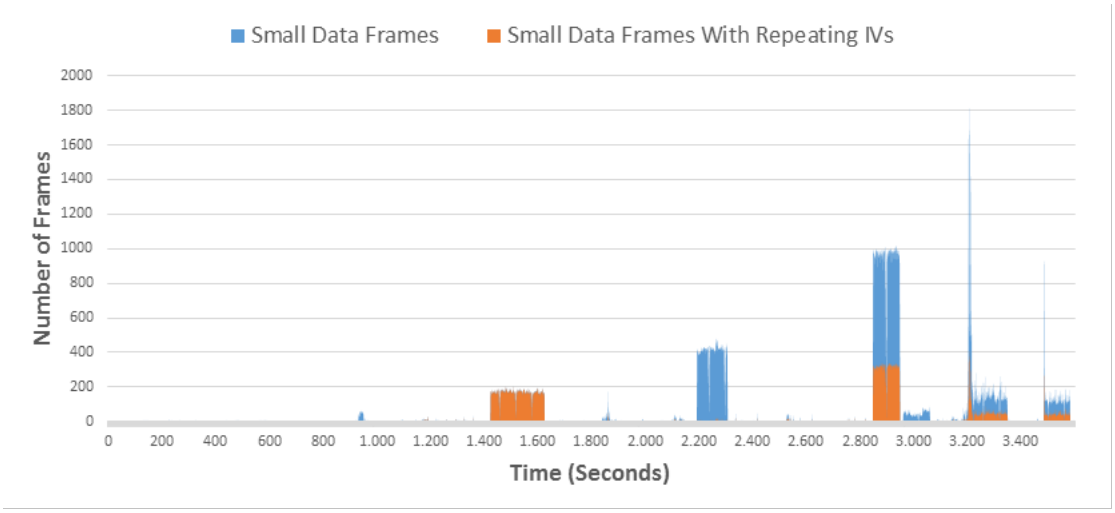


FIGURE 10.3: Traffic Pattern During ARP Injection Attack

10.1.3 Impersonation Attacks

Impersonation attacks introduce an additional AP in the neighbourhood broadcasting Beacon frames that advertise a pre-existing valid network (i.e., that of the victim's). The common denominator of all Impersonation Attacks is that the number of Beacon frames of the victim network is approximately doubled. Quite frequently these attacks are combined with a short flood of Deauthentication frames as an initial step, so that the attacker may force the STAs to connect to its own rogue AP.

Typically, attackers rely on the Airbase tool of the Aircrack suite to launch Evil Twin attacks. As expected, additional Beacon frames are broadcasted but in this case they have significantly different characteristics. For example, the Timestamp field has a fixed value (0x0000000000000000) for all the forged beacon frames, and the Tagged Parameters field contains steadily a different number of parameters.

Figure 10.4 displays the number of Beacon Frames having the ESSID of the victim network. The reader should notice that there are timeframes during which the amount of these Beacons is almost doubled. These durations correspond to impersonation attacks, and this conclusion is verified by the fact that approximately half of these frames possess intrusive characteristics.

Cafe Latte attacks are more complex in nature. As a type of Impersonation attack, these attacks will introduce additional Beacon frames, all having the ESSID of the victim network. As expected, these frames will also bear the same signature characteristics as the ones transmitted during an Evil Twin attack when the Airbase tool is utilized. However, Cafe Latte assaults will simultaneously inject encrypted Data frames of small

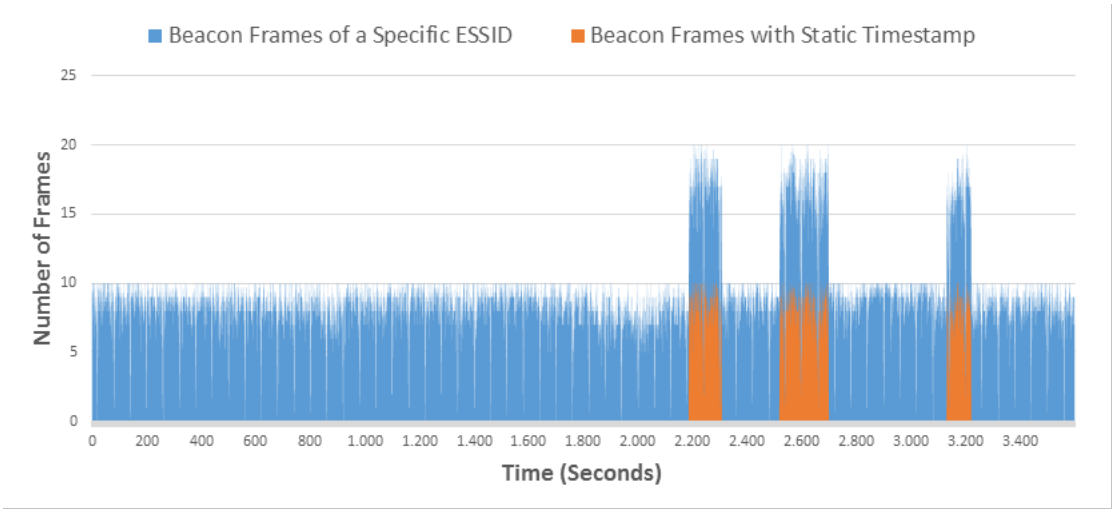


FIGURE 10.4: Traffic Pattern During Evil Twin Attack

size much like an normal injection attack making it harder to clearly distinguish it from an ARP Injection or Evil Twin attack for instance.

As a final note, in all cases described above the received Signal Strength of all forged frames (as indicated by the corresponding Radiotap Header field) will probably fall within a different range of values (usually forged frames have higher Signal Strength) than that of the validly generated ones. This criterion is not undisputed but when applied as statistical mean and combined with other factors, it is usually indicative of an attack.

The attack signatures according to the training set of the AWID dataset can be formulated as following:

- *IF* (Frame Type == Deauth) *AND* (Number of Deauth > AVG_{Deauth})* *AND* (Reason Code == 7) *AND* (Sequence Number Out of Order > $AVG_{SequenceNumber}$) *AND* (Signal Strength ! = $AVG_{Signal_Strength}$) *THEN* Deauthentication Flooding
- *IF* (Frame Type == Auth Request) *AND* (Auth Request > $AVG_{AuthRequest}$) *AND* (Listen Interval == 0) *AND* (Sequence Number == 0) *AND* (Timestamp == 0) *THEN* Authentication Flooding
- *IF* (Frame Type == Beacon) *AND* (Beacon > AVG_{Beacon}) *AND* (Sequence Number == 0) *AND* (Timestamp == 0) *THEN* Beacon Flooding
- *IF* (Frame Type == Probe Response) *AND* (Probe Response > $AVG_{ProbeResponse}$) *AND* (SA == "Other") *AND* (Beacon Interval ! = 0.102400) *AND* (Sequence Number Out of Order > $AVG_{SequenceNumber}$) *THEN* Probe Response Flooding

- *IF* (Frame Type == Data) *AND* (DATA > AVG_{Data}) *AND* (Data Size == "Small") *AND* (Repeated IVs > 0) *THEN* ARP Injection
- *IF* (Frame Type == Data) *AND* (DATA > AVG_{Data}) *AND* (Data Size == 'Small") *AND* (Destination Address == ff:ff:ff:ff:ff:ed) *AND* (DS Status Flag == 1) AND (More Fragments Flag == 1) *THEN* Fragmentation
- *IF* (Frame Type == Beacon) *AND* (Beacon == $2*AVG_{Beacon}$) *AND* (Timestamp == 0) *AND* (ESSID == "Victim") *THEN* Evil Twin
- *IF* (Frame Type == Beacon) *AND* (Beacon == $2*AVG_{Beacon}$) *AND* (Timestamp == 0) *AND* (ESSID == "Victim") *AND* (DATA > AVG_{Data}) *AND* (Data Size == 'Small") *AND* (Repeated IV > 0) *THEN* Cafe Latte

10.2 Attribute Selection Based on Empirical Criteria

In ML-based classification the existence of attributes not directly influencing the belonging class of a record may have a negative impact on the speed and the predictive effectiveness of the classifier. Many attribute selection techniques (also referred to as feature reduction) have been proposed in literature. However, attribute selection in intrusion detection may prove a double edged sword: on one hand it may increase speed by eliminating redundant data and simultaneously improve the detection accuracy, by wiping out misleading fields. On the other hand, it may remove attributes that are exclusively related with unseen malignant behaviour (novel attacks) or attributes related with outliers. Especially in intrusion detection, one does not have the luxury of overlooking outliers since they are tightly coupled with intrusive behaviour.

In accordance with the revelations of section 10.1 we attempted to manually reduce the number of fields describing each record in the AWID dataset. After thorough examination we empirically deduced that only 20 attributes are immediately related to the attacks contained in the training set. In order to estimate the magnitude of the impact in the speed and efficiency of the test algorithms, we repeated the experiments described in 10.1 with the reduced variant of the dataset. The chosen attributes can be seen in table 10.1.

The results of the evaluation indicate that while there was a small (and in many cases negligible) increase in the overall accuracy, there was a definite boost in the training speed of almost all algorithms. More specifically, the shrinkage in training time varied from 10.75% for the Random Forest algorithm to up to 89.35% for the Adaboost algorithm. The only exception to this rule was the ZeroR algorithm which actually required more time.

TABLE 10.1: The Remaining Attributes After Feature Reduction

Explanation	Field
Signal Strength	radiotap_dbm_antsignal
Type of Frame	wlan_fc_type_subtype
To or From Distribution System	wlan_fc_ds
Frame is a Fragment	wlan_fc_frag
Destination Address	wlan_da
Source Address	wlan_sa
Fragment Number	wlan_frag
Sequence Number	wlan_seq
Preamble	wlan_mgt_fixed_capabilities_preamble
Short Slot Time	wlan_mgt_fixed_capabilities_short_slot_time
Listen Interval	wlan_mgt_fixed_listen_ival
Timestamp	wlan_mgt_fixed_timestamp
Beacon	wlan_mgt_fixed_beacon
Reason Code	wlan_mgt_fixed_reason_code
Size of Tagged Parameters	wlan_mgt_tagged_all
ESSID	wlan_mgt_ssid
IV	wlan_wep_iv
Extended IV	wlan_tkip_extiv
Data Length	data_len
Label	-

Likewise, we observed a improvement in the classification precision which was of milder intensity. More specifically, the achieved gain ranged from 0.1% for the Random Forest algorithm to 4.5% for the Random Tree one. Analytical results can be seen in table 10.3.

Interestingly, only the Naive Bayes algorithm showcased a significant gain in prediction accuracy of the impersonation attacks. Yet, with 4,419 correctly classified instances out of 20,079 in total for the class (22%) the achievements are still unsatisfactory. In this round of experiments the Naive Bayes algorithm actually, presented the lowest FP rate, the J48 achieved the highest TP rate while Random Forest had the best accuracy for the Normal class, Naive Bayes for the Flooding and Impersonation classes, and J48 for the Injection one.

In conclusion, the experimental results confirm that out of the 156 features in the AWID dataset, 20 have the most important role in predicting malicious traffic. The fact that not only the prediction accuracy didn't drop but slightly increased verifies that our empirical evaluations that led to manual feature reduction were correct.

TABLE 10.2: Evaluation of Various Classification Algorithms on the 20 Feature Set. Best performer in red.

Algorithm	Correctly Classified%	Incorrectly Classified%	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Time
AdaBoost	92.2073	7.7927	0.922	0.922	0.85	0.922	0.885	0.673	161.12
Hyperpipes	92.2363	7.7637	0.922	0.919	0.879	0.922	0.885	0.935	3.52
J48	96.2574	3.7426	0.963	0.436	0.962	0.963	0.948	0.752	568.92
Naive Bayes	90.5504	9.4496	0.906	0.399	0.917	0.906	0.909	0.774	29.67
OneR	94.5741	5.4259	0.946	0.642	0.9	0.946	0.922	0.652	156.98
Random Forest	95.8247	4.1753	0.958	0.493	0.959	0.958	0.944	0.958	739.78
Random Tree	96.2258	3.7742	0.962	0.438	0.959	0.962	0.948	0.762	49.3
ZeroR	92.2073	7.7927	0.922	0.922	0.85	0.922	0.885	0.5	3.65

TABLE 10.3: Confusion Matrices of Various Classification Algorithms on the 20 Feature Set. Best performer in red.

Normal	Flooding	Injection	Impersonation	Classified As
530785	0	0	0	Normal
8097	0	0	0	Flooding
16682	0	0	0	Injection
20079	0	0	0	Impersonation

(A) Adaboost

Normal	Flooding	Injection	Impersonation	Classified As
530785	0	0	0	Normal
8097	0	0	0	Flooding
16515	0	167	0	Injection
20079	0	0	0	Impersonation

(B) Hyperpipes

Normal	Flooding	Injection	Impersonation	Classified As
530588	116	6	75	Normal
2553	5544	0	0	Flooding
2	0	16680	0	Injection
18644	148	0	1287	Impersonation

(C) J48

Normal	Flooding	Injection	Impersonation	Classified As
497199	8971	11899	12716	Normal
2123	5974	0	0	Flooding
3027	0	13655	0	Injection
14187	1473	0	4419	Impersonation

(D) Naive Bayes

Normal	Flooding	Injection	Impersonation	Classified As
530765	0	14	6	Normal
8097	0	0	0	Flooding
3038	0	13644	0	Injection
20079	0	0	0	Impersonation

(E) OneR

Normal	Flooding	Injection	Impersonation	Classified As
530746	1	1	37	Normal
2600	5497	0	0	Flooding
2763	0	13893	0	Injection
18607	0	28	1472	Impersonation

(F) Random Forest

Normal	Flooding	Injection	Impersonation	Classified As
530700	3	0	82	Normal
2442	5494	161	0	Flooding
273	0	16253	156	Injection
18609	0	0	1470	Impersonation

(G) Random Tree

Normal	Flooding	Injection	Impersonation	Classified As
530785	0	0	0	Normal
8097	0	0	0	Flooding
16682	0	0	0	Injection
20079	0	0	0	Impersonation

(H) ZeroR

Chapter 11

Termid: Robust Prediction of Spurious Network Traffic

This chapter describes the implementation of a distributed execution strategy of the Ant-Miner algorithm. Originally proposed by Parpinelli et al. [125], the algorithm applies an ant colony optimization heuristic to the classification task to discover classification rules. The algorithm is modified accordingly, and applied to a multicore/cluster environment. The performance evaluation is conducted against the AWID dataset and the acquired results show significant improvement with respect to execution time. More interestingly, this approach seems to indirectly increase the detection accuracy by exploring a larger portion of the search space, thus extracting more high quality rules.

