Wireless Security and Traffic Modeling Using Benford's Law

by

Nayla Hamadeh

B.Eng., American University of Beirut, 2002

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Electrical Engineering

The University of New Mexico

Albuquerque, New Mexico

June, 2004

Dedication

To my parents, for all their love and support

"The love of a family is life's greatest blessing"
– Anonymous

Acknowledgments

First, I would like to thank my parents, without whom i wouldn't be where i am today. Mom and Dad, thank you for always believing in me and encouraging me to give my best at everything I do. Your love and support, even thousands of miles away, keep pushing me forward. And to my lovely sister Nada, whose beautiful grey eyes and smile always bring light to the darkest days, thank you for being the best sister one can hope for.

I would also like to thank my advisor Prof. Chaouki Abdallah for his patient guidance, constant encouragement and valuable advice throughout these two years that I have spent at UNM. William Arthur Ward once said "The mediocre teacher tells. The good teacher explains. The superior teacher demonstrates. The great teacher inspires". Thank you professor for being the greatest of all teachers, and for that I am forever grateful.

I would also like to acknowledge Prof. Gregory Heileman and Prof. Majeed Hayat for taking time of their busy schedule to be part of my committee. I also would like to acknowledge Prof. Max Costa for his great interest in my topic and all the intriguing ideas and constructive comments he has provided me with. I want to extend my gratitude to Mr. Lou Sullo, Dr. Art St-George and all the staff at the Computer and Information Resources and Technology department (CIRT) for allowing me to be part of a networking team and have practical field experience. I would also like to thank Mr. Luay Shawwa for all his contributions and assistance specially in providing us with the wireless security products.

I would also like to extend my sincere gratitude to all those who have helped me complete this thesis, namely Henry Jerez and Jean Ghanem for their help with the testing of wireless security solutions. I would also like to thank all the great people I have met here at UNM for making Albuquerque a home away from home.

Finally, I would like to extend my deepest gratitude to the person who stood by me for the past six years, making every step of the difficult road seem simple! Jean, I thank you for all the moments that we spent together, moments that gave me enough strength to keep going even in moments of despair. Franklin P. Jones once said that "Love doesn't make the world go 'round. Love is what makes the ride worthwhile". You showed me that the ride can, not only be worthwhile, but also extremely enjoyable! Thank you for being the light in my life!

Wireless Security and Traffic Modeling Using Benford's Law

by

Nayla Hamadeh

ABSTRACT OF THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Electrical Engineering

The University of New Mexico

Albuquerque, New Mexico

June, 2004

Wireless Security and Traffic Modeling Using Benford's Law

by

Nayla Hamadeh

B.Eng., American University of Beirut, 2002

M.S., Electrical Engineering, University of New Mexico, 2004

Abstract

Wireless networks have gained great popularity over the last few years, being able to move around while still maintaining connection is appealing to everyone. Along with the development of wireless networks came the need to provide a suitable level of security to ensure the privacy of the users. Several standards have been proposed and it is one of this thesis's objectives to describe and explain them pointing out their shortcomings. Additionally, there exist several products on the market today that promise to render the task of controlling access and managing wireless networks easier. In this thesis, in addition to overviewing the existing security solutions for wireless networks, we propose to model network traffic, specifically traffic from and to a webserver, using Benford's Law. Benford's Law also known as the law of anomalous numbers, states that the frequency of occurrence of digit d as the leading digit in naturally occurring numbers follows a logarithmic distribution, and not a uniform one as previously thought.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The last decade has witnessed the considerable growth of wireless networks. Traditional ways of networking no longer match the fast development of technology that thrives to release the world of the constraints imposed by wires. Mobility has become one of the most features sought and current networks fall short in satisfying that need.

Wireless networks bring flexibility to the world; with no constraints, the users are given the chance to move around and roam freely while maintaining their connection to the existing networks. Installing wireless equipment in an enterprise has proven to be much cheaper than physically wiring the rooms, and needless to say less cumbersome.

Wireless networks group more than one category of products; cellular networks, Bluetooth and the different 802.xx flavors all fall under this general term. Whether GSM, CDMA or the new third generation cellular phones (UMTS) are used, a considerable proportion of the world's population is currently using cellular services, accompanied with a great improvement in cellular phone manufacturing, where nowadays, phones are not only a tool for communication but also a fashion statement. Another

wireless standard is Bluetooth, a protocol used in the communication between several devices in a small area usually less than 30 feet in diameter. The limited area of coverage makes it somewhat inconvenient for deployment of Bluetooth in enterprises or offices with large concentration of users. To address the different issues of coverage and mobility in wireless networks, the Institute of Electrical and Electronics Engineers (IEEE) has and is still developing standards for wireless Local Area Networks (WLAN), Personal Area Networks (PAN) and Metropolitan Area Networks (MAN). A survey of the different standards is available in [21]. The standard that is of most interest to us is the IEEE 802.11 also known as Wi-Fi. 802.11 comes in three flavors: a, b and g. What differentiates these three types is the modulation used, the number of channels provided, the frequency of operation and the bandwidth. Table 1.1 summarizes the characteristics and underlines the differences among the different 802.11 standards. An overview of Bluetooth and IEEE 802.11 is provided in [42].

	802.11a	**802.11b**	**802.11g**
Frequency	5 GHz	2.4 GHz	2.4 GHz
Bandwidth	54 Mbps	11 Mbps	54 Mbps
Number of Channels	12 channels	11 channels	11 channels
Modulation	OFDM multiple carrier	DSSS Single carrier	OFDM

Table 1.1: IEEE 802.11 standards: a, b, g

The fast growth of wireless networks was accompanied by an increasing need for security. In Wireless networks, data is not protected by wires, but simply propagates, accessible to anyone who possesses a wireless device. Many security standards have been developed and while some have failed to provide the degree of privacy sought, others provide temporary solutions. Still, the look-out for the best solution continues in the hope of finding the ultimate security provider.

1.1 Objective of this Thesis

This thesis has two objectives. First, it provides an overview of the different security solutions available on the market today. The survey covers every popular solution exploited in wireless networks ranging from MAC filtering to the upcoming IEEE 802.11i standard, to different out-of-the-box products that can be purchased for enterprise level security. The second goal is to provide a novel method to model traffic on the network based on Benford's Law. We not only try to offer a new explanation of the law but also show that it can be efficiently used in the modeling of network traffic, which would hopefully lead to a new technique in detecting abnormalities in a network, especially wireless ones.

1.2 Overview of Thesis

The thesis is composed of 2 major parts. The first part is concerned with the current wireless security solutions and the second with Benford's Law modeling.

Part one is divided into three chapters. In Chapter 2, we identify the four elements of security: authentication, confidentiality, integrity and availability. We then present the several types of attacks that a network can experience while emphasizing the effects of attacks specifically on wireless networks. Chapter 3 overviews the diverse measures currently used in wireless networks to provide a level of security that ensures that both the users and devices of the network are well protected. That chapter also describes the Wired Equivalent Privacy (WEP) protocol while identifying its weaknesses and known flaws, the Wi-Fi Protected Access (WPA) protocol, the upcoming 802.11i protocol which should be standardized by the end of 2004, and VPN tunnels, generally a security solution for wired networks. Each of these methods is explained and then evaluated based on the level of privacy it provides and how

simple it is to be overcome. In addition to the security measures that anyone can administer, there are security solutions on the market today that address wireless privacy and offer management tools for large networks. Three of these products, Roving Planet, Bluesocket, and Cranite are examined in Chapter 4. The only product that we actually had the chance to test in our laboratory was Roving Planet, the remaining two were simply overviewed based on the documentation available on their respective websites.

The second part of the thesis consisting of Chapter 5, deals with Benford's Law. First, the history and concept are explained, followed by the different proofs trying to clarify why such a phenomenon exists. We also offer a new method to approximate the law based on maximum entropy distribution and, although we could not match the exact values provided by the law, we were able to approximate it. This chapter also surveys the different areas where Benford's Law has been applied and proposes to employ it in the area of networking, more precisely to the number of hits on a webserver, the inter-arrival times between any two consecutive requests and the number of bytes exchanged in each connection. As will be seen in that chapter, Benford's Law turns out to be a good method to model such traffic, which could be utilized in the detection of abnormalities in the network, both wired and wireless. Finally, Chapter 6 provides our conclusions and recommendations for future work.

Chapter 2

Network Security

2.1 Elements of Security

A wired network is usually secured by the walls of the building that contains it, and even then, the security is not absolute as there exist more than one method to penetrate it or control its functioning. Wireless Networks, on the other hand, have no physical boundaries and thus are even more vulnerable to attacks. Securing networks, whether wired or wireless, as specified by [57], [46], [61] and [60], depends mainly on three pillars: the "CIA", Confidentiality, Integrity and Availability. Added to these three is authentication that secures the gate to the network. In this section, each of the four elements of security is explained.

2.1.1 Authentication

Authentication consists of verifying that the person sending the message is really who he/she claims they are: i.e. making sure that the message has originated from the intended sender and not a third party pretending to be a legitimate user. Unlike

wired networks where the user must have a computer physically connected in order to gain access to the network, in wireless environments, everyone can view the network but those who wish to actually use it, must provide their credentials and prove who they really are. The issue of wireless authentication will be further explained in subsequent chapters of this thesis.

2.1.2 Confidentiality/ Privacy

Confidentiality or Privacy, "For your eyes only", ensures that only the destined recipient is entitled to view and read the contents of the message. The definition of confidentiality extends beyond unauthorized disclosure of information; sometimes, the mere monitoring of activity is also considered a violation of privacy. In fact, data such as the date of message, the length and the destination, are usually used by Internet Service Providers (ISP), for analysis purposes. The best solution to ensure privacy is by using encryption. Encryption can range from steganography, where the intended information is hidden in the message, to more sophisticated techniques involving protocols and standards such as the Rivest, Shamir, & Adleman public key encryption technology (RSA) , the Data Encryption Standard (DES) and most recently the Advanced Encryption Standard (AES). In wireless networks, further measures need to be taken since they have no real physical boundaries. The security of wireless networks will be fully discussed in chapter 3.

2.1.3 Integrity

Integrity consists of making sure that the message received is the exact same message that was sent, i.e. to ensure that no duplication, insertion, deletion or modification of the message has occurred. As will be seen in Section 2.2, integrity is the target of many attacks that aim to alter the message in order to obtain a specific sought

response or even unveil private information such as credentials. In wireless networks, protecting the integrity of a message is of the essence since packets can easily be sniffed and a change in message content could have major consequences.

2.1.4 Availability

Availability may not be the most important security element and some literature neglect even mentioning it but, nonetheless, if the network is not available, we cannot address any of the other security elements. In both wired and wireless networks, the users can get connected only if the network exists. Availability is the target of Denial-of-Service (DOS) attacks as will be explained in Section 2.2. Availability ensures reliability and stability of the network, since users require uninterrupted access to resources.

2.2 Common Attacks

All networks are vulnerable and prone to attacks by their nature. Attacks are commonly classified according to which of the security elements they target. Attacks can be caused by an external hacker, far away from the network, or by an insider or legitimate user. The latter type of attack is referred to as the insider threat problem. Whether the attack is orchestrated from the inside or the outside, the consequences on the operation of the network can be devastating. In [60], [27], [46] and [63], different types of attacks are explained. In what follows, we will shed some light on the three major types of attacks.

2.2.1 Eavesdropping

Eavesdropping is a direct attack on confidentiality. The attacker monitors the activity of the network and may even read the content of the messages exchanged. Eavesdropping is a serious issue for both wired and wireless networks. In a Local Area Network (LAN), all machines and media are interconnected, therefore picking up packets destined for another machine is not a difficult task. In wireless networks, the task is even simpler, packets are already in the air and the chore is reduced to simply gathering them as illustrated in Figure 2.1.

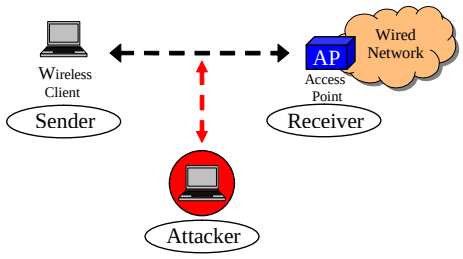

Figure 2.1: The Attacker listens to the communication between the Sender and the Receiver and possibly saves some of these messages for later use

If no encryption is used, the content of the message is easily accessible and even with some of the security currently used in wireless environments such as WEP, retrieving information has proven feasible with tools available online due to known weaknesses in the protocol. This particular issue will be addressed in the next chapter. Eavesdropping can be passive: the hacker silently sniffs packets and discovers characteristics about the source, destination, and size of the message. It can also be active: the hacker intentionally injects packets in the network in order to reveal additional information. Active eavesdropping can be accomplished only if the attacker has access to the transmission. In this case, the attack also targets authentication since the hacker is masquerading as a legitimate user.

2.2.2 Man-in-the-Middle Attack

Man-in-the-Middle (MITM) attacks generally target the integrity of the message. They rely on the fact that an attacker is intercepting all communication between two parties. The attacker can simply read the information exchanged, thus violating the privacy of the message, or modify the content of the message (which is usually the case), subsequently attacking its integrity. MITM attacks are numerous and frequently divided into three categories: simple MITM, Session highjacking, and Replay attacks.

Simple Man-in-the-Middle attacks are real-time attacks where the hacker is monitoring all communication and modifying packets. MITM exist in both wired and wireless networks. In LANs, intercepting communication between two parties requires spoofing of DNS or ARP requests so that it becomes possible to obtain addresses of target machines. In wireless networks, the fact that users can associate to Access points without positively identifying their legitimacy, renders it easy for an attacker to establish a rogue access point using the same Identifier (SSID) as the network. Hence, unsuspected users will mistakenly associate to that access point and possibly reveal sensitive information. The process of a simple MITM attack is illustrated in Figure 2.2.

Session highjacking involves the theft of a session by an attacker. One of the parties communicating is completely replaced by the attacker who poses as a legitimate user. In a wireless context, the parties consist of the Access Point and the user. An attacker can steal an authenticated user's session and use it as his own to gain access to the network. Figure 2.3 depicts this operation.

Finally, a replay attack, the third type of MITM, is also used to gain access

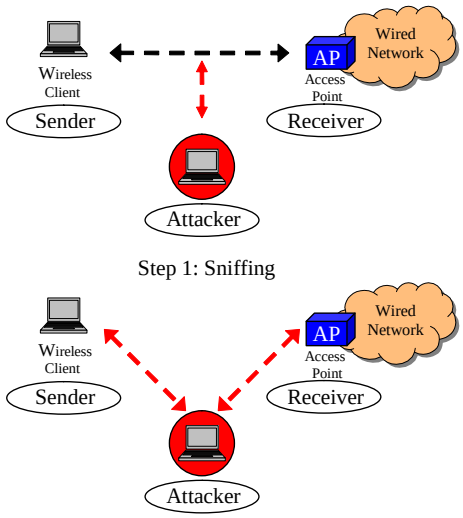

Figure 2.2: Simple Man-in-the-Middle Attack. The attacker first listens to the communication and then acts as the second party with respect to both the sender and the receiver

to the network, but not in real-time. The attacker captures messages exchanged, specifically those related to authentication, and then at a later time, uses these same messages to authenticate posing as a legitimate party. The process is shown in Figure 2.4.

All Man-in-the-Middle attacks rely mainly on eavesdropping as a first step and then plan the next move that will allow the attacker entry to the network.