11.1 Introduction

Anomaly detection is one of the most important applications of data mining. Due to variate nature of the problem many numerous approaches have been proposed, the most important of which were analysed in chapter 7. When applied in intrusion detection each of these approaches has its own limitations and inefficiencies, including low prediction rates, convergence to suboptimal solutions, poor escalation behaviour as the dataset size increases, low training speed and incomprehensible (to humans) decision models.

Among these approaches, the discovering of classification rules is a promising one. Essentially, this technique creates a number of grammar rules to define classes in a dataset. Evolutionary algorithms are frequently employed for rule discovery, but interesting results are obtained by the application of SI (for further details the reader should refer to section 8.1) for rules extraction.

Parpinelli, et al. [125] were the first to apply Ant Colony Optimization (ACO) for extracting classification rules with success. One of the advantages of their proposed algorithm, Ant-Miner, is that it generates simple and comprehensive rules that can be reviewed, validated and altered if necessary by humans. That is, comprehensibility is important in some highly sensitive application scenarios (e.g., security, health) where the discovered knowledge has to be reviewed by a human expert. What is more, the ACO nature of the algorithm contributes to its flexibility and robustness arguably surpassing the traditional approaches, in some datasets. More specifically, their method uses a heuristic value based on entropy measure. On the downside, the training speed of this approach is unsatisfactory, being several orders of magnitude higher than traditional approaches such as the C.45 [209] classifying algorithm for example. This fact alone has discouraged the application of the Ant-Miner algorithm in intrusion detection.

In this chapter we present a classification rule extraction algorithm based on Ant-Miner. Our algorithm is appropriate for large datasets such as the ones produced through monitoring the traffic of large wireless networks for example. The algorithm is extended and altered in order to be able to run in a highly distributed cluster environments. We apply our proposed approach in the field of intrusion detection. Through empirical results obtained from experimental evaluation on AWID and several other datasets, we prove that our approach produces more accurate and more comprehensive rules than the ones obtained by the best conventional approaches. Although one can argue that the distributed nature of the algorithm has high deployment complexity, we juxtapose that the training phase of the classification engine, after the proposed ameliorations, is remarkably fast, and due to low requirements of algorithm in memory the overall financial cost is maintained low.

11.2 Description of the Ant-Miner Algorithm

Ant-Miner is a supervised rule extraction algorithm which combines a measure of entropy and ACO techniques to complete the classification task of data mining efficiently. In this approach each classification rule follows the format:

$$IF\{term_1\ AND\ term_2\ AND\ \ldots\ AND\ term_n\}\ THEN\ Class_i$$

where each term consists of three parts: attribute atr_i, operator $=$, value val_j. The value is constrained to the domain of the respective attribute. The operator is always the equality, since Ant-Miner supports only nominal values.

The algorithm requires a number of ants to be given as a parameter. Each of these is responsible for the incremental construction or modification of a single classification rule.

A generation of ants lasts until every ant has constructed a rule or until a series of ants has constructed the exact same rule. That is the solution is converging to a specific rule. The ants construct a rule, die and then the next generation of the same number of ants takes over and repeats the same procedures. For each generation, the rules constructed by all ants are maintained on memory and the best of them is chosen to be added to the ordered list of discovered rules. Then, the cases of the dataset that are covered by the chosen rule, are removed from the training set. Thus, each subsequent generation of ants works with a reduced dataset. The generations succeeds one another for as long as the dataset contains more cases than a user-defined threshold. A high level overview of the algorithm can be seen in listing 1.

Algorithm 1 Ant Miner

Require: $List\_of\_Training\_Instances$,
$Max\_Uncovered\_Cases$,
$Number\_of\_Ants$,
$Number\_of\_Rules\_Converging$

$Training\_Set \leftarrow List\_of\_Training\_Instances$
$Discovered\_Rules\_List \leftarrow \{\}$
while LENGTH($Training\_Set$) $\geq Max\_Uncovered\_Cases$ **do**
  $ant = 1$ ▷ Index of ants
  $convergence = 1$ ▷ Index of the rules converging
  PHEROMONEINIT( ) ▷ Initialize with the same pheromone
  repeat
    $R_t \leftarrow$ CONSTRUCTRULE() ▷ Incrementally construct a rule by adding one term at a time
    $R_t \leftarrow$ PRUNERULE(R_t) ▷ Remove irrelevant terms
    ADDTOLIST($Current\_Generation\_Rule\_List$, R_t) ▷ Add the rule to a temporary list
    $Q_t \leftarrow$ CALCULATERULEQUALITY(R_t) ▷ Calculate the quality of the rule
    UPDATEPHEROMONE(R_t) ▷ Increase the pheromone on the trails followed by the ant, decrease on the rest
    if ($R_t == R_{t-1}$) **then**
      $convergence \leftarrow convergence + 1$
    else
      $convergence \leftarrow 1$
    end if
  until $(ant \geq Number\_of\_Ants) OR (convergence \geq Number\_of\_Rules\_Converging)$
  $R_{best} \leftarrow$ CHOOSEBESTRULE($Current\_Generation\_Rule\_List$) ▷ Retrieve the best quality rule among all ants
  ADDTOLIST($Discovered\_Rule\_List$, R_{best}) ▷ Save that rule
  $Covered\_Cases\_List \leftarrow$ CALCULATECASESCOVERED(R_{best}) ▷ Get the instances correctly covered by that rule
  REMOVEFROMLIST($Training\_Set$, $Covered\_Cases\_List$) ▷ Remove these instances from the dataset
end while

From the description above it is clear that the main procedures of the algorithm are: the rule construction, the rule pruning, and the pheromone updating. These three basic routines will be analysed in the following.

11.2.1 Pheromone Initialization

On each generation of ants all terms τ_{ij} are associated with the same amount of pheromone, so that when the first ant starts its search, all paths have the same amount of pheromone, thus the choice of a term is independent of the pheromone factor. Since each term $term_{ij}$ can be perceived as a segment of a path that can be followed by an ant, the initial amount of pheromone deposited at each path is:

$\tau_{ij} = \frac{1}{\sum_{i=1}^{a} b_i}$

where a is the total number of attributes and b_i corresponds to the number of possible values of that attribute.

11.2.2 Selecting Terms

In Ant-Miner each ant starts with an empty rule, and adds one term at a time to its current partial solution. The rule is extended with new terms until: (a) either all attributes have been used for rule construction, (b) any new term added would simply reduce the quality of the rule, by making it cover less cases than a user-defined threshold. The choice of a term to be added next depends on (a) a heuristic function η, and (b) on the amount of pheromone associated with that term τ. More specifically, the probability is given by the equation:

$P_{ij} = \frac{\eta_{ij}\tau_{ij}(t)}{\sum_{i=1}^{a} x_i \sum_{j=1}^{b_i}(\eta_{ij}\tau_{ij}(t))}$

where η_{ij} is the value of a problem-dependent heuristic function for term $term_{ij}$, this value is based on the entropy of the term and will be analysed in greater extend in the process, τ_{ij} is the amount of pheromone associated with the term, a is the total number of attributes in the dataset, b_i is the number of possible values for the *ith* attribute, and x_i is value indicating whether attribute A_i has been used for the construction of the current partial rule or not.

11.2.2.1 Heuristic Function

The heuristic function plays a pivotal role in the selection of the terms to be added to a candidate rule. In essence, it reflects the amount of entropy associated with a term. For each $term_{ij}$ of the form $A_i = V_{ij}$ the heuristic function is:

$$\eta_{ij} = \frac{log_2(k) - H(W|A_i=V_{ij})}{\sum_{i=1}^{a} x_i \sum_{j=1}^{b} log_2(k) - H(W|A_i=V_{ij})}$$

The entropy denoted as $H(W|A_i = V_{ij})$ in the equation is defined as:

$$H(W|A_i = V_{ij}) = -\sum_{w=1}^{k}(P(w|A_i = V_{ij}))log_2(P(w|A_i = V_{ij}))$$

where W is the class, k is the total number of classes in the dataset, $P(w|A_i = V_{ij})$ is the empirical probability of a term having attribute A_i set to value V_{ij} to belong to the class to the total number of terms with attribute A_i set to value V_{ij}. More specifically:

$$P(w|A_i = V_{ij}) = \frac{freq_{T_{ij}^w}}{|T_{ij}|}$$

11.2.3 Rule Pruning

Pruning is a process which (a) aims to increase the quality of the generated rules, (b) contributes to the elimination of the overfitting of the rule to the training cases, and (c) promotes the generation of simpler rules. This process iteratively removes, irrelevant terms, that might have been included in the rule. A cycle of term removal and quality evaluation occurs for as long as the rule is left with just one term or until there is no term whose removal will further improve the quality of that rule. The quality of the reduced rule is evaluated using the equation presented in subsection 11.2.4.

11.2.4 Pheromone Updating

The amount of pheromone of a term $term_{ij}$ that is part of the rule discovered by an ant, is increased proportionally to the quality of that rule. This reflects that the probability of that term to be re-chosen in the future, is increased, with respect to the level of accuracy of that rule. The quality of a given rule, is calculated as:

$$Q = \frac{TP}{TP+FN}\frac{TN}{FP+TN}$$

where Q is the quality of the rule and TP, TN, FP, FN are the same metrics defined in section 9.1.4.

The pheromone associated with a term is increased according to the formula:

$\tau_{ij}(t+1) = \tau_{ij}(t) + Q\tau_{ij}, \forall i, j \in R$

where $\tau_{ij(t)}$ is the pheromone associated with the term in the current generation t, $\tau_{ij}(t+1)$ is the pheromone associated with the term at the next generation, R is the set of terms occurring in the discovered rule.

To reduce the probability of an irrelevant term to be chosen in the future, the amount of pheromone associated with each $term_{ij}$ that does not occur in the discovered rule must be subject to an analogous decline. This can be seen as the virtual equivalent of pheromone evaporation, observed on real ant colonies. Precisely, the pheromone decrease is calculated as:

$\tau_{ij}(t+1) = \frac{\tau_{ij}(t)}{\sum_{i=1}^{a}\sum_{j=1}^{b}\tau_{ij}(t)}$

11.2.5 Classifying New Instances

After the training set has constructed a model, the contained rules must be applied in the order they were discovered, for a test case to be classified. If the antecedents of a discovered rule satisfy the test case, then that case is assigned to that rule's consequent. If no rule on the list covers the test case, then that case is assigned to the majority class of those training instances, left uncovered by any of the discovered rules.

11.3 Previous Work

Ant-Miner is a relatively recent algorithm and its potential is not fully exploited to date. However, there are numerous works that have attempted to alleviate its inefficiencies and increase its speed or accuracy. This section will briefly describe the most important of these approaches. Recall that a study of applications of this algorithm in intrusion detection is included in section 8.1.1.1.

The work in [210] incorporates an alternative heuristic function, which is based on a simple sample density estimation, that is argued to be less computational intense than the original one. This heuristic measure may be less accurate, but the authors advocate that pheromone mechanism compensates the possible errors of the heuristic values. More specifically, the new heuristic function is defined as:

$\eta_{ij} = \frac{C_{T_{ij}}}{|T_{ij}|}$

where $C_{T_{ij}}$ is the class of the majority of the instances in partition T_{ij}. Through experimental evaluations the authors show that this method has equal performance as the original Ant-Miner.

In [211] the authors are motivated by the observation that during a generation of ants the pheromone of each term changes, while η remains the same. In this way, subsequent ants tend to choose the same terms used in the rule constructed by the previous ants, because the pheromone consecration of these terms is increased. This leads to a failure to produce alternative rules, and as a result, the ants converge to a single rule too fast. The authors propose a new pheromone updating method and a new state transition rule to increase the accuracy of classification. Moreover, the pheromone update is subject to the following equation:

$$\tau_{ij}(t) = (1-p)\tau_{ij}(t+1) + (1 - \frac{1}{1+Q}\tau_{ij}(t-1), \forall i,j \in R$$

where p is a parameter reflecting the pheromone evaporation rate.

On the other hand, the term selection strategy depends not only on the heuristic function η_{ij} and pheromone τ_{ij}, but also on two new newly introduced random numbers.

ACO-Miner [212] is an improved version of the Ant-Miner that incorporates a better term selection rule, a more appropriate pheromone updating rule, and an alternative heuristic function. The authors stress that the original Ant-Miner suffers by the following shortcomings: the term selection is computationally expensive and lacks of balancing between exploration and exploitation, (b) discovery of new rules is undermined if the quality measure Q is very small, and (c) the entropy of $term_{ij}$ is always the same regardless of the contents of the rule in which the term occurs. Therefore, they proposed that the term selection process should be subject to newly introduced parameters such as the relative importance of trail (which states that if there has been a lot of traffic on $term_{ij}$ then it is highly desirable) and its visibility (which dictates that close terms should be favoured).

The authors in [213] propose a new rule pruning technique for Ant-Miner, which is said to further reduce the size of the discovered rules and at the same time decrease its computational cost. In this case, only the rules with more terms than the user-specified value r are subjected to the pruning mechanism and they are shortened strictly up to that size. For each term within the rule antecedent, the probability of selecting that term to be removed is calculated based on the term's information gain (which is pre-computed) and a random value, result of a roulette wheel selection technique. The r remaining terms are fed into the original rule pruning mechanism. This hybrid approach may be faster, however the results indicate that it leads to reduction of accuracy.

The authors in [214] propose an alternative approach which produces an unordered set of classification rules, unlike the traditional approach in which the interpretation of any rule requires knowledge of all the previously discovered ones. This fact contributes to the better readability of the constructed rules. In fact, this result is achieved by

having groups of ants searching/constructing rules of a specific class only. The problem dependent heuristic function, has been adapted accordingly, in this case, it is the Laplace-corrected confidence for each term, which is calculated as:

$\eta_{ij} = \frac{|term_{ij},k|+1}{|term_{ij}|+V}$

where $|term_{ij}, k|$ is the number of training instances that have the term $term_{ij}$ and class k, $|term_{ij}|$ is the number of training cases having $term_{ij}$, and V is the number of possible values in the class attribute's domain.

The work [215] presents an extension to Ant-Miner, namely the cAnt-Miner, which is able to cope with continuous valued attributes during the rule construction process. cAnt-Miner does not requires a discretization method in a preprocessing step since it is able to generate discrete intervals of the continuous attributes "on-the-fly". This operation benefits the discovery of more accurate rules than the traditional approaches which apply the discretization process on a preprocessing phase. Naturally, the entropy calculation procedure cannot be straightforwardly applied to attributes with continuous values. For this reason, the entropy of the a term $term_i$ is calculated by selecting a threshold value v to dynamically splits the continuous attribute a_i into two parts $ai < v$ and $ai \geq v$. Then, the best threshold value is the value v that minimizes the entropy of the partition.