2.2.3 Denial of Service Attack

Denial-of-Service (DOS) attacks target the availability of the network. The goal of a DOS attack is to flood the network with useless traffic that it becomes so busy that even legitimate users are denied access. For wireless networks, DOS attacks can

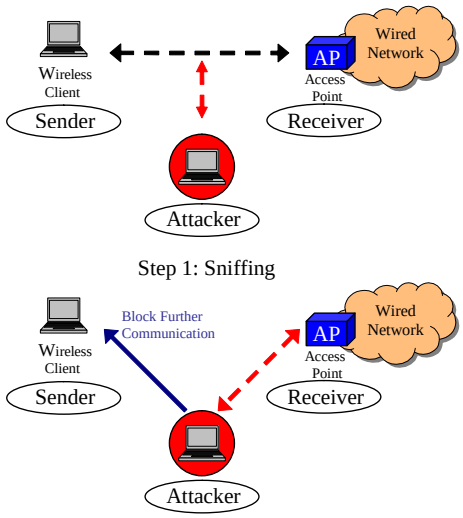

Figure 2.3: Session Highjacking Attack. The attacker first listens to the communication and then acts as the sender thus blocking all communications with the original party

range from physically damaging the Access point to creating enough interference in the frequency spectrum occupied by Wi-Fi standards (5 GHz for 802.11a and 2.4 GHz for 802.11b/g). Jamming the signal is not however a very easy task and requires several pieces of specialized equipment.

2.3 Summary

The security of a network depends on the presence of the elements explained in this chapter. Authentication, confidentiality, integrity, and availability ensure that the network is safely shielded from attackers. However, when any of these elements is at risk, via any of the attacks described in Section 2.2, administrators need to take appropriate measures to guarantee that these elements are well preserved. The wireless community has offered many ways and ratified several standards to provide

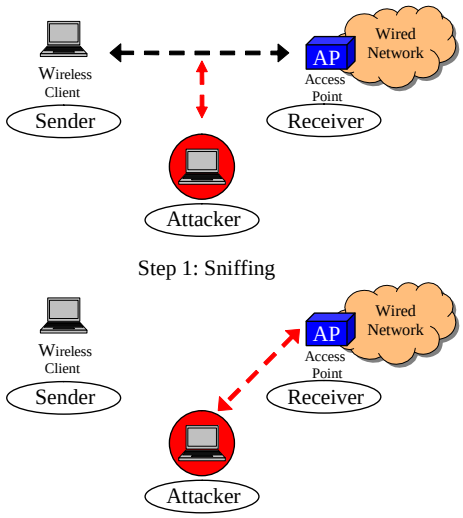

Figure 2.4: Replay Attack. The attacker first listens to the communication and then uses the information from that session to open a new communication channel with the second party.

an adequate level of security to wireless networks, as will be seen in Chapter 3

Chapter 3

Wireless Security Measures

Along with the growth of wireless networks came the increasing demand for stronger security. Several measures have been used, some specified by standards, others agreed upon in the wireless community. In this chapter, we present the different security measures that have been used since the emergence of the wireless world.

3.1 Open Authentication, Service Set Identifier, MAC Filtering

Open authentication, Service Set Identifiers (SSID), and MAC address filtering are the most primitive forms of security for wireless networks. Each was used thoroughly for a period of time before being judged vulnerable and consequently getting sequestered from the wireless security world. Nevertheless, in order to properly understand how wireless networks reached the current state, one needs to look at their history and these three methods represent the first step administrators use to control access to their networks.

Open authentication, as described in [14] and [56], is the default authentication protocol that became popular when the 802.11 standard first emerged. The idea behind it is that access to the network is granted to anyone who requests it. In the mid 90s, when wireless was spreading, open authentication was useful for many 802.11 compliant devices such as bar code readers that lacked the capability to perform complex authentication algorithms but rather require direct and quick connection to the network. Nowadays, it is not recommended that a network be open to all those desiring access since it is not only jeopardizing the components of the networks but also all its users. In open authentication, the stations and their clients perform a rudimentary mutual authentication as seen in Figure 3.1 where the client sends an Association Request to the access point and the latter replies with an Association Response, giving the client instantaneous access to the network. The problem with this scenario is that all frames are sent in clear text and thus can easily be picked up by an attacker.

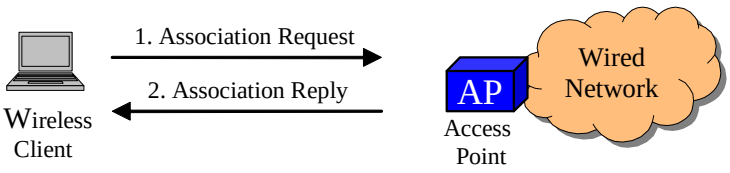

Figure 3.1: Open Authentication

Service Set Identifiers (SSID) act as identifiers of Wireless Local Area Networks (WLAN). A SSID can be viewed as a construct that allows logical separation between several WLANs. All documentation available on wireless networks ([66],[63], [59], [56], [19]) consider SSIDs to be an indication of the presence of wireless networks and warn about using it as a mean to secure the network since it provides no data privacy whatsoever. The wireless user, in order to go online, needs to choose the network to connect to by selecting the appropriate SSID. All access points belonging to the

same network must have the same SSID, and one access point may be configured to handle several networks and thus several SSIDs. By default, all access points periodically advertise their presence by broadcasting beacon frames containing the network's SSID and the channel used for communication. By explicitly sending the SSID, the access point is effectively sending an open invitation to anyone with wireless equipment including malicious attackers. This is evidently harmful since the security of the network is almost non-existent, especially if the SSID is used as a shared secret. In the early days of wireless networking, it was suggested that to keep the network safe, beacon broadcast must be disabled, an option available in all access points. After all, no one can harm a network if he does not know it exists. Nevertheless, this has proven to be useless since the SSID can be recovered relatively easy. In fact, there exist several online software such as Netstumbler ([9]) for Microsoft Windows and Kismet ([8]) for Linux, that detect all networks available in a certain spot along with their respective SSID, channels, and whether or not encryption is being used. Moreover, another way to pick up the SSID is for an attacker to send a frame to any legitimate user that causes him to disassociate from the access point, so that when the client tries to re-associate, the attacker can pick up the frame and extract the SSID, available in clear text in the header of each packet. Therefore, SSID should never be used as a security method since even when the broadcast is disabled, even the most unskilled attacker can manage to get it.

Another approach to securing a network is by controlling access to it using MAC address filtering or authentication. Medium Access Control (MAC) addresses are globally unique layer 2 identifiers that are manufacturer-specific. Actually, each manufacturer is allocated a 3 bytes organizationally unique identifier by the IEEE to use as prefix for the physical address of each of the devices they produce. That said, by looking at a MAC address, one can deduce the manufacturer. A MAC address thus identifies the actual physical device and not the human behind it as would be generally required. In wireless environments, access to the network can

be controlled using MAC address filtering ([66], [63], [14], [56], and [19]): deny someone access unless their MAC address is part of the list of allowed addresses. MAC authentication is not specified in the 802.11 standard but is supported in most access points. Figure 3.2 shows the process of authentication when MAC filtering is used: the wireless client sends an association request to the access point. In turn the AP checks its access list to ensure that this address is present and accordingly grant or deny user access.

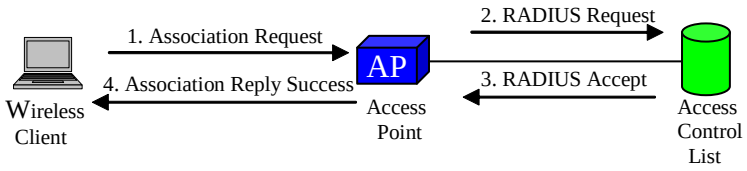

Figure 3.2: Authentication using MAC Filtering

Although widely used, this method has its drawbacks. First, for small networks it may be feasible to keep track of all MAC addresses permitted access but for large networks and enterprises, this method does not scale well. Second, MAC authentication does not provide any protection from insider attacks where the attacker already has an authorized WLAN card. Perhaps the main drawback is that MAC addresses can be sniffed since they are sent in clear text. An attacker can detect the MAC address and use it to get access to the network, posing as a legitimate user. In [64], Wright describes techniques used to disrupt wireless networks through MAC address spoofing where an attacker intentionally modifies his own MAC address to one belonging to an unsuspecting user. What makes this task achievable is the fact that most wireless cards allow the user to replace the Universally Administered Address (UAA), i.e. the original address, by a Locally Administered Address (LAA) using the card's software or a third party software available on the Internet. Although it

can be done, breaking the security of MAC filtering does require some knowledge and effort and attackers who simply want to go online, will probably avoid such hassle by looking for easier targets. Thus as affirmed in [63], MAC filtering does "raise the bar a little".

3.2 Wired Equivalent Privacy (WEP)

The first security solution for wireless networks as specified by the 802.11 standard was WEP. As its name indicates, it promises to deliver to wireless users the same level of security they would get if they were on a wired LAN. WEP is an encryption standard that was marketed for a period of time as "THE" security solution for wireless LANs. WEP promises two features: access control, i.e. preventing users with incorrect keys from gaining access to the network, and privacy by protecting the network traffic through encryption. The 802.11 standard specifies WEP to be a 40-bit secret phrase, but manufacturers thought it better to increase the size to 104 bits. Almost all literature discussing wireless networks include WEP-related information ([35], [19], [46] and [61]). The WEP standard aims to establish confidentiality, integrity, and authentication, the elements of security discussed in Section 2.1; confidentiality through frame body encryption, integrity via Integrity Check Sequence, and shared key authentication. However, as will be explained later, WEP fails to accomplish its goals.

The authentication process with the access point is illustrated in Figure 3.3. The client sends an authentication request to the access point. The access point replies with a 128-byte of text challenge. The client then encrypts the challenge using the WEP key and sends it back to the access point that checks if the encrypted reply is indeed what he expects and sends accordingly either a success or failure message.

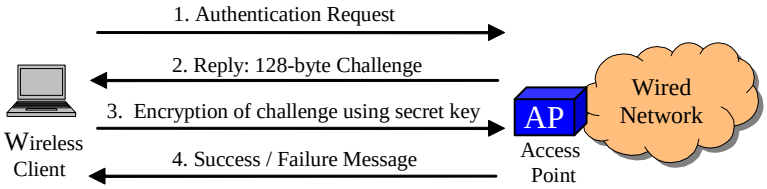

Figure 3.3: WEP Authentication

When WEP encryption is used, the bandwidth is slightly reduced: when 40-bit keys are used, the bandwidth is reduced by no less than 1 Mbps, and more than 2Mbps when 104-bit keys are used. This still remains a tolerable price to pay if in return security is achieved.

For encryption, WEP relies in the RC4 cipher developed by Rivest, a symmetric secret key method where ciphertext is obtained by combining a keystream with the original message. Stream ciphers are a compromise between security and practicality. The WEP encryption process is divided into 3 main steps: checksumming, encryption using RC4, and transmission. Since WEP vows to preserve the integrity of the message, the first step is to utilize the checksum. WEP uses the 32-bit Cyclic Redundancy Check (CRC) to generate an Integrity Check Value (ICV) which is then concatenated to the message to get the plaintext that is to be encrypted, i.e. $P =< M, c(M) >$, where c represents the checksum operation. Once the plaintext is ready, the encryption using RC4 begins. A 24-bit Initialization Vector (IV) is generated and merged with the secret key, the combination is then inputted to a Pseudo-Random Number Generator (PRNG) in order to get a keystream with size equal to the length of the plaintext. The keystream is next XORed (exclusive OR) with the plaintext to get the ciphertext: $C = P \bigoplus RC4(IV, key)$. Once the ciphertext is acquired, it is placed in a WEP frame as in Figure 3.4 along with the Initialization Vector (IV) used in the process and transmitted to the destination.

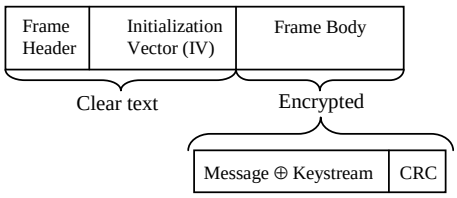

Figure 3.4: WEP Frame

The process described above is illustrated in Figure 3.5. To decrypt the message, the reverse procedure is followed: first, use the secret key and IV to recover the keystream which will be XORed with the ciphertext to obtain the plaintext. CRC-32 checksum is then used to make sure that no modification of the data has taken place.

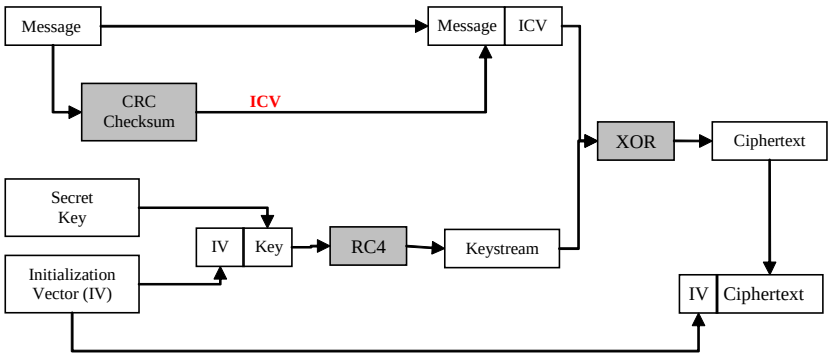

Figure 3.5: WEP Encipherment

Although used for quite sometime and still being used in several wireless environments, WEP has proven to be inefficient and vulnerable to more than one type of attacks. WEP's flaws are discussed in ([66], [63], [59], [14], [56], [19], [35], [46], [27], [61]). Firstly, all stations and clients in a wireless network share the same secret key, which completely obliterates any secrecy: a secret shared among hundreds of people can hardly be called a secret. Second, as pointed out by [66] and [27], WEP authentication is a one-way authentication, meaning that only the client needs to

prove his legitimacy to the access point and never the other way around, which implies that the client can easily become the victim of a malicious attacker with a rogue access point. Another issue with WEP is what appeared in a 2000 paper by Walker ([62]) where he explains that regardless of the size of the secret key, it is infeasible to achieve privacy with WEP mainly due to a problem in its design. Add to that the fact that CRC checksum is insufficient as was proved in ([22]) since controlled changes to the ciphertext are not reflected in the checksum and hence go unnoticed. But perhaps the biggest vulnerability in WEP is its misuse of the RC4 algorithm. In spring 2001, Fluhrer et al ([34]) presented the weaknesses in the key scheduling algorithm of RC4, and not long after Stubblefiled et al ([58]) implemented the attack described by [34]. In fact the problem is the keystream reuse: Initialization Vectors are 24-bit long, which means that there are 2^{12} possible combinations, a number easily exhausted within a moderate network in a short amount of time, and since the secret key is always kept the same, an attacker monitoring the traffic can build a table of keystreams and their corresponding IVs and reveal the plaintext. Actually, for two packets with the same IV, XORing the ciphertexts yields the same result as XORing the corresponding plaintext messages, therefore, by knowing one plaintext the other is easily recoverable either by guessing or sending a known message to the target machine and waiting for the corresponding ciphertext. This task is rendered simple with tools such as AirSnort ([1]) and WEPCrack ([11]) available for free download online. Despite the effort of researchers in trying to correct the flaws of WEP, such as RSA's Fast Packet Keying Solution which generated a unique key for each packet, the wireless community decided to move forward and create standards that provide the level of security that best suits everyone's needs.