The approach presented in [216] describes more extensive alternations to the to original Ant-Miner. More specifically, the authors introduce the concept of multiple types of pheromone, each one governing a different class of the dataset. That is, a different pheromone update strategy for penalizing low-quality rules and rewarding the high-quality ones, by incorporating a quality contrast intensifier is also considered. Additionally, the proposed scheme supports the logical negation operator, in the antecedents of constructed rules, thus resulting into the creation of more dynamic and accurate rules. Moreover, this work adopts the concept of "stubborn ants", in which an ant chooses terms considering its own previous experience. A final modification provisions that ants choose their own values for specific parameters thus having in a way, their own personality.

11.3.1 Parallel Approaches

Ant-Miner is provably efficient in its prediction tasks. We argue that one of the aspects of this algorithm which is heavily neglected is the training speed factor of the algorithm. Most of the approaches include experiments with medium sized datasets which may give the illusion that algorithm converges fast. Indeed, this fact was diagnosed relatively

early and some approaches exist that have attempted to speed up the rule discovery process by adopting parallel computation approaches.

The [217, 218] were the first works to bring the Ant-Miner algortithm to the parallel realm. In this approach each processor is assigned to a specific class and all the rules it discovers must cover it. A group of ants are allocated on each processor to search for the antecedent part. Driven by the observation that most works neglect the importance of attributes to the efficiency of classification, the authors modify the Ant-Miner algorithm accordingly so that the term selection process will take into account the importance of each attribute to the class, along with the pheromone, and the heuristic information. To do so, the authors utilize the Bayes discriminate analysis to measure the influence of attribute j to class p. Moreover, their approach prunes the rules during the process of rule construction which is demonstrated to significantly reduce the time complexity and improve the solution quality.

In the same context the work in [219] describes an adaptation of Ant-Miner algorithm, appropriate for large datasets, namely Parallel Ant-Miner2. This model provisions that ants are first separated into groups; each assigned to a different processor. The ants of each group are responsible for producing rules for a single class only. Additionally, to complete the pheromone update process, ants are allowed to communicate other ants in the same group as well as with the best ant of any other group. This helps ants eliminate irrelevant terms of the rule. More specifically, the pheromone update rule for ants within the same group is:

$$\tau_{ijp}(t+1) = (1-p)\tau_{ijp}(t) + p\frac{\sum \tau_{ijp}(t)}{|X|}$$

where $|X|$ is the number of ants on processor p, and $\sum \tau_{ijp}(t)$ is the sum of all pheromone produced by all ants on processor p for $term_{ij}$.

On the other hand, the pheromone update rule that applies to the communication with the best ant of other groups is:

$$\tau_{ijp}(t+1) = (1-p)\tau_{ijp}(t) + p\frac{\sum_{p=1}^{G} \tau_{ijp_{best}}(t)}{|G|}$$

where $|G|$ is the number of groups/classes, $\tau_{ijp_{best}}(t)$ is the best pheromone produced by the ants of processor p, and $\sum_{p=1}^{G} \tau_{ijp_{best}}(t)$ is the summation of the pheromones produced by the best ants of all the processors.

The authors in [220] propose a parallel version of Ant-Miner developed according to the master-slave model. The authors were motivated by the fact that discretization of continuous attributes as well as rule construction are the most computationally expensive operations of the algorithm. Initially, the master node broadcasts the training set to all other slave nodes, along with the initial values of pheromone. Each slave node has

a group of ants, which are responsible for the construction of local rules. When all rules are generated, they are send to the master node. Upon reception, the master node moves on to the evaluation of the quality of the constructed rules, the reduction of the attributes, and the pheromone update.

11.4 Termid: A Distributed Ant-Miner Strategy for Intrusion Detection

The distributed nature of our proposed approach renders Ant-Miner suitable for wireless intrusion detection environments. The reader should keep in mind that in such scenarios, a number of monitor nodes is spread within the coverage of the network. These nodes produce their own portions of the dataset which may contain totally different instances or partially overlapping ones. Under these circumstances, the conventional approaches require the following actions to be made:

- Extract the partial dataset. Due to the sensitivity of their contents, the partial datasets have to be securely transmitted from the monitor nodes to a central location for further processing.
- Concatenate the partial datasets. This process is not trivial as identical records may exist among partial datasets (probably captured by neighbouring monitor nodes), which have to be eliminated in the aggregated one.
- Retransmit the entire dataset to a detector node. The transmission of the unified dataset to a central point raises concerns regarding the security and time/bandwidth requirements of this action. Additionally, the detector node should have enough resources to handle the processing of large datasets.

Therefore, it should be stressed that, our approach has the advantage of not requiring the transmission of the dataset or any part of it. On the contrary, our model provisions that only conclusions (rules) about the actual data can be transmitted instead. Inductively, since the data exchanged in the network will have much smaller size, commotion to the network is avoided. Additionally, each node can work with a partial dataset which allows for the IDS to be integrated into the monitor node. Even more, since each processor is able to work with a subset of the entire dataset the training completes much sooner, and for the same reason, the memory requirements of the classifier are maintained in low levels.

11.4.1 Description of The Solution

Our distributed model coined as *Termid*, follows the master/slave architecture. The master node is using the slave nodes as oracles, to complete the selection of the best global rule, but itself it has no knowledge of the data. Here, the term "master node" may be misleading, as these nodes can be perceived as the virtual counterpart of the environment, a terrain on which only the fittest survive, and not as a centralized authority that dictates the behaviour of ants. Therefore, the term environment is adopted for such nodes.

The slave nodes evaluate the quality of a rule, extracted by each other node, according to their own limited knowledge of the dataset. The selection is made based on the overall quality of the rules the master node possess. The term ants is used for the slave nodes.

Initially, each trail will be assigned with the same amount of pheromone by the environment node according to the formula described in 11.2.1. In this context, a trail represents a unique combination of attribute/value pairs.

The ant will construct the rule by adding one term at a time. In this case, the decision of the next term to be added to the rule will be based solely on the knowledge of the partial dataset. Let $LocalRule_p$ be the rule produced from ant p in this step. Note, that each ant is agnostic of all other portions of dataset and has no means of direct communication with the rest.

After $LocalRule_p$ is pruned, it will be broadcasted to each other ant through the environment. The process of pruning is similar to that described in 11.2.3. The environment node will broadcast the $LocalRule_p$ to the rest of the ants, asking their opinion about its quality. In this step, each ant calculates an evaluation list comprising of TP, TN, FP, FN values (as defined in 9.1.4) against their local partial training set. If the rule $LocalRule_p$ does not cover any case in the local set or if it is empty, then apparently only the TN and FN values will be > 0.

Upon receiving responses from all ants, the environment node will calculate the global quality of each local rule based on all the evaluation lists. Then, it will broadcast the best of e rules $GlobalRule$ to every ant. Based on this variable, the ants will remove the instances that match $GlobalRule$, thus reducing the corresponding local dataset.

At the final step, the environment node will update the pheromone trails, i.e. pheromone value associated with each term, and broadcast the new pheromone to all ants. A high level description of the processes executed by the slave nodes and the master one can be seen in algorithms 2 and 3 respectively.

Algorithm 2 Termid - Ant Side

Require: $Local\_Training\_Instances$,
$Max\_Uncovered\_Cases$,

1: $Training\_Set \leftarrow Local\_Training\_Instances$
2: $pheromone \leftarrow$ receiveFromEnvironment()
3: **while** Length($Training\_Set$) $\geq Max\_Uncovered\_Cases$ **do**
4: $LocalRule_p \leftarrow$ constructRule()
5: $LocalRule_p \leftarrow$ pruneRule($LocalRule_p$)
6: sendToEnvironment($LocalRule_p$, p)
7: $RuleList \leftarrow$ receiveFromEnvironment()
8: $EvaluatedRuleList \leftarrow$ evaluateRule($RuleList$)
9: sendToEnvironment($RuleEvaluationList$)
10: $GlobalRule_i \leftarrow$ receiveFromEnvironment()
11: $Covered\_Cases\_List \leftarrow$ calculateCasesCovered($GlobalRule_i$)
12: removeFromList($Training\_Set$, $GlobalRule_i$)
13: $pheromone \leftarrow$ receiveFromEnvironment()
14: **end while**
15: $Status \leftarrow completed$

Algorithm 3 Termid - Environment Side

1: CalculateInitialPheromone( )
2: **while** $number\_of\_completed \neq N$ **do**
3: **for all** p in AntList **do**
4: $Rule_p \leftarrow$ receiveFromAnt()
5: sendToAnt($Rule_p$, $AntList - p$)
6: $Evaluation_{Rule_p} \leftarrow$ receiveFromAnt()
7: $EvaluatedRuleList \leftarrow$ calculateQuality($Evaluation_{Rule_p}$)
8: **end for**
9: $GlobalRule \leftarrow$ getBestRule($EvaluatedRuleList$)
10: updatePheromoneOnTrails($GlobalRule$)
11: sendToAnt($GlobalRule$, $AntList$)
12: **end while**

11.5 Evaluation

Our experiments were conducted with 5 virtual machines (1 for environment node, 5 for ant nodes) with 1GB of memory and 1 core each running on a single physical machine (Intel i7 with 8 GB or memory, on an SSD hard drive). We must underline that since the virtual machines were running on the same physical one, the communication part was neglectable, although this would not be the case in real life conditions. Also, take into account, that an underlying subsystem for health monitoring the nodes was also running on the same machine but its specification is outside the scope of this study.

11.5.1 Complexity Analysis

In order to analyse the computational complexity of the distributed Ant-Miner strategy both the ant and environment nodes need to be taken under consideration.

First off, the environment node sets the initial pheromone value to all possible terms. The values are kept fixed until the update step. Their calculation requires $O(\alpha \cdot \upsilon)$ where α is the number of attributes and υ is then number of possible values per attribute. Then, the loop begins, a single iteration of which involves the following actions:

- The internal loop which will be executed for N times which is equal to the number of ants. This loop involves 4 operations, thus it is $N \cdot c_1 + N \cdot c_2 + N \cdot c_3 + N \cdot c_4$ and the resulting complexity is $O(N)$.
- The *getBestRule*() subroutine is a classic maximum selection function with an average complexity $O(N)$.
- The next step, namely pheromone updating, aims at of increasing the value of the pheromone variable of the terms used in the rule and decreasing the pheromone of unused terms. The former process consumes $O(\kappa)$ where κ is the number of attributes used in the *GlobalRule* and the latter takes $O(\alpha)$. Since $\kappa \leq \alpha$ the whole process of pheromone update takes $O(\alpha)$.

Therefore, the total complexity of a single while loop executed in the environment node operations is $O(N) + O(N) + O(a)$ which collapses to $O(N)$ especially because α is a small value.

Each ant node iteratively performs a set of tasks which essentially are reduced to the following:

- *Rule Construction* - the choice of the term to be added to the current rule requires the evaluation of the probability P (defined in 11.2.2), which in turn involves the calculation of the heuristic function η and pheromone τ. The values of both heuristic function and pheromone are precomputed as a preprocessing step and remain constant, throughout the construction of the rule. Thus, the complexity of rule construction is $O(\kappa \cdot \alpha)$.
- *Rule Pruning* - this process involves the evaluation of a steadily decreasing number of terms for each candidate rule. In the first iteration, κ potential new rules (each consisting of a different term removed), are evaluated. The second pruning iteration evaluates $\kappa-1$, the third $\kappa-2$ and so on. Therefore, the entire rule pruning

process consumes at most $(\kappa-1)\cdot\kappa+(\kappa-2)\cdot(\kappa-1)+(\kappa-3)\cdot(\kappa-2)+\cdots+1\cdot 2$ which results to $O(\kappa^3)$.

- *Rule List Evaluation* - this process involves (a) the counting of instances matching each rule (TP) in the list given by the environment node, (b) the ones covered but not matching (FP), (c) those not covered but with the same class (TN), and (d) those not covered but with different class (FN). Since the number of rules is equal to the number of ants minus its own, and there are 4 operation involved, the total complexity is $O(N \cdot L_p)$ where L_p is the size of the training set in ant node p. Note, that in an ideal case, the size of the L_p is $\frac{G}{N}$ where G is the size of the entire dataset. Additionally, the size of the training set will gradually decrease.

- *Covered Cases Removal* - this step iteratively checks whether the $GlobalRule_i$ matches each instance in the local training set. Matching involves the comparison of the terms in the rule with the corresponding attribute values of each instance. Therefore, the complexity of this operation is $O(L_p \cdot \kappa)$. After a matching is done, the instance is removed from the training set.

Considering all the above, the total complexity of a single while loop of an ant node will be $O(\kappa\cdot\alpha)+O(\kappa^3)+O(N\cdot L_p)+O(L_p\cdot\kappa)$ which reduces to $O(k^3+N\cdot L+L\cdot\kappa)$. The reader should note that unlike the centralized Ant-Miner algorithm, where the complexity is further multiplied with the number of ants, in our approach, this operation does not occur thus leading to a much effective overall training phase.

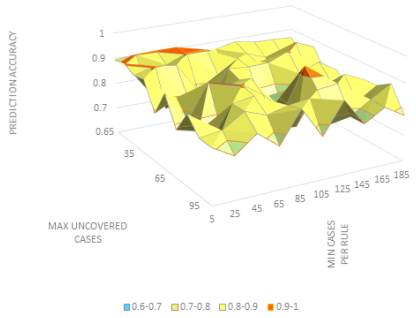
11.5.2 Accuracy Against Toy Datasets

Our first set of experiments was conducted against some small, publicly available datasets that are commonly used in data mining. This was done primarily to validate the prediction accuracy of our approach with some easy benchmarks without caring about the training speed. Secondarily, through these experiments we were given the opportunity to study the influence of the user-defined parameters, required by the algorithm, upon the predictive accuracy and training speed. The basic aspects of these datasets are briefly presented in table 11.2.

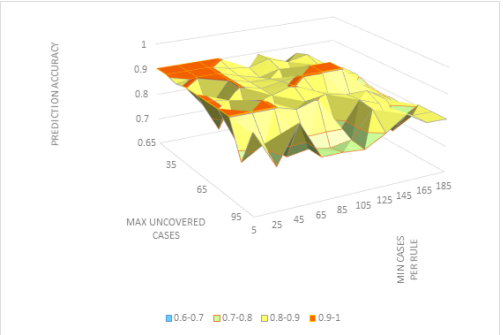
A comparison of the predictive accuracy of our distributed approach and the original Ant-Miner, as well as the top performing Ant-Miner approach, documented in the literature, is provided in table 11.3. The results indicate that Termid has prediction accuracy, not less than the original Ant-Miner approach, and in one case it even outruns the best performer. We can safely conclude that the distributed architecture of our proposal does not undermine the prediction accuracy of the algorithm.