3.3 IEEE 802.1x

As seen in the last section, WEP provides very little security and the need for better wireless protection grew as more people started to deploy wireless environments. One of the solutions was to use the IEEE 802.1x standard to control authentication; in other terms, be aware of who is accessing the network.

802.1x was originally designed for wired networks, but has proven to be equally useful in wireless networks, since it provides a level of control over who has access to the network and allows for an authenticated user to be uniquely identified. In fact, 802.1x identifies users by username and not by their MAC address, which takes care of the vulnerabilities concerning MAC spoofing described in Section 3.1.

The 802.1x is based on a protocol that enables port-based authentication; all traffic is blocked on a port-by-port basis until the client has correctly entered his credentials. 802.1x defines 3 components to the authentication conversation as specified in ([19]) and ([35]) and illustrated in Figure 3.6

- Supplicant: the client or end user who wishes to access the network
- Authenticator: the middle man who blocks or allows traffic. It acts as a bridge between the supplicant and the Authentication server, but does not contain any information. This is usually the Access Point
- Authentication Server: contains all user information, and processes incoming requests. Usually a RADIUS server.

802.1x relies on the Extensible Authentication Protocol (EAP). EAP is a layer 2 authentication ([27]), built around the challenge-response communication paradigm. The protocol is "extensible" because any authentication mechanism can be encapsulated within the EAP packets. EAP was originally developed for the Point-to-Point

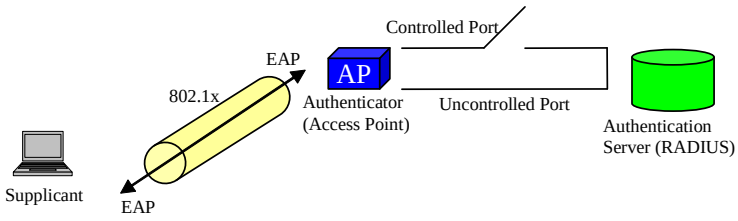

Figure 3.6: 802.1x Framework

Protocol (PPP) connections to provide a more flexible framework for authenticating users ([53]). Several methods of EAP are available and currently in use, and will be explained later on in the Section.

802.1x employs a dual-port model: an *Uncontrolled* port is used to allow only EAP packets through, and a *Controlled* port that allows traffic through only if the authentication was successfully completed. The 802.1x authentication process is illustrated in Figure 3.7. The supplicant sends an EAP start frame. The authenticator replies with an EAP Request/Identify packet to invite the client to identify himself. The supplicant then replies with an EAP Request/ Identify frame including his username, this frame is forwarded by the authenticator to the Authentication Server for processing. The Server then sends an EAP Reply message through the Authenticator to the Supplicant with a challenge and request for credentials. The client replies to the challenge and the response is again forwarded to the Authentication server for evaluation. Once, the server checks the result against the information stored, an EAP success or Failure is generated and the authenticator either opens the Controlled port for further communication or keeps it closed and disregard any future messages form the supplicant that are not EAP packets. The communication between the supplicant and the authenticator is done over the wireless link and uses the EAP over LAN (EAPOL) protocol. The link between the Authenticator and the authentication server is over the wired network.

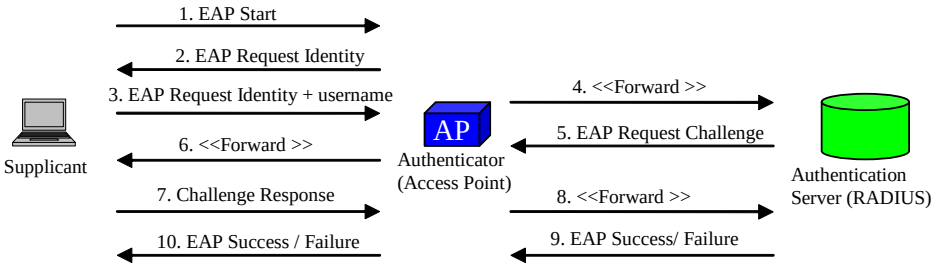

Figure 3.7: 802.1x Protocol

As was mentioned previously, the EAP protocol can be used in different methods, each has its pros and cons and each provides a certain level of security and comes at a certain implementation price. The first of these methods is the EAP-MD5 ([19]) and ([12]). This method provides the lowest level of security but it is the easiest to implement. The username is usually passed in clear text and the password encoded using MD5 hash, which makes it prone to dictionary attacks. EAP-MD5 does not support mutual authentication and hence is vulnerable to Man-in-the-Middle attacks as seen in Section 2.1, whereas an attacker can install a rogue access point and lure users to believe that it is part of the network. Another EAP method is the EAP-TLS or EAP Transport Layer Security. This offers the strongest security but is difficult to implement. It does provide mutual authentication eliminating possibility of MITM attacks. EAP-TLS works by establishing an end-to-end encrypted tunnel between the client and the access point, both of which have to use digital certifications to authenticate each other. The drawback of this approach is that an enterprise certificate of authority infrastructure is needed and the client authenticates through Public Key Infrastructure (PKI) certificate exchange. To avoid the hassle that accompanies the TLS method, EAP-TTLS or Tunneled TLS was designed to offer the same level of security of TLS while being less cumbersome. TTLS provides mutual authentication, and is constructed in 2 stages: first, an encrypted tunnel is established between the client and server, through which the server passes his certificate to the client. Then,

the client's credentials are sent to the server for evaluation. The advantage of this method over TLS is that only the server needs to provide certificates, the client uses simple passwords ([63]). Another method similar to EAP-TTLS is Protected EAP (PEAP). PEAP allows for mutual authentication, and server certificates are needed, but the client can use any other EAP authentication method such as MS-CHAP. The operation of PEAP starts by setting up a TLS connection over EAP, then the server followed by the client are authenticated and finally session keys are generated. Although PEAP and EAP-TTLS are considered very secure, ([18]) describes a Man-in-the-Middle attack on systems using these two methods and ([47]) and ([66]) illustrate a Session Highjacking attack whereas a hacker can steal a session in progress and masquerade as the legitimate user. It is important to mention that 802.1x does not provide any integrity guarantees since it is mainly an authentication process; if the wireless network requires confidentiality of the data, additional measures should be considered. An additional method for EAP authentication is the Lightweight EAP (LEAP) which is a CISCO proprietary solution that requires the use of CISCO equipment ([4]) and ([56]) but offers mutual authentication of the client and the access point. LEAP is a closed EAP type, i.e. its specifications are not made public, but what is known is that it uses a modified version of MS-CHAP v.2 for challenges and responses. However, recently vulnerabilities have been discovered in the LEAP protocol: in ([65]), Wright describes an attack on LEAP that results in breaking security; he also presents an implementation of the attack that can be downloaded and used freely. The vulnerability of LEAP as explained by Wright ([65]) and by CISCO ([28]) resides in a weakness in MS-CHAP which permits pre-computed dictionary attacks and weak DES key selection for challenge/response that allow the recovery of 2 bytes of hash. As a result of this vulnerability, CISCO has advised its users to refrain from using LEAP and switch to the newly developed EAP standard EAP-FAST, Flexible Authentication via Secure Tunneling since LEAP is "not worth fixing". EAP-Fast is an IETF effort ([25]) that uses symmetric key algorithms

to achieve a tunnelled authentication process. The setup of the tunnel relies on a Protected Access Credential (PAC), followed by the clients' authentication. CISCO ([29]) promises that the FAST solution is not proprietary and does not require a certificate authority. A summary of all EAP authentication methods are shown in Table 3.1.

Method	Typical Implementation	Authentication	Deployment Difficulty	Wireless Security
MD5	Challenge-based password	One-way authentication	Easy	Poor
TLS	Certificate Based two-way authentication	Mutual Authentication	Hard	Best
TTLS/PEAP	Server authentication via certificates; clients via other methods	Mutual Authentication	Moderate	Better
LEAP	Password-based	Mutual Authentication	Moderate	Recently Broken

Table 3.1: Comparison between EAP methods

3.4 Wi-Fi Protected Access - WPA

The security of wireless networks became more and more critical with the fast growing deployment of wireless environments. WEP has proven to be a very weak solution that could not be relied upon to provide a tightly controlled environment where the privacy of the users and security of the network as a whole are strongly required. To fix the vulnerabilities of WEP, the Wi-Fi Alliance designed the Wi-Fi Protected Access (WPA) to secure all 802.11, whether a, b or g, devices. WPA is a subset of the upcoming 802.11i standard discussed in the following section, that would bring to the wireless world the much awaited security. WPA solves the problems of WEP by providing confidentiality, integrity, and authentication, three of the most important security elements as explained in Section 2.1. WPA addresses the security needs of small offices and homes (SOHO) as well as large enterprises: WPA-PSK (Pre-Shared

Key) is available for SOHOs with limited resources where simple methods with as little maintenance as possible are preferred, and WPA using 802.1x and TKIP for large enterprises with enough resources.

For small offices and homes, a WPA with Pre-Shared Key (PSK) is recommended. The method consists in manually entering a passphrase on all the access points and clients that are part of the small network. A description of the method is found in [13] and [48]. The Pre-Shred Key can be a 256-bit number or a passphrase 8 to 63 bytes long. The PSK is not sent with the packets nor is it directly used in the encryption, but using it as Pairwise Master Key (PMK) is derived to be later used in obtaining a Pairwise Transient Key (PTK) then used in encrypting each packet. If the PSK is a 256-bit number, then it is used as the PMK. If a passphrase is used as the PSK, then the PMK is obtained as follows:

$$\text{PMK} = \text{PBKDF2}(\text{passphrase, ssid, ssidlength}, 4096, 256) \tag{3.1}$$

where the PBKDF2 method is from the Password Based Cryptography Standard. In other terms, this means that the passphrase, SSID, and length of the SSID are concatenated then hashed 4096 times to finally obtain a 256 bit number, the PMK. Therefore, the whole process depends on the secrecy of the passphrase, since the rest of the elements, SSID and hashing method, are known. In [48], Moskowitz explains a weakness in the PSK procedure where if a PSK passphrase is chosen to be less than 20 characters long, it could be recovered using a dictionary attack. Therefore, as prevention, it is advisable to either use a long passphrase composed of several words and numbers, or to generate a random 256-bit number for use as PSK. It is also important to point out that the PSK is a secret shared among many users, which means that it can be compromised fairly easy. However, in small offices or homes the number of users is relatively small and the environment is more or less controlled.

For enterprises, WPA takes a totally different approach and the security is further enforced. The process is divided into two stages: a secure authentication process us-

ing 802.1x and encryption using the Temporal Key Integrity Protocol (TKIP) which also included the Message Integrity Code (Michael) that ensures that the message has not been altered. The 802.1x authentication process follows the process explained in Section 3.3 using one of the EAP methods, a choice left to the administrator. Using 802.1x provides enhanced authentication where the client and the access point have to authenticate each other to check for the legitimacy of the other party, hence solving the problem of potential rogue access points present in networks employing WEP. Once the authentication is successfully completed, a master TKIP key is exchanged between the client and the access point, and a four-way handshake in which the keys are installed on both machines completes the process [13]. At this point, the regular communication between the client and the access point can begin where every packet exchanged is encrypted using TKIP with a unique key, a feature completely missing from WEP, where all packets are encrypted with the same key. The TKIP encryption is the intermediate replacement for WEP, and the upcoming 802.11i uses it as a stepping stone towards more robust solutions ([53], [15], [13], [19]). TKIP consists of three main features: a per-packet mixing function, an improved Message Integrity Check (MIC) called Michael and an enhanced Initialization Vector (IV) scheme. In TKIP, the client starts with two keys: a 128-bit encryption "Temporal Key" (TK) and a 64-bit Integrity key obtained during the 802.1x negotiations. The TKIP encryption mechanism, illustrated in Figure 3.8 starts by performing the XOR operation between the MAC address of the client and the 128-bit Temporal Key to obtain a Phase I Intermediate Key. This key is then mixed with a sequence number, a 48-bit Initialization Vector, to produce Phase II key. The preceding steps ensure that the keys are unique to the user since the client MAC address is used to generate them and that each packet has its own key because of the sequence number. The Phase II key is then handed to the RC4 algorithm to compute a keystream which will be XORed with the plaintext to obtain the ciphertext to be transmitted. The plaintext does not consist of the message only, but the message along with the Message

Integrity Code (MIC). Michael, i.e. MIC, uses a protected one-way hash function using the MIC key, the source address, destination address and the plaintext to produce an 8-byte hash that will be appended to the message. Since the integrity check is no longer a linear function as it was the case in WEP's 32-bit CRC, a modification in the packet is directly detected and the packet will be dropped.

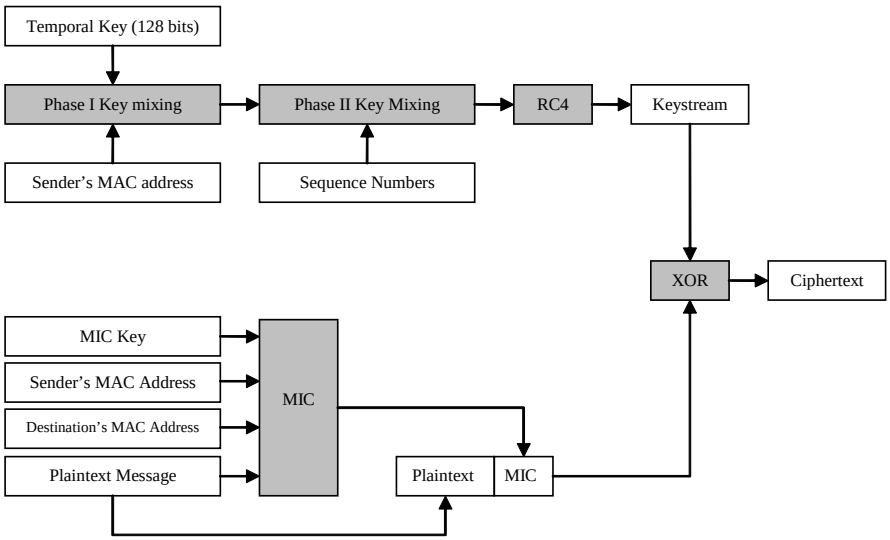

Figure 3.8: TKIP Encryption Process

Thus, TKIP eliminates the potential for replay attacks, forgery attacks resulting in tampering with the messages, and weak key attacks which were feasible when WEP was used. Table 3.2 shows a comparison between WEP and WPA. However, WPA by itself is not a standard but a mere preparation to the 802.11i standard that will drive wireless security down a more robust path.

	WEP	**WPA**
Encryption	Flawed, Cracked	Fixed all WEP Flaws
	40-bit keys (standard) 104-bit keys (manufacturers)	128-bit keys
	Static – same key used by everyone on network	Dynamic session keys, per user, per session, per packet keys
	Manual Distribution of Keys on all devices	Automatic distribution of keys
Authentication	Flawed- WEP key used for authentication	Strong user authentication using 802.1x
	One-way authentication	Mutual Authentication

Table 3.2: Comparison between WEP and WPA

3.5 IEEE 802.11i

The security provided by the Wi-Fi Protected Access (WPA) (Section 3.4) was an interim solution provided by the Wi-Fi Alliance to provide an immediate alternative to WEP, whose vulnerabilities have made wireless networks unsafe. WPA was in fact a subset of the 802.11i standard being drafted by the IEEE and set for official release before the end of the year 2004. 802.11i, also known as WPA2, promises to solve all the securities and concerns of the wireless communities by providing strong basis for authentication, confidentially and integrity of the data. 802.11i relies on the Robust Security Network (RSN) that is composed of an authentication mechanism, 802.1x, and encryption using the Advanced Encryption Standard (AES) which is a Federal Information Processing Standard (FIPS) compliant standard. The drawback of 802.11i is that it is not backwards compatible with first generation wireless equipment due to the need for high processing power and support for intensive encryption algorithm as required by AES. Therefore, the transition to 802.11i may prove to be costly and may require some time ahead for planning.