11.5.2.1 Empirical Estimation of Parameters

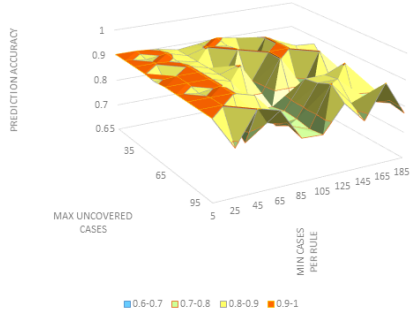
Figure 11.1 illustrates the variation of prediction accuracy with respect to the maximum uncovered cases and minimum one per rule parameters, for constant population of 5, 25, 75, 95 ants, against the tic-tac-toe dataset. Note, that throughout the literature the number of ants utillized for various experiments has usually a rather large value (e.g. 3,000). We argue that the distributed nature of our approach renders these values redundant.



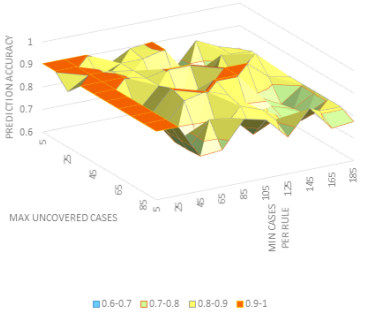
(A) With Constant Value of 5 Ants



(B) With Constant Value of 25 Ants



(C) With Constant Value of 75 Ants



(D) With Constant Value of 95 Ants

FIGURE 11.1: Variation of Prediction Accuracy According to Varying Maximum Uncovered Cases and Minimum Cases per Rule Parameters

Beginning from figure 11.1a we observe a variation of accuracy which only sporadically reaches its peak, which indicates that this small number of ants is not sufficient for generating good quality rules. As the number of ants increases the Max Uncovered Cases parameter becomes progressively more irrelevant while the Min Cases per Rule parameter steadily traps high accuracy rules between a space of 5 to 25 values.

11.5.3 Accuracy Against AWID

We moved on to conducting a more focused chain of experiments against the AWID-CLS-R-Trn and AWID-CLS-R-Tst versions of the AWID dataset. These datasets had gone through the process of feature reduction according to the conclusions extracted in

section 10.2. The reduced sets contain 20 attributes with the former being comprised of 1,795,575 data instances, unequally divided into 4 classes (1 normal, 3 intrusive), and the latter containing 575,643 instances. A more detailed description of the datasets characteristics is provided in table 9.4.

11.5.3.1 Predictive Accuracy

Both the environment and the ant nodes were deployed as different threads of execution operating on the same machine. Threading is a means of concurrency that greatly resembles the distributed nature of a realistic distributed deployment, thus reducing the testing costs for the purposes of initial experimentation. A comparison in matters of accuracy can be seen in table 11.4.

11.5.3.2 Training Speed

Although one of the most importance inefficiencies of Ant-Miner algorithm is its training speed, we managed to reduce the average training time, by splitting the burden to multiple ants operating on different nodes. For a system consisting of 10 ants, we were able to obtain training times that outperformed the best of the conventional methods (Naive Bayes).

11.5.4 Profiling the Algorithm Procedures

Towards evaluating the performance of our proposed approach in detail, we distinguished the 5 most important procedures executed by the nodes: (a) Rule Construction, (b) Rule pruning, (c) Pheromone updating, (d) Deleting instances, (e) Updating Entropy and performed profiling by instrumenting the binary executables with a custom tool. The statistical average over 100 executions of the time consumption of the corresponding functions in the source code was taken. The results can be seen in figure 11.2.

From the experimental results we can notice that the 3 most demanding processes are deleting the instances from the dataset, rule construction and rule pruning. Interestingly, while removing instances from the dataset is by far the most demanding process in the initial iterations of the processes, this situation changes for the iterations that follow, as the dataset shrinks. Thus, in the later iterations, the dominant processes in matters of time consumption are, rule construction and rule pruning.

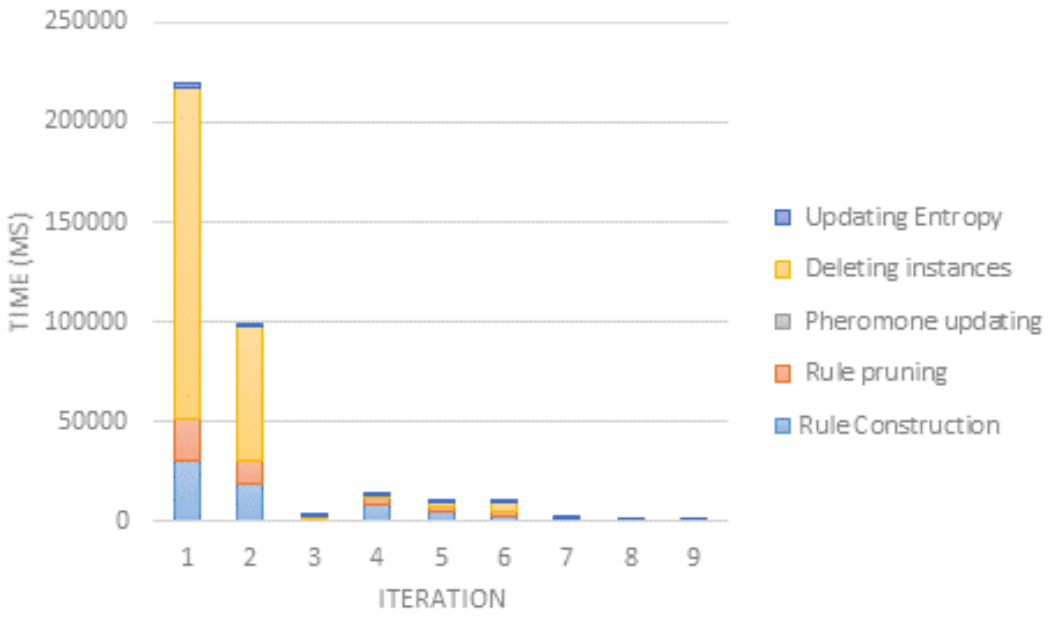


FIGURE 11.2: Profiling of Basic Operations

11.5.4.1 Simplicity of Rules

The result of the training phase produces a model comprised of 8.75 (about 8 to 9) rules in average and 6.7 (about 6 to 7) terms per rule. This result influences the testing phase, since each term per rule has to be compared against each instance in the test set. We underline the simplicity of the generated rules, especially when compared to other approaches like random tree, where the produced tree size surpass 500 nodes in some cases.

Dataset	Ant-Miner	Ant-Miner2	Ant-Miner3	ACO-Miner	Unordered	TACO-Miner	Ant-Miner+	cAnt-Miner	$\mu\psi$Ant-Miner	mAnt-Miner+	Ant-Miner-C	C.45
Breast Cancer (L)	75.28	75.91	78.39	77.14	78.42	77.76	77.47	-	-	74.79	-	-
Breast Cancer (W)	95.04	91.54	90.92	97.15	92.38	97.98	96.4	95.57	94.51	-	97.54	94.84
Heart Disease (C)	57.48	-	-	78.28	64.84	-	99.75	79.27	57.82	-	77.78	78.43
Dermatology	94.29	-	-	96.57	80.5	97.26	-	-	97.4	-	-	-
Hepatitis	90	-	-	94.63	95.42	95.5	-	84.89	-	-	87.33	68.25
Tic-Tac-Toe	73.04	71.13	68.94	98.43	72.45	71.87	-	-	98.76	75.57	100	94.03
Credit Card (Au)	86.09	84.3	83.61	-	-	-	84.05	86.6	82.92	-	89.42	81.99
Wine	90	85.33	83.5	-	-	-	94.59	95.14	95.22	-	98.24	96.6
Iris	95.33	81.8	77	-	-	-	94.51	-	-	-	97.33	94

TABLE 11.1: Comparison of Accuracy of Ant-Miner Approaches Against Several Public Datasets

Dataset	Records	Attributes	Classes
Tic-Tac-Toe	958	9	2
Breast Cancer (L)	286	9	2
Car	1728	6	4
Vote	435	16	2

TABLE 11.2: Characteristics of Toy Datasets

Dataset	Termid	Ant-Miner	Best Performer
Tic-Tac-Toe	.844	.730	1
Breast Cancer (L)	.823	.752	.784
Car	.848	.849	.980
Vote	.949	.949	.956

TABLE 11.3: Prediction Accuracy on Toy Datasets

TABLE 11.4: Accuracy of Termid Compared to Other Algorithms on the 20 Feature Set (%)

Our Proposal	AdaBoost	J48	Naive Bayes	OneR	Random Forest	Random Tree
95.22	92.2073	96.2574	90.5504	94.5741	95.8247	96.2258

TABLE 11.5: Time requirements of Termid Compared to Other Algorithms on the 20 Feature Set (in secs)

Termid	J48	Naive Bayes	Random Forest	Random Tree
25.91	568.92	29.67	739.78	49.3

Chapter 12

Conclusion and Future Directions

12.1 Conclusions

Based on the study conducted in the previous chapters, in this section we enumerate the basic conclusions that can be extracted from this PhD thesis.

12.1.1 The Nature of Wireless Attacks

Despite the revolutionary features of wireless technologies it is proved in practice that the end-user has to settle with more serious security risks. From our study (mainly chapter in 2) it becomes obvious that even the most recent and theoretically improved, amendments of virtually all wireless technologies, suffer from severe security inefficiencies which make them susceptible to a number of attacks. Judging from the amount of documented attacks (reviewed in chapters 3, 4, 5) it is proved that in some cases the methodology for executing such assaults is so trivial and the tools for orchestrating them are so widely available, that even low skilled ill motivated entities can unleash their attacks.

At the same time, the wireless medium has not only acted like a totally new playground for attackers and armed them with new potentials, but its idiosyncrasies rendered the existing intrusion detection and prevention mechanisms designed for the wired realm, inappropriate.

From an execution-method point of view, one can derive the common denominator of wireless attacks. More specifically, we claim that practically all MAC layer attacks are executed by either (a) taking advantage of unprotected packets to manipulate the management of the network, (b) introducing a rogue base station on the proximity of

a real one, and having it replicate its signalling behaviour to trick the end-users, or (c) inject encrypted packets of a specific structure to confuse the network.

Unlike the most well-known taxonomies in literature a taxonomy based on the execution method can be proposed as following:

- *Flooding* - Attacks capitalizing on the transmission of a large number of unencrypted frames, such as [221], 4.3.1 or 3.0.5.1.
- *Injection* - Attacks which are based on the injection of custom-shaped encrypted frames.
- *Impersonation* - Attacks powered by the introduction of a BS-like equipment on the network that claims it actually is the legit one.

12.1.2 Studying Wireless Attacks

For every study in network intrusion detection a realistic, representative dataset is considered a necessary asset. Nevertheless, according to our study (conducted in chapter 9) the available choices for a trustworthy testbed of such kind, are not only limited but also of poor quality. Partially, due to the sensitivity of their content and in part because of the difficulty of collecting and labelling such databases, the publicly available choices are narrowed down to datasets whose contents are artificial, tampered, ones that do not correspond to realistic conditions, or ones that are obsolete. Surprisingly, this is the case with the most popular dataset in intrusion detection, the KDD'99, which ends up producing highly questionable results when used as benchmark for intrusion detection.

We formulated the most desirable characteristics of a contemporary dataset for intrusion detection as:

- *Area Specific* - To apply to specific domains of interest.
- *Realistic Traffic* - To effectively describe the normal and anomalous conditions on a studied system, including the traces of the attacks.
- *Labelled* - To accommodate tagged records.
- *Integrity* - To contain complete records.
- *Satisfactory Size* - To be large enough in order to not undermine the accuracy of the detection process.
- *Variation* - To include a wide range of attacks as well as their modifications.

- *Freshness* - To be subject to updates.

A custom tailored dataset was introduced and by extensive analysis of this dataset we were able to realize that tree construction algorithms have an inherent advantage in intrusion detection. Nevertheless, we should underline that (a) the traditional machine learning algorithms disappoint when it comes to adequately detecting known and more importantly unknown attacks, and (b) most of these algorithms have extreme time and memory requirements for the training process.

12.1.3 Intrusion Detection with Bio-Inspired Algorithms

Having a mechanism for detecting the occurrence of attacks and attempts of misuse, before they prove disastrous is highly desirable for any type of computer network, wired, or wireless. However, the problem of intrusion detection and more specifically, network intrusion detection, is not a trivial one. Anomaly-based IDS, which rely on some type of soft computing technique, are promising for detection of unknown attacks. However, the application of such systems in real-life systems is scarce and limited. Despite the significant research in the field, the majority of such systems suffer from:

- Inaccurate predictions percentages.
- High false positive rates.
- Poor escalation behaviour as the dataset size increases.
- Low training speed.
- Incomprehensible (to humans) decision models.

It is without a doubt that the methods nature choose for solving complex problem, have always been a source of inspiration to scientists. Focusing on intrusion detection the problem is to define which portions of data from a given dataset represent normal behaviour and which ones lay outside these boundaries (thus consisting anomalies). According to the comprehensive review presented in 8, nature-inspired/bio-inspired techniques have been developed and successfully applied to the problem of intrusion detection. The main reasons for such an undertaking is their potential supremacy due to their following attributes:

- Ability to provide more accurate results by overcoming local minima/maxima.
- Self-adapting character.

- Highly parallel nature.

However, the problem of intrusion detection expands and differentiates, when applied to wireless networks. One the one hand, the processing of extremely large chunks of traces/data is required, and on the other, in such cases it is necessary to cope with a highly dynamic sense of normality.

While the theoretic advantages of nature-inspired approaches are clear, the implementation of a real-life system, is needed to experimentally verify this hypothesis. To this end, we developed Termid, a distributed IDS which conducts the detection process based on the Ant-Miner rule induction algorithm. We experimentally proved that such nature-inspired algorithms are easily parallelisable and at the same time they can be equally competitive with the legacy ones.

12.2 Thesis Contributions

The primary contribution of this thesis is the creation and public offering of a dataset specifically targeting wireless network intrusion detection. This dataset contains both normal and intrusive traffic, has enough size to fully describe all its classes, contains realistic traffic from a real network, and its attacks are distributed according to estimations of their severity. Generally, it conforms with the 7 principles a contemporary dataset must follow as stated in section 9. To the best of our knowledge it is the only dataset for wireless networks relevant to intrusion detection. For more details the interested reader should refer to [222] as well as section 9.

Another notable contribution of this work is the implementation of a distributed execution strategy of the pre-existing Ant-Miner classification rules extracting algorithm. The proposed algorithm applies an ant colony optimization heuristic to the classification task in order to extract $If \ldots Then$ rules. After being modified accordingly, it is deployed in a multicore/cluster environment, which provably boost its speed and efficiency. The algorithm becomes able to detect intrusions in large datasets, such as the ones obtained in wireless networks. The work in [223] along with section 11 contains the details of this undertake.

A contribution not to be neglected lays in the comprehensive survey of the known attacks for three of the most popular wireless technologies namely, the IEEE 802.11, IEEE 802.16 the UMTS technologies. Works [222], [47] and [224], [225] elaborate on this topic.

An exhaustive review of swarm intelligence methods in the field of intrusion detection is the final contribution of this thesis. This is included in its full extend in [226] and analysed here in chapter 8.

Table 12.1 provides global map of all the accomplishments relevant to this PhD thesis with respect to contributions in literature for quick reference.

TABLE 12.1: Overall PhD Thesis Contribution

Chapter	Contribution	Publication
2,3,4,5	Overview & Assessment of Wireless Threats	[224], [225], [47], [222]
8	A Dataset for Wireless Intrusion Detection (AWID)	[222]
7	Review of Nature Inspired Intrusion Detection Systems	[226]
10	A Distributed Ant Colony IDS (Termid)	[223]

12.3 Future Research Directions

The PhD thesis at hand has mainly contributed to the domain of intrusion detection in wireless networks with a new dataset and a fast and efficient algorithm/architecture based on nature-inspired algorithms for conducting intrusion detection. Yet, a number of areas is still left unexplored, and further work could be conducted. In this subsection, we elaborate on these possible directions.

- *Review of possible vulnerabilities of LTE* - Long Term Evolution (LTE) is a wireless standard based on the GSM/EDGE and UMTS network technologies, but increases their capacity and speed using a different radio interface along with other core network improvements. An in depth review of its security mechanisms and a presentation of the attacks that apply to it, should be beneficial for the research community due to its galloping adoption.

- *Expanding AWID with traces from different types of technologies* - AWID is a dataset containing real traces extracted from a working 802.11, WEP protected network. One can argue that this security mechanism is obsolete and the SOHO setting of the monitored network may not adequately apply to large scale settings. Moving towards the industry oriented networks seems to be the next natural transition for the AWID project.

- *Exploring alternative parallelization strategies* - Fast training and instant classification is important for in network intrusion detection. However, the extremely large volume of data (and signalling overhead) makes timely detection a troublesome task. Nature-inspired algorithms are in general greedy ones, requiring large amounts of time for their training. Still, interesting alternative parallelization

techniques such as Map Reduce or GPU-based acceleration approaches are left unexplored.

- *Efficient Discretization Technique* - Ant-Miner (and as a consequence Termid) has the ability of working with nominal values only. Several works have been published, that extend Ant-Miner's capacity by making it able to cope with continuous values, most notably [215]. Unfortunately, all these strategies seem to have high memory and CPU requirements. Applying such discretization techniques in big data scenarios, greatly undermines the potential gains of the algorithm from paralellization. A more efficient discretization technique should be considered as a high priority.

- *Integration of Termid with other approaches* - Termid surpassed the conventional approaches in detecting anomalous traffic. However, it did not perform adequately in detecting the normal class or the attacks for which it wasn't trained for. Such as potential lies in the integration of Termid with other approaches to increase its accuracy.

- *Multilevel Experiments On a Cloud Platform* - In this thesis the set of experiments was conducted on a single physical machine in which multiple virtual machines were spawned 11.5. While this approach has been particularly helpful for the initial phase of our studies, more sophisticated experiments could be completed in cloud platforms (like Amazon's EC2), with different number of virtual machines and characteristics, towards estimating Termid's scaling factor. By doing so, the network communication times will also be taken into account, thus offering much more complete results.

Appendix A

Record Fields of the AWID Dataset

Field Name	Type
frame_interface_id	Numeric
frame_dlt	Numeric
frame_offset_shift	Numeric
frame_time_epoch	Numeric
frame_time_delta	Numeric
frame_time_delta_displayed	Numeric
frame_time_relative	Numeric
frame_number	Numeric
frame_len	Numeric
frame_cap_len	Numeric
frame_marked	Nominal
frame_ignored	Nominal
radiotap_version	Numeric
radiotap_pad	Numeric
radiotap_length	Numeric
radiotap_present_tsft	Nominal
radiotap_present_flags	Nominal
radiotap_present_rate	Nominal
radiotap_present_channel	Nominal
radiotap_present_fhss	Nominal
radiotap_present_dbm_antsignal	Nominal
radiotap_present_dbm_antnoise	Nominal
radiotap_present_lock_quality	Nominal

radiotap_present_tx_attenuation	Nominal
radiotap_present_db_tx_attenuation	Nominal
radiotap_present_dbm_tx_power	Nominal
radiotap_present_antenna	Nominal
radiotap_present_db_antsignal	Nominal
radiotap_present_db_antnoise	Nominal
radiotap_present_rxflags	Nominal
radiotap_present_xchannel	Nominal
radiotap_present_mcs	Nominal
radiotap_present_ampdu	Nominal
radiotap_present_vht	Nominal
radiotap_present_reserved	Nominal
radiotap_present_rtap_ns	Nominal
radiotap_present_vendor_ns	Nominal
radiotap_present_ext	Nominal
radiotap_mactime	Numeric
radiotap_flags_cfp	Nominal
radiotap_flags_preamble	Nominal
radiotap_flags_wep	Nominal
radiotap_flags_frag	Nominal
radiotap_flags_fcs	Nominal
radiotap_flags_datapad	Nominal
radiotap_flags_badfcs	Nominal
radiotap_flags_shortgi	Nominal
radiotap_datarate	Numeric
radiotap_channel_freq	Numeric
radiotap_channel_type_turbo	Nominal
radiotap_channel_type_cck	Nominal
radiotap_channel_type_ofdm	Nominal
radiotap_channel_type_2ghz	Nominal
radiotap_channel_type_5ghz	Nominal
radiotap_channel_type_passive	Nominal
radiotap_channel_type_dynamic	Nominal
radiotap_channel_type_gfsk	Nominal
radiotap_channel_type_gsm	Nominal
radiotap_channel_type_sturbo	Nominal
radiotap_channel_type_half	Nominal
radiotap_channel_type_quarter	Nominal
radiotap_dbm_antsignal	Numeric

radiotap_antenna	Numeric
radiotap_rxflags_badplcp	Nominal
wlan_fc_type_subtype	Numeric
wlan_fc_version	Numeric
wlan_fc_type	Numeric
wlan_fc_subtype	Numeric
wlan_fc_ds	Numeric
wlan_fc_frag	Nominal
wlan_fc_retry	Nominal
wlan_fc_pwrmgt	Nominal
wlan_fc_moredata	Nominal
wlan_fc_protected	Nominal
wlan_fc_order	Nominal
wlan_duration	Numeric
wlan_ra	Numeric
wlan_da	Numeric
wlan_ta	Numeric
wlan_sa	Numeric
wlan_bssid	Numeric
wlan_frag	Numeric
wlan_seq	Numeric
wlan_bar_type	Numeric
wlan_ba_control_ackpolicy	Nominal
wlan_ba_control_multitid	Nominal
wlan_ba_control_cbitmap	Nominal
wlan_bar_compressed_tidinfo	Nominal
wlan_ba_bm	Numeric
wlan_fcs_good	Nominal
wlan_mgt_fixed_capabilities_ess	Nominal
wlan_mgt_fixed_capabilities_ibss	Nominal
wlan_mgt_fixed_capabilities_cfpoll_ap	Numeric
wlan_mgt_fixed_capabilities_privacy	Nominal
wlan_mgt_fixed_capabilities_preamble	Nominal
wlan_mgt_fixed_capabilities_pbcc	Nominal
wlan_mgt_fixed_capabilities_agility	Nominal
wlan_mgt_fixed_capabilities_spec_man	Nominal
wlan_mgt_fixed_capabilities_short_slot_time	Nominal
wlan_mgt_fixed_capabilities_apsd	Nominal
wlan_mgt_fixed_capabilities_radio_measurement	Nominal

wlan_mgt_fixed_capabilities_dsss_ofdm	Nominal
wlan_mgt_fixed_capabilities_del_blk_ack	Nominal
wlan_mgt_fixed_capabilities_imm_blk_ack	Nominal
wlan_mgt_fixed_listen_ival	Numeric
wlan_mgt_fixed_current_ap	Numeric
wlan_mgt_fixed_status_code	Numeric
wlan_mgt_fixed_timestamp	Numeric
wlan_mgt_fixed_beacon	Numeric
wlan_mgt_fixed_aid	Numeric
wlan_mgt_fixed_reason_code	Numeric
wlan_mgt_fixed_auth_alg	Numeric
wlan_mgt_fixed_auth_seq	Numeric
wlan_mgt_fixed_category_code	Numeric
wlan_mgt_fixed_htact	Numeric
wlan_mgt_fixed_chanwidth	Numeric
wlan_mgt_fixed_fragment	Numeric
wlan_mgt_fixed_sequence	Numeric
wlan_mgt_tagged_all	Numeric
wlan_mgt_ssid	Nominal
wlan_mgt_ds_current_channel	Numeric
wlan_mgt_tim_dtim_count	Numeric
wlan_mgt_tim_dtim_period	Numeric
wlan_mgt_tim_bmapctl_multicast	Nominal
wlan_mgt_tim_bmapctl_offset	Numeric
wlan_mgt_country_info_environment	Numeric
wlan_mgt_rsn_version	Numeric
wlan_mgt_rsn_gcs_type	Numeric
wlan_mgt_rsn_pcs_count	Numeric
wlan_mgt_rsn_akms_count	Numeric
wlan_mgt_rsn_akms_type	Numeric
wlan_mgt_rsn_capabilities_preauth	Nominal
wlan_mgt_rsn_capabilities_no_pairwise	Nominal
wlan_mgt_rsn_capabilities_ptksa_replay_counter	Numeric
wlan_mgt_rsn_capabilities_gtksa_replay_counter	Numeric
wlan_mgt_rsn_capabilities_mfpr	Nominal
wlan_mgt_rsn_capabilities_mfpc	Nominal
wlan_mgt_rsn_capabilities_peerkey	Nominal
wlan_mgt_tcprep_trsmt_pow	Numeric
wlan_mgt_tcprep_link_mrg	Numeric

wlan_wep_iv	Numeric
wlan_wep_key	Numeric
wlan_wep_icv	Numeric
wlan_tkip_extiv	Numeric
wlan_ccmp_extiv	Numeric
wlan_qos_tid	Numeric
wlan_qos_priority	Numeric
wlan_qos_eosp	Numeric
wlan_qos_ack	Numeric
wlan_qos_amsdupresent	Numeric
wlan_qos_buf_state_indicated	Numeric
wlan_qos_bit4	Numeric
wlan_qos_txop_dur_req	Numeric
wlan_qos_ps_buf_state	Numeric
data_len	Numeric
label	Nominal

Appendix B

Swarm Intelligence Algorithms Used in Intrusion Detection

Algorithm 4 Pseudocode for Standard Particle Swarm Optimization-Algorithm

```
1: Initialize Swarm( )
2: repeat
3:     for ρ = 0 to P do
4:         evaluate( ρ )
5:         updatePersonalBest(ρ)
6:         updateGlobalBest(ρ)
7:     end for
8:     for ≒0 to D do
9:         updateVelocity(d)
10:        updatePosition(d)
11:    end for
12: until StoppingCriterion == true
```

Algorithm 5 Pseudocode for Basic Ant Colony Clustering Algorithm

```
1: scatterDataOnGrid( Dataset )
2: placeAntsOnGridRandomly( )
3: for ρ = 1 to MaxIterations do
4:     MoveAnt( j, StepSize )
5:     if (antCarriesItem() == true) AND (currentPositionHasItem() ==
   false) then
6:         dropItem( )
7:     end if
8:     if (antCarriesItem() == false) AND (currentPositionHasItem() ==
   true) then
9:         pickItem( )
10:    end if
11:    updateGlobalBest( )
12: end for
```

Appendix C

Formal Definition of Threat in 802.16

Likelihood $L : f_L = (x, y, z)$where$x \in C, y \in D, z \in 2R$

$$Likely = \begin{cases} f_L = (E_x, E_a, L_o) \\ f_L = (M_a, E_a, L_o) \\ f_L = (M_a, E_a, M_o) \\ f_L = (I_n, E_a, L_o) \\ f_L = (I_n, S_o, L_o) \\ f_L = (I_n, E_a, M_o) \end{cases} \tag{C.1}$$

$$Possible = \begin{cases} f_L = (Ex, So, Lo) \\ f_L = (Ex, Ea, Mo) \\ f_L = (Ex, So, Mo) \\ f_L = (Ex, Ea, Hi) \\ f_L = (Ex, So, Hi) \\ f_L = (Ma, So, Lo) \\ f_L = (Ma, So, Mo) \\ f_L = (Ma, Ea, Hi) \\ f_L = (Ma, So, Hi) \\ f_L = (In, So, Mo) \\ f_L = (In, Ea, Hi) \\ f_L = (In, So, Hi) \end{cases} \tag{C.2}$$

$$Unlikely = \begin{cases} f_L = (Ex, Ha, Lo) \\ f_L = (Ex, Ha, Mo) \\ f_L = (Ex, Ha, Hi) \\ f_L = (Ma, Ha, Lo) \\ f_L = (Ma, Ha, Mo) \\ f_L = (Ma, Ha, Hi) \\ f_L = (In, Ha, Lo) \\ f_L = (In, Ha, Mo) \\ f_L = (In, Ha, Hi) \end{cases} \tag{C.3}$$

Impact $I : f_I = (k, l, m)$where$k \in S, l \in T, m \in OR$

$$Low = \begin{cases} f_I = (Sm, Sh, An) \\ f_I = (Sm, Ln, An) \\ f_I = (Sm, Sh, DoS) \\ f_I = (Me, Sh, An) \\ f_I = (Me, Ln, An) \\ f_I = (Me, Sh, DoS) \end{cases} \tag{C.4}$$

$$Medium = \begin{cases} f_I = (Sm, Sh, An) \\ f_I = (Sm, Ln, An) \\ f_I = (Sm, Sh, DoS) \\ f_I = (Me, Sh, An) \\ f_I = (Me, Ln, An) \\ f_I = (Me, Sh, DoS) \end{cases} \tag{C.5}$$

$$Medium = \begin{cases} fI = (Sm, Sh, ToS) \\ fI = (Sm, Ln, ToS) \\ fI = (Sm, Sh, LoP) \\ fI = (Sm, Ln, LoP) \\ fI = (Me, Sh, ToS) \\ fI = (Me, Ln, ToS) \\ fI = (Me, Sh, LoP) \\ fI = (Me, Ln, LoP) \\ fI = (La, Ln, DoS) \\ fI = (La, Sh, ToS) \\ fI = (La, Ln, LoP) \\ fI = (La, Sh, LoP) \\ fI = (Ka, Sh, LoP) \\ fI = (Me, Ln, LoP) \end{cases} \tag{C.6}$$

Threat $T : f_T = f_L + f_I$

$$Major = \begin{cases} f_I = High \wedge f_L = Likely \\ f_I = High \wedge f_L = Possible \end{cases} \tag{C.7}$$

$$Moderate = \begin{cases} f_I = Medium \wedge f_L = Possible \\ f_I = Medium \wedge f_L = Likely \end{cases} \tag{C.8}$$

$$Minor = \begin{cases} f_I = Low \wedge f_L = Likely \\ f_I = Low \wedge f_L = Possible \\ f_I = Low \wedge f_L = Unlikely \\ f_I = Medium \wedge f_L = Unlikely \\ f_I = High \wedge f_L = Unlikely \end{cases} \tag{C.9}$$

Bibliography

[1] Bode Karl. Wireless traffic to reach 11.2 exabytes a month @ONLINE, June 2014. URL http://www.dslreports.com/shownews/Cisco-Wireless-Traffic-to-Reach-112-Exabytes-a-Month-By-2017-123040.