802.11i (see [19], [53], [15], [38], [55], [45], [32], and [26]) consists of two main processes: authentication and encryption. The authentication level is secured by the 802.1x standard (Section 3.3) along with any of the EAP methods. Once the

authentication is successful, a set of keys that will be employed in encryption is derived ([26]). Figure 3.9 shows the hierarchy of keys used in the process.

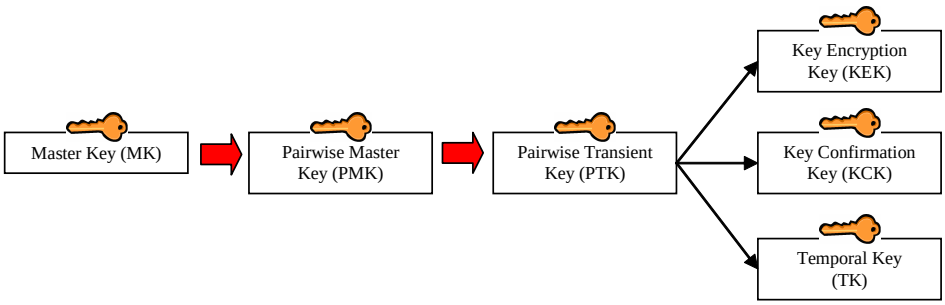

Figure 3.9: Pairwise Key Hierarchy ([26])

First, a Master Key (MK) is provided for the supplicant and the authentication server (usually a RADIUS server); this key will be on a per-session basis and is an indication of a positive access decision. The MK is then used to derive the Pairwise Master Key (PMK), which is forwarded by the authentication server to the access point to authorize the supplicant to utilize the 802.11 medium. From the PMK, a set of Pairwise Transient Keys (PTK) are generated: the Key Confirmation Key (KCK) is used as evidence of the possession of the PMK; the Key Encryption Key (KEK) is used to distribute the encryption keys to members of the session and finally the Temporal Key (TK) for encryption of the data.

The 802.11i standard specifies AES as encryption standard with the possibility of using TKIP (explained in Section 3.4). The Advanced Encryption Standard (AES) is the result of years of research and is designed to replace the RC4 cipher. AES is a symmetric encryption algorithm, meaning that the same key is used for both encryption and decryption of the cipher text. Differing from RC4 which operates on a per-bit basis, AES is a block cipher that deals with 128-bit chunks of data at a time. The AES encryption specified in the 802.11i standard is the counter mode with CBC-MAC (Cipher Block Chaining- Message Authentication Code) also known

as the counter mode-CCMP protocol. The counter mode is used for confidentiality and the CCMP for ensuring integrity and authentication. The use of counter-mode CCMP is mandatory for 802.11i compliance.

The first step towards security is to guarantee that the message received is free of any form of alteration to its content, i.e. its integrity is preserved. This is ensured by the CBC-MAC operation that calculates a Message Integrity Code (MIC). The CCMP uses a 48-bit Initialization Vector (IV) called Packet Number (PN) in its operation. In fact, the MIC calculation is seeded with the IV formed by a flag value, the Packet Number (PN) as well as other information extracted from the frame header of the message to be transmitted. The IV is fed to an AES block and the output is XORed with elements from the frame header and then fed into the next AES block. This process continues until all blocks are exhausted and a final CBC-MAC value of 128 bits is computed. The upper 64 bits of this value are extracted and placed aside to be later used in the computation of the MIC that will be appended to the final frame.

The encryption process of the data in the message is initiated by a counter preloaded into an AES cipher block and whose output is XORed with 128 bits of the plaintext. Subsequently, the counter is incremented and the process repeated 128 bits at a time until all the plaintext has been processed. Next, the counter is reset to 0 and inputted to an AES block, and the output is XORed with the CBC-MAC value computed above to get the MIC which is then appended to the end of the encrypted frame in preparation for transmission. This procedure is illustrated in Figure 3.10 taken from [32]. The decryption process consists in following the reverse path to recover the plaintext message and then comparing the MIC value computed to check the integrity of the message.

802.11i may provide the "perfect" security solution, at least for the time being, but it does not come cheap. 802.11i's implementation is complex and may not op-

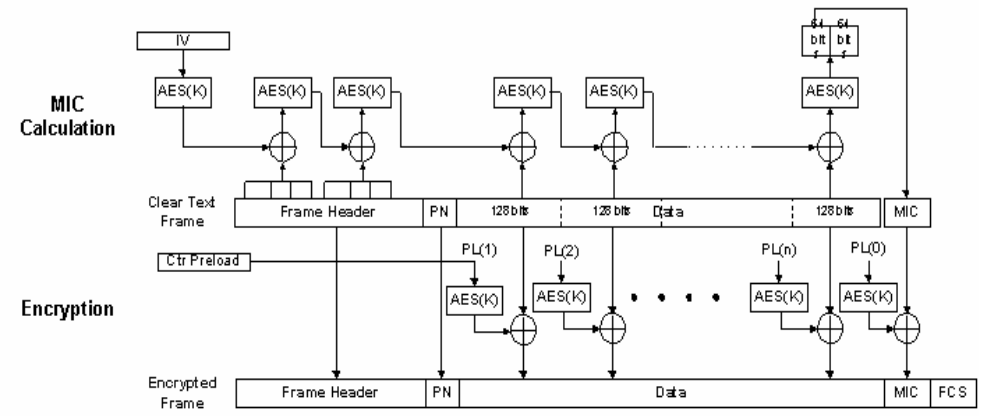

Figure 3.10: The AES Encryption Process ([32])

erate with currently available wireless access points and NIC cards, due to the high processing power required by the AES encryption process. However, one might consider such cost acceptable when compared to the wireless security products available in the market today, which will be discussed in Chapter 4, to ensure that his network is adequately secured. 802.11i has not been standardized yet, but is expected to be before the end of the year 2004. As a summary, Table 3.3 compares the different security alternatives present today: WEP, TKIP (WPA) and CCMP.

	WEP	**TKIP**	**CCMP**
Cipher	RC4	RC4	AES
Key Size	40 or 104 bits	128 bits encryption 64 bits authentication	128 bits
Key Life	24-bit IV	48-bit IV	48-bit IV
Packet Key	Concatenation	Mixing Function	Not Needed
Integrity			
Data	CRC-32	Michael	CCM
Header	None	Michael	CCM
Replay	None	Use IV	Use IV
Key Management	None	EAP-based	EAP-based

Table 3.3: Comparing WEP, TKIP and CCMP

3.6 Virtual Private Networks (VPN)

During the days when WEP was the mostly used security measure, some administrators turned to Virtual Private Networks as a way to secure their network and control access to it. A Virtual Private Network (VPN) is a method used to extend the resources of private networks across an un-trusted medium (usually the Internet). In a world of WEP vulnerabilities, VPNs allow the reliance on higher level protocols to secure data. VPNs are generally used on wired networks and their applications and types vary, however, in this section, we will limit our discussion to their use in wireless environment and will refrain from discussing their detailed operation.

When dealing with Virtual Private Networks in a wireless context, two main methods are possible ([19], [27]): Point-to-Point Tunneling Protocol (PPTP) and Layer 2 Tunneling Protocol (L2TP) with Internet Protocol Security (IPSec). It is important to point out that the two methods can be used simultaneously on the same system.