[2] Osborne Mark. Widz - the wireless intrusion detection system @ONLINE, June 2014. URL http://www.loud-fat-bloke.co.uk/tools/widzv1.5.zip.

[3] Cisco. Adaptive wireless ips software @ONLINE, June 2014. URL http://www.cisco.com/c/en/us/products/collateral/wireless/adaptive-wireless-ips-software/data_sheet_c78-501388.html.

[4] Fluke Networks. Airmagnet wifi analyzer @ONLINE, June 2014. URL http://www.flukenetworks.com/enterprise-network/wireless-network/AirMagnet-WiFi-Analyzer.

[5] Motorola. Airmagnet wifi analyzer @ONLINE, June 2014. URL http://www.motorolasolutions.com/XL-EN/Business+Products+and+Services/Software+and+Applications/WLAN+Management+and+Security+Software/AirDefense-Security-and-Compliance.

[6] Alexandros Tsakountakis, Georgios Kambourakis, and Stefanos Gritzalis. Towards effective wireless intrusion detection in ieee 802.11 i. In *Security, Privacy and Trust in Pervasive and Ubiquitous Computing, 2007. SECPerU 2007. Third International Workshop on*, pages 37–42. IEEE, 2007.

[7] James P Anderson. Computer security threat monitoring and surveillance. Technical report, Technical report, James P. Anderson Company, Fort Washington, Pennsylvania, 1980.

[8] Dimitrios Damopoulos, Sofia A Menesidou, Georgios Kambourakis, Maria Papadaki, Nathan Clarke, and Stefanos Gritzalis. Evaluation of anomaly-based ids for mobile devices using machine learning classifiers. *Security and Communication Networks*, 5(1):3–14, 2012.

[9] Dimitrios Damopoulos, Georgios Kambourakis, and Georgios Portokalidis. The best of both worlds: a framework for the synergistic operation of host and cloud anomaly-based ids for smartphones. In *Proceedings of the Seventh European Workshop on System Security*, page 6. ACM, 2014.

[10] IEEE. 802.11-1997 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. @ONLINE, June 2014. URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=654749.

[11] IEEE. 802.11w-2009 - 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 amendment 4: Protected management frames. @ONLINE, June 2014. URL http://standards.ieee.org/findstds/standard/802.11w-2009.html.

[12] Benjamin Bertka. 802.11 w security: Dos attacks and vulnerability controls. In *Proc. of Infocom*, 2012.

[13] Bastian Könings. Evaluation of the 802.11 w amendment for existing dos attacks.

[14] Weijia Wang and Haihang Wang. Weakness in 802.11 w and an improved mechanism on protection of management frame. In *Wireless Communications and Signal Processing (WCSP), 2011 International Conference on*, pages 1–4. IEEE, 2011.

[15] Md Sohail Ahmad and Shashank Tadakamadla. Short paper: security evaluation of ieee 802.11 w specification. In *Proceedings of the fourth ACM conference on Wireless network security*, pages 53–58. ACM, 2011.

[16] Martin Eian and SF Mjolsnes. A formal analysis of ieee 802.11 w deadlock vulnerabilities. In *INFOCOM, 2012 Proceedings IEEE*, pages 918–926. IEEE, 2012.

[17] ETSI. Collection of rc4 analysis @ONLINE, June 2014. URL http://www.wisdom.weizmann.ac.il/~itsik/RC4/rc4.html.

[18] Henry S Warren. *Hacker's delight*. Addison-Wesley Professional, 2003.

[19] Cablelabs. Data-over-cable service interface specifications, baseline privacy plus interface specification @ONLINE, June 2014. URL http://www.cablelabs.com/specifications/CM-SP-BPI+-C01-081104.pdf.

[20] IEEE. 802.16-2001, i.s. ieee standard for local and metropolitan area networks part 16: Air interface for fixed broadband wireless access systems @ONLINE, June 2014. URL http://standards.ieee.org/getieee802/download/802.16-2001.pdf.

[21] IEEE. 802.16-2004, i.s. ieee standard for local and metropolitan area networks part 16: Air interface for fixed broadband wireless access systems @ONLINE, June 2014. URL http://standards.ieee.org/getieee802/download/802.16-2004.pdf.

[22] IEEE. Ieee standard for local and metropolitan area networks part 16: Air interface for fixed broadband wireless access systems @ONLINE, June 2014. URL http://standards.ieee.org/getieee802/download/802.16e-2005.pdf.

[23] IEEE. Ieee standard for local and metropolitan area networks part 16: Air interface for fixed broadband wireless access systems @ONLINE, June 2014. URL http://standards.ieee.org/getieee802/download/802.16j-2009.pdf.

[24] IEEE. Ieee standard for local and metropolitan area networks part 16: Air interface for fixed broadband wireless access systems @ONLINE, June 2014. URL http://standards.ieee.org/findstds/standard/802.16m-2011.html.

[25] 3GPP. Technical specification, 3gpp, 3g security; security architecture (release 8), ts 33.102 v8.3.0 @ONLINE, June 2014. URL http://www.etsi.org/deliver/etsi_ts/129200_129299/129280/08.03.00_60/ts_129280v080300p.pdf.

[26] ETSI. Kazumi algorithm specification, etsi ts 135 202 v7.0.0 @ONLINE, June 2014. URL http://www.etsi.org/website/document/algorithms/ts_135202v070000p.pdf.

[27] ISO/IEC. Information technology; security techniques; entity authentication part 4: Mechanisms using a cryptographic check function @ONLINE, June 2014. URL http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=31488.

[28] Wenyuan Xu, Wade Trappe, Yanyong Zhang, and Timothy Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 46–57. ACM, 2005.

[29] Wenyuan Xu, Ke Ma, Wade Trappe, and Yanyong Zhang. Jamming sensor networks: attack and defense strategies. *Network, IEEE*, 20(3):41–47, 2006.

[30] Daniel C Nash, Thomas L Martin, Dong S Ha, and Michael S Hsiao. Towards an intrusion detection system for battery exhaustion attacks on mobile computing devices. In *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, pages 141–145. IEEE, 2005.

[31] Yihong Zhou, Dapeng Wu, and Scott M Nettles. Analyzing and preventing mac-layer denial of service attacks for stock 802.11 systems. In *Workshop on BWSA, Broadnets*, 2004.

[32] Sanjay Ramaswamy, Huirong Fu, Manohar Sreekantaradhya, John Dixon, and Kendall E Nygard. Prevention of cooperative black hole attack in wireless ad hoc networks. In *International Conference on Wireless Networks*, volume 2003, 2003.

[33] Mohammad Al-Shurman, Seong-Moo Yoo, and Seungjin Park. Black hole attack in mobile ad hoc networks. In *Proceedings of the 42nd annual Southeast regional conference*, pages 96–97. ACM, 2004.

[34] Anthony Wood and John A Stankovic. Denial of service in sensor networks. *Computer*, 35(10):54–62, 2002.

[35] Haining Wang, Danlu Zhang, and Kang G Shin. Detecting syn flooding attacks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1530–1539. IEEE, 2002.

[36] Scott Fluhrer, Itsik Mantin, and Adi Shamir. Weaknesses in the key scheduling algorithm of rc4. In *Selected areas in cryptography*, pages 1–24. Springer, 2001.

[37] Rafik Chaabouni et al. Break wep faster with statistical analysis. Technical report, technical report, EPFL, LASEC, 2006.

[38] Erik Tews, Ralf-Philipp Weinmann, and Andrei Pyshkin. Breaking 104 bit wep in less than 60 seconds. In *Information Security Applications*, pages 188–202. Springer, 2007.

[39] Andreas Klein. Attacks on the rc4 stream cipher. *Designs, Codes and Cryptography*, 48(3):269–286, 2008.

[40] KoreK. Chopchop (experimental wep attacks) @ONLINE, June 2014. URL http://www.netstumbler.org/showthread.php?t=12489.

[41] Md Sohail Ahmad and Vivek Ramachandran. Cafe latte with a free topping of cracked wep retrieving wep keys from road warriors, 2007.

[42] IEEE. Ieee standard for information technology- telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements. part 11: Wireles lan medium access control (mac) and physical layer (phy) specifications. amendment 5: Enhancements for higher throughput @ONLINE, June 2014. URL http://standards.ieee.org/getieee802/download/802.11n-2009.pdf.

[43] Kemal Bicakci and Bulent Tavli. Denial-of-service attacks and countermeasures in ieee 802.11 wireless networks. *Computer Standards & Interfaces*, 31(5):931–941, 2009.

[44] Wenjun Gu, Zhimin Yang, Can Que, Dong Xuan, and Weijia Jia. On security vulnerabilities of null data frames in ieee 802.11 based wlans. In *Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on*, pages 28–35. IEEE, 2008.

[45] Wenjun Gu, Zhimin Yang, Dong Xuan, Weijia Jia, and Can Que. Null data frame: A double-edged sword in ieee 802.11 wlans. *Parallel and Distributed Systems, IEEE Transactions on*, 21(7):897–910, 2010.

[46] Mina Malekzadeh, Abdul Ghani, Abdul Azim, Jalil Desa, and Shamala Subramaniam. Empirical analysis of virtual carrier sense flooding attacks over wireless local area network. *Journal of Computer science*, 5(3), 2009.

[47] Constantinos Kolias, Georgios Kambourakis, and Stefanos Gritzalis. Attacks and countermeasures on 802.16: Analysis and assessment. *Communications Surveys & Tutorials, IEEE*, 15(1):487–514, 2013.

[48] Taeshik Shon and Wook Choi. An analysis of mobile wimax security: vulnerabilities and solutions. In *Network-Based Information Systems*, pages 88–97. Springer, 2007.

[49] Bharat Bhargava, Yu Zhang, Nwokedi Idika, Leszek Lilien, and Mehdi Azarmi. Collaborative attacks in wimax networks. *Security and Communication Networks*, 2(5):373–391, 2009.

[50] Sheraz Naseer, Muhammad Younus, and Attiq Ahmed. Vulnerabilities exposing ieee 802.16 e networks to dos attacks: a survey. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD'08. Ninth ACIS International Conference on*, pages 344–349. IEEE, 2008.

[51] Maryam Shojaee, Naser Movahhedinia, and Behrouz Tork Ladani. Traffic analysis for wimax network under ddos attack. In *Circuits, Communications and System*

(PACCS), 2010 Second Pacific-Asia Conference on, volume 1, pages 279–283. IEEE, 2010.

[52] J Han, Mohamad Yusoff Alias, and Goi Bok Min. Potential denial of service attacks in ieee802. 16e-2005 networks. In *Communications and Information Technology, 2009. ISCIT 2009. 9th International Symposium on*, pages 1207–1212. IEEE, 2009.

[53] Frank A. Ibikunle. Security issues in mobile wimax (802.16e). In *Proceedings of the 2009 IEEE Mobile WiMAX Symposium*, MWS '09, pages 117–122. IEEE Computer Society, 2009. ISBN 978-0-7695-3719-1. doi: 10.1109/MWS.2009.50. URL http://dx.doi.org/10.1109/MWS.2009.50.

[54] Rodney Rambally and Vikas Solomon Abel. An analysis of wimax security vulnerabilities.

[55] Leonardo Maccari, Matteo Paoli, and Romano Fantacci. Security analysis of ieee 802.16. In *Communications, 2007. ICC'07. IEEE International Conference on*, pages 1160–1165. IEEE, 2007.

[56] Youngwook Kim, Hyoung-Kyu Lim, and Saewoong Bahk. Shared authentication information for preventing ddos attacks in mobile wimax networks. In *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pages 765–769. IEEE, 2008.

[57] Analysis of mobile wimax security: vulnerabilities and solutions.

[58] Andreas Deininger, Shinsaku Kiyomoto, Jun Kurihara, and Toshiaki Tanaka. Security vulnerabilities and solutions in mobile wimax. *IJCSNS International Journal of Computer Science and Network Security*, 7(11):7–15, 2007.

[59] Sen Xu and Chin-Tser Huang. Attacks on pkm protocols of ieee 802.16 and its later versions. In *Wireless Communication Systems, 2006. ISWCS'06. 3rd International Symposium on*, pages 185–189. IEEE, 2006.

[60] Ayesha Altaf, M Younus Javed, and Attiq Ahmed. Security enhancements for privacy and key management protocol in ieee 802.16 e-2005. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD'08. Ninth ACIS International Conference on*, pages 335–339. IEEE, 2008.

[61] Chin-Tser Huang and J Morris Chang. Responding to security issues in wimax networks. *IT Professional*, 10(5):15–21, 2008.

[62] Evren Eren. Wimax security architecture-analysis and assessment. In *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2007. IDAACS 2007. 4th IEEE Workshop on*, pages 673–677. IEEE, 2007.

[63] Shon Taeshik, Koo Bonhyun, Park Jong Hyuk, and Chang Hangbae. Novel approaches to enhance mobile wimax security. *EURASIP Journal on Wireless Communications and Networking*, 2010, 2010.

[64] William F Ehrsam, Carl HW Meyer, John L Smith, and Walter L Tuchman. Message verification and transmission error detection by block chaining, February 14 1978. US Patent 4,074,066.

[65] JESSE WALKER. Overview of ieee 802.16 security. 2004.

[66] Sami Vaarala, Antti Nuopponen, and Teemupekka Virtanen. Attacking predictable ipsec esp initialization vectors. In *Information and Communications Security*, pages 160–172. Springer, 2002.

[67] Christopher B McCubbin, Ali Aydin Selçuk, and Deepinder P Sidhu. Initialization vector attacks on the ipsec protocol suite. In *wetice*, pages 171–175, 2000.

[68] NIST. Data encryption standard (des) @ONLINE, June 2014. URL http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf.

[69] Georgios Kambourakis, Elisavet Konstantinou, and Stefanos Gritzalis. Revisiting wimax mbs security. *Computers & Mathematics with Applications*, 60(2):217–223, 2010.

[70] Sen Xu, Chin-Tser Huang, and Manton M Matthews. Secure multicast in wimax. *Journal of Networks*, 3(2), 2008.

[71] Muzammil Khan, Attiq Ahmed, and Ahmad Raza Cheema. Vulnerabilities of umts access domain security architecture. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD'08. Ninth ACIS International Conference on*, pages 350–355. IEEE, 2008.

[72] Patrick PC Lee, Tian Bu, and Thomas Woo. On the detection of signaling dos attacks on 3g/wimax wireless networks. *Computer Networks*, 53(15):2601–2616, 2009.

[73] Jari Arkko and Henry Haverinen. Extensible authentication protocol method for 3rd generation authentication and key agreement (eap-aka). 2006.

[74] Bernard Aboba, Larry Blunk, John Vollbrecht, James Carlson, Henrik Levkowetz, et al. Extensible authentication protocol (eap). Technical report, RFC 3748, June, 2004.

[75] Ulrike Meyer and Susanne Wetzel. A man-in-the-middle attack on umts. In *Proceedings of the 3rd ACM workshop on Wireless security*, pages 90–97. ACM, 2004.

[76] Ulrike Meyer and Susanne Wetzel. On the impact of gsm encryption and man-in-the-middle attacks on the security of interoperating gsm/umts networks. In *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, volume 4, pages 2876–2883. IEEE, 2004.

[77] Zahra Ahmadian, Somayeh Salimi, and Ahmad Salahi. New attacks on umts network access. In *Wireless Telecommunications Symposium, 2009. WTS 2009*, pages 1–6. IEEE, 2009.

[78] Elad Barkan, Eli Biham, and Nathan Keller. Instant ciphertext-only cryptanalysis of gsm encrypted communication. In *Advances in Cryptology-CRYPTO 2003*, pages 600–616. Springer, 2003.

[79] Aircrack-ng. Aircrack-ng, June 2014. URL http://www.aircrack-ng.org.

[80] MDK3. Mdk3, June 2014. URL http://homepages.tu-darmstadt.de/~pnlarbig/wlan/n#mdk3.

[81] File2air. File2air, June 2014. URL http://www.willhackforsushi.com/?pagenid=19.

[82] Lorcon-old. Lorcon-old, June 2014. URL https://aur.archlinux.org/packages/lorcon-old-git/?setlang=en.

[83] Lorcon2 Library. Lorcon2 library, June 2014. URL https://code.google.com/p/lorcon/.

[84] Michel Barbeau. Wimax/802.16 threat analysis. In *Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, pages 8–15. ACM, 2005.

[85] Krishnun Sansurooah. An assessment of threats of the physical and mac address layers in wimax/802.16. In *Australian Information Security Management Conference*, page 79, 2006.

[86] ETSI. Telecommunications and internet protocol harmonization over networks (tiphon) release 4; protocol framework definition; methods and protocols for security; part 1: Threat analysis, June 2014. URL http://docbox.etsi.org/EC_Files/EC_Files/ts_10216501v040101p.pdf.

[87] Kwanghun Han and Sunghyun Choi. Performance analysis of sleep mode operation in ieee 802.16 e mobile broadband wireless access systems. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, volume 3, pages 1141–1145. IEEE, 2006.

[88] Juan Deng, Richard R Brooks, and James Martin. Assessing the effect of wimax system parameter settings on mac-level local dos vulnerability. *International Journal of Performability Engineering*, 8(2):183, 2012.

[89] Huawei Technologies Co. Huawei wimax base station3703, June 2014. URL http://www.huawei.com/le/download.do?f=2682.

[90] John Bellardo and Stefan Savage. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *USENIX security*, pages 15–28, 2003.

[91] Henry S Teng, Kaihu Chen, and SC Lu. Adaptive real-time anomaly detection using inductively generated sequential patterns. In *Research in Security and Privacy, 1990. Proceedings., 1990 IEEE Computer Society Symposium on*, pages 278–284. IEEE, 1990.

[92] Osmar R Zaïane. Introduction to data mining. 1999.

[93] Paulo J Lisboa and Azzam FG Taktak. The use of artificial neural networks in decision support in cancer: a systematic review. *Neural networks*, 19(4):408–415, 2006.

[94] Maciej A Mazurowski, Piotr A Habas, Jacek M Zurada, Joseph Y Lo, Jay A Baker, and Georgia D Tourassi. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural networks*, 21(2):427–436, 2008.

[95] Elpiniki I Papageorgiou. A new methodology for decisions in medical informatics using fuzzy cognitive maps based on fuzzy rule-extraction techniques. *Applied Soft Computing*, 11(1):500–513, 2011.

[96] Zheng Rong Yang, Rebecca Thomson, Philip McNeil, and Robert M Esnouf. Ronn: the bio-basis function neural network technique applied to the detection of natively disordered regions in proteins. *Bioinformatics*, 21(16):3369–3376, 2005.

[97] Jian Liang and Ruxu Du. Model-based fault detection and diagnosis of hvac systems using support vector machine method. *International Journal of refrigeration*, 30(6):1104–1114, 2007.

[98] HP Ng, SH Ong, KWC Foong, PS Goh, and WL Nowinski. Medical image segmentation using k-means clustering and improved watershed algorithm. In *Image Analysis and Interpretation, 2006 IEEE Southwest Symposium on*, pages 61–65. IEEE, 2006.

[99] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):30–42, 2012.

[100] Eibe Frank and Remco R Bouckaert. Naive bayes for text classification with unbalanced classes. In *Knowledge Discovery in Databases: PKDD 2006*, pages 503–510. Springer, 2006.

[101] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2: 45–66, 2002.

[102] Srinivas Mukkamala, Guadalupe Janoski, and Andrew Sung. Intrusion detection using neural networks and support vector machines. In *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, volume 2, pages 1702–1707. IEEE, 2002.

[103] Nahla Ben Amor, Salem Benferhat, and Zied Elouedi. Naive bayes vs decision trees in intrusion detection systems. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 420–424. ACM, 2004.

[104] Leonid Portnoy. Intrusion detection with unlabeled data using clustering. 2000.

[105] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.

[106] Daniel Barbara, Ningning Wu, and Sushil Jajodia. Detecting novel network intrusions using bayes estimators. In *First SIAM Conference on Data Mining*. SIAM, 2001.

[107] Christopher Kruegel, Darren Mutz, William Robertson, and Fredrik Valeur. Bayesian event classification for intrusion detection. In *Computer Security Applications Conference, 2003. Proceedings. 19th Annual*, pages 14–23. IEEE, 2003.

[108] Mrutyunjaya Panda and Manas Ranjan Patra. Network intrusion detection using naive bayes. *International journal of computer science and network security*, 7 (12):258–263, 2007.

[109] Vladimir Vapnik. *The nature of statistical learning theory*. springer, 2000.

[110] Xian Rao, Chun-Xi Dong, and Shao-Quan Yang. An intrusion detection system based on support vector machine. *Journal of Software*, 4(14), 2003.

[111] J. R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, March 1986. ISSN 0885-6125.

[112] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. 1993. ISBN 1-55860-238-0.

[113] Xiao-Bai Li. A scalable decision tree system and its application in pattern recognition and intrusion detection. *Decision Support Systems*, 41(1):112–130, 2005.

[114] Ville Hautamäki, Ismo Kärkkäinen, and Pasi Fränti. Outlier detection using k-nearest neighbour graph. In *ICPR (3)*, pages 430–433, 2004.

[115] Yihua Liao and V Rao Vemuri. Use of k-nearest neighbor classifier for intrusion detection. *Computers & Security*, 21(5):439–448, 2002.

[116] Klaus Julisch. Clustering intrusion detection alarms to support root cause analysis. *ACM Transactions on Information and System Security (TISSEC)*, 6(4):443–471, 2003.

[117] Yu Guan, Ali-Akbar Ghorbani, and Nabil Belacel. Y-means: A clustering method for intrusion detection. 2003.

[118] Bovas Abraham and George EP Box. Bayesian analysis of some outlier problems in time series. *Biometrika*, 66(2):229–236, 1979.

[119] Gerardo Beni and Jing Wang. Swarm intelligence in cellular robotic systems. In Paolo Dario, Giulio Sandini, and Patrick Aebischer, editors, *Robots and Biological Systems: Towards a New Bionics?*, volume 102 of *NATO ASI Series*, pages 703–712. Springer Berlin Heidelberg, 1993. ISBN 978-3-642-63461-1. doi: 10.1007/978-3-642-58069-7_38. URL http://dx.doi.org/10.1007/978-3-642-58069-7_38.

[120] J.-L. Deneubourg, S. Aron, S. Goss, and J.M. Pasteels. The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior*, 3(2):159–168, 1990. ISSN 0892-7553. doi: 10.1007/BF01417909. URL http://dx.doi.org/10.1007/BF01417909.

[121] S. Goss, S. Aron, J.L. Deneubourg, and J.M. Pasteels. Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76(12):579–581, 1989. ISSN 0028-1042. doi: 10.1007/BF00462870. URL http://dx.doi.org/10.1007/BF00462870.

[122] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *Computational Intelligence Magazine, IEEE*, 1(4):28–39, 2006.

[123] Marco Dorigo and Gianni Di Caro. New ideas in optimization. chapter The Ant Colony Optimization Meta-heuristic, pages 11–32. 1999. ISBN 0-07-709506-5. URL http://dl.acm.org/citation.cfm?id=329055.329062.

[124] Emad Soroush, Mohammad Saniee Abadeh, and Jafar Habibi. A boosting ant-colony optimization algorithm for computer intrusion detection. In *Proceedings of the 2006 International Symposium on Frontiers in Networking with Applications (FINA 2006)*, 2006.

[125] Rafael S Parpinelli, Heitor S Lopes, and Alex Alves Freitas. Data mining with an ant colony optimization algorithm. *Evolutionary Computation, IEEE Transactions on*, 6(4):321–332, 2002.

[126] Junbing He and Dongyang Long. An improved ant-based classifier for intrusion detection. In *Natural Computation, 2007. ICNC 2007. Third International Conference on*, volume 4, pages 819–823. IEEE, 2007.

[127] Chandrasekar Ramachandran, Sudip Misra, and Mohammad S Obaidat. Fork: A novel two-pronged strategy for an agent-based intrusion detection scheme in ad-hoc networks. *Computer Communications*, 31(16):3855–3869, 2008.

[128] Mohammad Saniee Abadeh, Jafar Habibi, and Emad Soroush. Induction of fuzzy classification systems via evolutionary aco-based algorithms. *computer*, 35:37, 2008.

[129] Mohammad Saniee Abadeh, Jafar Habibi, M Daneshi, M Jalali, and M Khezrzadeh. Intrusion detection using a hybridization of evolutionary fuzzy systems and artificial immune systems. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 3547–3553. IEEE, 2007.

[130] H Alipour, E Khosrowshahi Asl, M Esmaeili, and M Nourhosseini. Aco-fcr: Applying aco-based algorithms to induct fcr. In *Proceedings of the World Congress on Engineering*, volume 1, 2008.

[131] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[132] Qiang Wang and Vasileios Megalooikonomou. A clustering algorithm for intrusion detection. In *Defense and Security*, pages 31–38. International Society for Optics and Photonics, 2005.

[133] Hisao Ishibuchi and T Nakaskima. Improving the performance of fuzzy classifier systems for pattern classification problems with continuous attributes. *Industrial Electronics, IEEE Transactions on*, 46(6):1057–1068, 1999.

[134] Deven Agravat, Urmi Vaishnav, and PB Swadas. Modified ant miner for intrusion detection. In *Machine Learning and Computing (ICMLC), 2010 Second International Conference on*, pages 228–232. IEEE, 2010.

[135] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH Computer Graphics*, volume 21, pages 25–34. ACM, 1987.

[136] Marjorie C Meehan. Sociobiology: The new synthesis. *JAMA*, 233(9):1006–1006, 1975.

[137] James Kennedy, Russell Eberhart, et al. Particle swarm optimization. In *Proceedings of IEEE international conference on neural networks*, volume 4, pages 1942–1948. Perth, Australia, 1995.

[138] Emmanuel Michailidis, Sokratis K Katsikas, and Efstratios Georgopoulos. Intrusion detection using evolutionary neural networks. In *Informatics, 2008. PCI'08. Panhellenic Conference on*, pages 8–12. IEEE, 2008.

[139] Qinghua Zhang and Albert Benveniste. Wavelet networks. *Neural Networks, IEEE Transactions on*, 3(6):889–898, 1992.

[140] Yuan Liu, Ruhui Ma, and Xing Lin. Network anomal detection wavelet neural network based on qpso. *Journal of Liaoning Technical University (Natural Science)*, 2:030, 2009.

[141] Li-li Liu and Yuan Liu. Mqpso based on wavelet neural network for network anomaly detection. In *Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on*, pages 1–5. IEEE, 2009.

[142] Martin Fodslette Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, 6(4):525–533, 1993.

[143] Hao Yang, Haiyun Luo, Fan Ye, Songwu Lu, and Lixia Zhang. Security in mobile ad hoc networks: challenges and solutions. *Wireless Communications, IEEE*, 11 (1):38–47, 2004.

[144] Ruhui Ma, Yuan Liu, and Xing Lin. Hybrid qpso based wavelet neural networks for network anomaly detection. In *Digital Media and its Application in Museum & Heritages, Second Workshop on*, pages 442–447. IEEE, 2007.

[145] Magnus Rudolph Hestenes and Eduard Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS, 1952.

[146] Ru Hui Ma and Yuan Liu. Wavelet fuzzy neural network based on modified qpso for network anomaly detection. *Applied Mechanics and Materials*, 20:1378–1384, 2010.

[147] Mark JL Orr et al. Introduction to radial basis function networks, 1996.

[148] Ruhui Ma, Yuan Liu, Xing Lin, and Zhang Wang. Network anomaly detection using rbf neural network with hybrid qpso. In *Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on*, pages 1284–1287. IEEE, 2008.

[149] Wen Jie Tian and Ji Cheng Liu. A new network intrusion detection identification model research. In *Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on*, volume 2, pages 9–12. IEEE, 2010.

[150] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[151] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[152] Jun Wang, Xu Hong, Rong-rong Ren, and Tai-hang Li. A real-time intrusion detection system based on pso-svm. In *International Workshop on Information Security and Application*, pages 319–321, 2009.

[153] James Kennedy and Russell C Eberhart. A discrete binary version of the particle swarm algorithm. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, volume 5, pages 4104–4108. IEEE, 1997.

[154] Jing Ma, Xingwei Liu, and Sijia Liu. A new intrusion detection method based on bpso-svm. In *Computational Intelligence and Design, 2008. ISCID'08. International Symposium on*, volume 1, pages 473–477. IEEE, 2008.

[155] Surat Srinoy. Intelligence system approach for computer network security. In *Proceedings of the Fourth IASTED Asian Conference on Communication Systems and Networks*, AsiaCSN '07, pages 89–95, 2007. ISBN 978-0-88986-658-4. URL http://dl.acm.org/citation.cfm?id=1712866.1712886.

[156] Tie-Jun Zhou, Yang Li, and Jia Li. Research on intrusion detection of svm based on pso. In *Machine Learning and Cybernetics, 2009 International Conference on*, volume 2, pages 1205–1209. IEEE, 2009.

[157] Lizhong Xiao, Zhiqing Shao, and Gang Liu. K-means algorithm based on particle swarm optimization algorithm for anomaly intrusion detection. In *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, volume 2, pages 5854–5858. IEEE, 2006.

[158] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14. California, USA, 1967.

[159] Chen Guolong, Chen Qingliang, and Guo Wenzhong. A pso-based approach to rule learning in network intrusion detection. In *Fuzzy Information and Engineering*, pages 666–673. Springer, 2007.

[160] Zhao Chang and Wang Wei-ping. An improved pso-based rule extraction algorithm for intrusion detection. In *Computational Intelligence and Natural Computing, 2009. CINC'09. International Conference on*, volume 2, pages 56–58. IEEE, 2009.

[161] J. L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien. The dynamics of collective sorting robot-like ants and ant-like robots. In *Proceedings of the First International Conference on Simulation of Adaptive Behavior on From Animals to Animats*, pages 356–363, 1990. ISBN 0-262-63138-5. URL http://dl.acm.org/citation.cfm?id=116517.116557.

[162] Erik D. Lumer and Baldo Faieta. Diversity and adaptation in populations of clustering ants. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior : From Animals to Animats 3: From Animals to Animats 3*, SAB94, pages 501–508, 1994. ISBN 0-262-53122-4. URL http://dl.acm.org/citation.cfm?id=189829.190043.

[163] André L Vizine, Leandro N de Castro, and Ricardo R Gudwin. Text document classification using swarm intelligence. *Proc. of KIMAS*, 2005.

[164] Vitorino Ramos and Ajith Abraham. Antids: Self orga nized ant-based c lustering model for intrusion det ection system. In *Soft Computing as Transdisciplinary Science and Technology*, pages 977–986. Springer, 2005.

[165] Wilson Tsang and Sam Kwong. Unsupervised anomaly intrusion detection using ant colony clustering model. In *Soft Computing as Transdisciplinary Science and Technology*, pages 223–232. Springer, 2005.

[166] Chi-Ho Tsang and Sam Kwong. Ant colony clustering and feature extraction for anomaly intrusion detection. In *Swarm Intelligence in Data Mining*, pages 101–123. Springer, 2006.

[167] Chi-Ho Tsang and Sam Kwong. Multi-agent intrusion detection system in industrial network using ant colony clustering approach and unsupervised feature extraction. In *Industrial Technology, 2005. ICIT 2005. IEEE International Conference on*, pages 51–56. IEEE, 2005.

[168] Yong Feng, Zhong-Fu Wu, Kai-Gui Wu, Zhong-Yang Xiong, and Ying Zhou. An unsupervised anomaly intrusion detection algorithm based on swarm intelligence. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, volume 7, pages 3965–3969. IEEE, 2005.

[169] Yong Feng, Jiang Zhong, Chun-xiao Ye, and Zhong-fu Wu. Clustering based on self-organizing ant colony networks with application to intrusion detection. In *Intelligent Systems Design and Applications, 2006. ISDA'06. Sixth International Conference on*, volume 2, pages 1077–1080. IEEE, 2006.

[170] Yong Feng, Jiang Zhong, Zhong-yang Xiong, Chun-xiao Ye, and Kai-gui Wu. Intrusion detection classifier based on dynamic som and swarm intelligence clustering. In *Advances in Cognitive Neurodynamics ICCN 2007*, pages 969–974. Springer, 2008.

[171] Teuvo Kohonen. *Self-organizing maps*, volume 30. Springer, 2001.

[172] Damminda Alahakoon, S Halgamuge, and Bala Srinivasan. Dynamic self-organizing maps with controlled growth for knowledge discovery. *Neural Networks, IEEE Transactions on*, 11(3):601–614, 2000.

[173] Qinglei Zhang and Wenying Feng. Network intrusion detection by support vectors and ant colony. In *Proceedings of the 2009 International Workshop on Information Security and Application (IWISA 2009)*, 2009.

[174] Man Zhao, Jing Zhai, and Zhouqian He. Intrusion detection system based on support vector machine active learning and data fusion. In Zhihua Cai, Chengyu Hu, Zhuo Kang, and Yong Liu, editors, *Advances in Computation and Intelligence*, volume 6382 of *Lecture Notes in Computer Science*, pages 272–279. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-16492-7. doi: 10.1007/978-3-642-16493-4_28. URL http://dx.doi.org/10.1007/978-3-642-16493-4_28.

[175] S Srinoy. An adaptive ids model based on swarm intelligence and support vector machine. In *Proceedings of the International Symposium on Communications and Information Technologies (ISCIT'06)*, pages 584–589, 2006.

[176] Parag M Kanade and Lawrence O Hall. Fuzzy ants as a clustering concept. In *Fuzzy Information Processing Society, 2003. NAFIPS 2003. 22nd International Conference of the North American*, pages 227–232. IEEE, 2003.

[177] Stephanie Forrest, Alan S. Perelson, Lawrence Allen, and Rajesh Cherukuri. Self-nonself discrimination in a computer. In *Proceedings of the 1994 IEEE Symposium on Security and Privacy*, SP '94, pages 202–, Washington, DC, USA, 1994. IEEE Computer Society. URL http://dl.acm.org/citation.cfm?id=882490.884218.

[178] Paul D Williams, Kevin P Anchor, John L Bebo, Gregg H Gunsch, and Gary D Lamont. Cdis: Towards a computer immune system for detecting network intrusions. In *Recent Advances in Intrusion Detection*, pages 117–133. Springer, 2001.

[179] Jung Won Kim. *Integrating artificial immune algorithms for intrusion detection*. PhD thesis, University College London (University of London), 2002.

[180] Julie Greensmith, Uwe Aickelin, and Steve Cayzer. Detecting danger: The dendritic cell algorithm.

[181] Jamie Twycross and Uwe Aickelin. Libtissue-implementing innate immunity. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 499–506. IEEE, 2006.

[182] Dipankar Dasgupta, Senhua Yu, and Nivedita Sumi Majumdar. Mila–multilevel immune learning algorithm and its application to anomaly detection. *Soft Computing*, 9(3):172–184, 2005.

[183] Charles Darwin and William F Bynum. *The origin of species by means of natural selection: or, the preservation of favored races in the struggle for life*. AL Burt, 2009.

[184] HaiH. Dam, Kamran Shafi, and HusseinA. Abbass. Can evolutionary computation handle large datasets? a study into network intrusion detection. In Shichao Zhang and Ray Jarvis, editors, *AI 2005: Advances in Artificial Intelligence*, volume 3809 of *Lecture Notes in Computer Science*, pages 1092–1095. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-30462-3. doi: 10.1007/11589990_146. URL http://dx.doi.org/10.1007/11589990_146.

[185] Alexander Hofmann, Carsten Schmitz, and Bernhard Sick. Rule extraction from neural networks for intrusion detection in computer networks. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 2, pages 1259–1265. IEEE, 2003.

[186] Sampada Chavan, Khusbu Shah, Neha Dave, Sanghamitra Mukherjee, Ajith Abraham, and Sugata Sanyal. Adaptive neuro-fuzzy intrusion detection systems. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, volume 1, pages 70–74. IEEE, 2004.

[187] Yihua Liao, V Rao Vemuri, and Alejandro Pasos. Adaptive anomaly detection with evolving connectionist systems. *Journal of Network and Computer Applications*, 30(1):60–80, 2007.

[188] Elizabeth Leon, Olfa Nasraoui, and Jonatan Gomez. Anomaly detection based on unsupervised niche clustering with application to network intrusion detection. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 1, pages 502–508. IEEE, 2004.

[189] Jiu-Ling Zhao, Jiu-Fen Zhao, and Jian-Jun Li. Intrusion detection based on clustering genetic algorithm. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, volume 6, pages 3911–3914. IEEE, 2005.

[190] Wei Lu. *An unsupervised anomaly detection framework for multiple-connection based network intrusions*. University of Victoria, 2006.

[191] Serge Fenet and Salima Hassas. A distributed intrusion detection and response system based on mobile autonomous agents using social insects communication paradigm. *Electronic Notes in Theoretical Computer Science*, 63:41–58, 2002.

[192] Noria Foukia. Idream: intrusion detection and response executed with agent mobility. In *Engineering self-organising systems*, pages 227–239. Springer, 2005.

[193] Soumya Banerjee, Crina Grosan, and Ajith Abraham. Ideas: intrusion detection based on emotional ants for sensors. In *Intelligent Systems Design and Applications, 2005. ISDA'05. Proceedings. 5th International Conference on*, pages 344–349. IEEE, 2005.

[194] Soumya Banerjee, Crina Grosan, Ajith Abraham, and PK Mahanti. Intrusion detection on sensor networks using emotional ants. *International Journal of Applied Science and Computations*, 12(3):152–173, 2005.

[195] Chia-Mei Chen, Bing Chiang Jeng, Chia Ru Yang, and Gu Hsin Lai. Tracing denial of service origin: Ant colony approach. In *Applications of Evolutionary Computing*, pages 286–295. Springer, 2006.

[196] Linkoln Laboratory. Darpa intrusion detection evaluation @ONLINE, June 2014. URL http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/2000data.html.

[197] CAIDA. The caida "ddos attack 2007" dataset @ONLINE, June 2014. URL http://www.caida.org/data/passive/ddos-20070804_dataset.xml.

[198] UNIBS. Unibs-2009 @ONLINE, June 2014. URL http://www.ing.unibs.it/ntw/tools/traces/.

[199] The Shmoo Group. Cctf-defcon10 @ONLINE, June 2014. URL http://cctf.shmoo.com/.

[200] ISCX. The iscx data set @ONLINE, June 2014. URL http://iscx.ca/datasets.

[201] Android Genome Project. Android malware genome project @ONLINE, June 2014. URL http://www.malgenomeproject.org/.

[202] Yajin Zhou and Xuxian Jiang. Dissecting android malware: Characterization and evolution. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 95–109. IEEE, 2012.

[203] KDD. Kdd cup 1999 data @ONLINE, June 2014. URL http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[204] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali-A Ghorbani. A detailed analysis of the kdd cup 99 data set. In *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009*, 2009.

[205] H Günes Kayacik, A Nur Zincir-Heywood, and Malcolm I Heywood. Selecting features for intrusion detection: a feature relevance analysis on kdd 99 intrusion detection datasets. In *Proceedings of the third annual conference on privacy, security and trust*, 2005.

[206] Matthew V Mahoney and Philip K Chan. An analysis of the 1999 darpa/lincoln laboratory evaluation data for network anomaly detection. In *Recent Advances in Intrusion Detection*, pages 220–237. Springer, 2003.

[207] Maheshkumar Sabhnani and Gursel Serpen. Why machine learning algorithms fail in misuse detection on kdd intrusion detection data set. *Intelligent Data Analysis*, 8(4):403–415, 2004.

[208] John McHugh. Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM transactions on Information and system Security*, 3(4):262–294, 2000.

[209] John Ross Quinlan. *C4. 5: programs for machine learning*, volume 1. Morgan kaufmann, 1993.

[210] Bo Liu, Hussein A Abbass, and Bob McKay. Density-based heuristic for rule discovery with ant-miner. In *The 6th Australia-Japan joint workshop on intelligent and evolutionary system*, volume 184. Citeseer, 2002.

[211] Bo Liu, Hussein A Abbass, and Bob McKay. Classification rule discovery with ant colony optimization. In *Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*, pages 83–83. IEEE Computer Society, 2003.

[212] Ziqiang Wang and Boqin Feng. Classification rule mining with an improved ant colony algorithm. In *AI 2004: Advances in Artificial Intelligence*, pages 357–367. Springer, 2005.

[213] Allen Chan and Alex Freitas. A new classification-rule pruning procedure for an ant colony algorithm. In *Artificial Evolution*, pages 25–36. Springer, 2006.

[214] James Smaldon and Alex A Freitas. A new version of the ant-miner algorithm discovering unordered rule sets. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 43–50. ACM, 2006.

[215] Fernando EB Otero, Alex A Freitas, and Colin G Johnson. cant-miner: an ant colony classification algorithm to cope with continuous attributes. In *Ant colony optimization and swarm intelligence*, pages 48–59. Springer, 2008.

[216] Khalid M Salama, Ashraf M Abdelbar, and Alex A Freitas. Multiple pheromone types and other extensions to the ant-miner classification rule discovery algorithm. *Swarm Intelligence*, 5(3-4):149–182, 2011.

[217] Ling Chen and Li Tu. Parallel mining for classification rules with ant colony algorithm. In *Computational Intelligence and Security*, pages 261–266. Springer, 2005.

[218] Yixin Chen, Ling Chen, and Li Tu. Parallel ant colony algorithm for mining classification rules. In *GrC*, pages 85–90, 2006.

[219] Omid Roozmand and Kamran Zamanifar. Parallel ant miner 2. In *Artificial Intelligence and Soft Computing–ICAISC 2008*, pages 681–692. Springer, 2008.

[220] Janaki Chintalapati, Maan Arvind, S Priyanka, N Mangala, and Jayaraman Valadi. Parallel ant-miner (pam) on high performance clusters. In *Swarm, Evolutionary, and Memetic Computing*, pages 270–277. Springer, 2010.

[221] Christos Xenakis and Christoforos Ntantogian. An advanced persistent threat in 3g networks: Attacking the home network from roaming networks. *Computers & Security*, 40:84–94, 2014.

[222] Angelos Stavrou Constantinos Kolias, Georgios Kambourakis and Stefanos Gritzalis. Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset. *Submitted in Communications Surveys & Tutorials, IEEE*, -(-):.–., 2013.

[223] Constantinos Kolias and Georgios Kambourakis. Termid: A distributed ant colony based ids. *Under Submision*, -(-):.–., 2014.

[224] Georgios Kambourakis, Constantinos Kolias, Stefanos Gritzalis, and Jong Hyuk-Park. Signaling-oriented dos attacks in umts networks. In *Advances in Information Security and Assurance*, pages 280–289. Springer, 2009.

[225] Georgios Kambourakis, Constantinos Kolias, Stefanos Gritzalis, and Jong Hyuk Park. Dos attacks exploiting signaling in umts and ims. *Computer Communications*, 34(3):226–235, 2011.

[226] Constantinos Kolias, Georgios Kambourakis, and M Maragoudakis. Swarm intelligence in intrusion detection: A survey. *computers & security*, 30(8):625–642, 2011.