PPTP or Point-to-Point Tunneling provides both authentication and encryption for its users. The authentication can be using MS-CHAP and MS-CHAP v2, thus a username/ password passed authentication or using EAP-TLS which requires both sides to use certificates from a Certificate Authority (CA). Details about EAP-TLS can be found in Section 3.3. However, when using MS-CHAP, it is imperative to employ strong passwords to avoid any type of dictionary attack. The encryption in PPTP relies in Microsoft Point-to-Point Encryption (MPE) which a stream cipher based on the RC4 algorithm. The encryption process begins right after the PPP authentication and link establishment.

Another alternative would be the Layer 2 Tunneling Protocol with IPSec. This method provides a per-packet guarantee of authenticity and integrity. L2TP uses the Point-to-Point Protocol (PPP) for authentication and the Internet Protocol Security

(IPSec) for data encryption using the Data Encryption Standard (DES) with 56-bit keys or the 3DES with 168-bit keys. This method relies on the Public Key Infrastructure (PKI) and requires authentication from both sides in addition to credentials from the user. The fist step in setting up L2TP is to create an IPSec Security Association (SA) by establishing the Internet Key Exchange (IKE) negotiations. During that phase, the client and the server exchange certificates and decide on the authentication methods and encryption keys that will be used for communication. Once this is completed, the tunnel is ready for message exchange.

If one were to compare the two methods, the first point to observe is the fact that PPTP is usually supported on more clients than L2TP which requires additional equipment and software since IPSec is not available on most devices. Second, L2TP requires a Public Key Infrastructure (PKI) which may create additional hassle that an administrator would rather avoid, PPTP could also be cumbersome if EAP-TLS is used, imposing the need of a Certificate Authority. On the other hand, L2TP along with IPSec provide better security since they offer authentication, integrity and confidentiality, whereas PPTP provides confidentiality and authentication only. The two methods, PPTP and L2TP impact performance and usability by adding an overhead to each packet. Furthermore, neither of the methods provides for roaming; the user is never able to move around from one access point to another which means that VPN are depriving the users from one key feature of wireless networks: mobility.

3.7 Summary

As can be inferred from this chapter, there exist several methods to secure a wireless network. Whether the requirement is to establish a hot spot with no real security, or to tightly secure the exchange of sensitive information using VPN tunnels, solutions are diverse and the level of protection they provide varies greatly: WEP has proven

to be flawed and not suitable for deployment where security is a requirement and the IEEE 802.11i standard has yet to be ratified, leaving WPA as a temporary solution for administrators who seek privacy of their users. When paired with the task of managing the entire network, maintaining networks and ensuring their security might be too much for an administrator to handle. To render things simple, several solutions have been proposed for the sole purpose of securing and managing wireless networks as will be presented in Chapter 4.

Chapter 4

Wireless Security Products

Securing wireless networks has become a huge issue, and while WEP has proven to be incapable of providing a decent level of security for wireless networks, 802.11i is yet to be standardized. WPA does provide a good level of protection, but is not yet popular. Nevertheless, none of these solutions offer management of the wireless network's components, an essential feature for large networks involving thousands of users. To address these issues, several products have been released on the market promising to render wireless networks impenetrable and to provide easy and efficient management of the wireless devices. In this chapter, we overview three products that aim to protect wireless environments: Roving Planet, Bluesocket and Cranite. Roving Planet is the only product that we actually tested in our laboratory, and the results are included in the first section.

4.1 Roving Planet Solution

In this section, we present Roving Planet, a hardware solution that helps control and manage large-scaled wireless networks.

The main objective of our testing was to investigate whether Roving Planet satisfies the requirements set forth by the University of New Mexico to ensure that wireless environments on campus are secure and controlled. It is also of importance for the University to be able to effectively control all the access points that belong to its network and manage them easily, in addition to detecting rogue access points that could jeopardize the whole network.

The main focus of our testing is the security aspect of Roving Planet and its control over the wireless Network, although Roving Planet encompasses many additional features such as management of access points and users as well as traffic and bandwidth management. The following sections contain the initial results and many of these testing issues that we encountered, as we were assured by Roving Planet, will be either solved in the upcoming versions of the solution or will be addressed soon.

4.1.1 Roving Planet Overview

The Roving Planet solution, [10], is designed as an OSI Layer 2 Bridge/pass-through, enabling simplified installation and more flexible deployment options. It is a hardware solution composed of 2 main parts: the CSD Engine and the CSD Agent. The CSD engine is the central management service. It maintains all event and usage data and controls and communicates with the CSD agents to deploy policies that manage network traffic and control the behavior of the users and applications. The CSD engine can be anywhere in the network and not necessarily in the traffic's path. On the other hand, the CSD agent is the direct controller of the access points. The agent enforces access and bandwidth policies, monitors traffic and reports all usages to the CSD engine. The agent should always be located in the data path, following the access points. Both engine and agent run on standard Intel-based Linux servers.

Roving Planet is managed through a secure web-based browser interface: the CSD Control Console. The control console offers tools to manage the entire network including users, policies, applications, access points, bandwidth management policies in addition to real time monitoring of WLAN activity and reporting functionality.

Roving Planet is highly compatible with CISCO products and relies on them for insuring confidentiality of the data. In other terms, Roving Planet does not offer any encryption of the data, but leaves the choice to the administrator to impose the level of security that is best suited for his network: WEP, WPA, VPNs, etc. It does however offer authentication in addition to management of the network.

Two authentication methods are possible: web-based authentication or authentication through secure tokens or 802.1x. The web-based authentication is a secure http (https) web page that prompts the user to enter his credentials. This web page is the first thing the user sees when he first associates with the access point. The user cannot surf the internet nor initiate any operation that requires Internet connection until he correctly enters a username and password. There are 2 types of web portals: one for regular computers and laptops, which is the usual web page and the others specifically designed to fit PDAs. The second type of authentication supported is 802.1x or authentication via certificates. The protocols supported as indicated by the Roving Planet documentation include EAP-TTLS, EAP-MD5, EAP-PEAP, EAP-LEAP. For testing results of the different authentication methods, refer to Section 4.1.2. Roving Planet is inter-operable with several authentication servers including LDAP, RADIUS, Microsoft Active Directory in addition to the local RADIUS running on the engine machine.

The most important feature of Roving Planet is its ability to provide Quality of Service for Wireless Networks and Role-based bandwidth management. In fact, users are gathered in groups locally, or as specified by the organizational hierarchy set in RADIUS or LDAP, and each group is assigned a priority and a certain fraction of the

bandwidth. This way, the amount assigned to each group is not fixed but changes depending on how much total bandwidth is provided. Another advantage of Roving Planet would be its ability to accommodate guests and visitors in the network and providing them with as little or as much privileges as is needed.

Applications are also controlled by Roving Planet: not every application can be run and not everyone on the network can run the same applications. Actually, applications are defined and access policies are set so that only authorized groups of users can use them and everyone else is denied access. Applications are defined by the port and protocol (TCP, UDP, ICMP) they use. It is also possible to set schedules by time, application, access point or user that directly run without requiring the presence of the administrator. These scheduling policies are referred to as "Modes".

Roving Planet offers Real-Time monitoring of all activities, in addition to the possibilities to generate reports using the CSD Reports Console available in the accompanying CD. Reports can be customized and the administrator is able to view usage per application, user group, access point, or simply the entire logs for a given period of time. The reports can also be exported and saved for future references.

4.1.2 Putting Roving Planet to the Test

In this section, we present the different tests that we conducted and the results of each. The version of Roving Planet that we evaluated was the RovingPlanet Central Site Director (CSD) Version 2.5.

For testing purposes, we set up a small network with 3 access points running 802.11b. In order to have diversity in our network, we used 2 CISCO Aironet 1200 series access points, one running VxWorks and the other IOS. The third access point used is an HP ProCurve Networking Wireless Access Point 420 running the latest firmware. The access points are connected to an HP ProCurve Switch 2524. Figure

4.1 illustrates the test network.

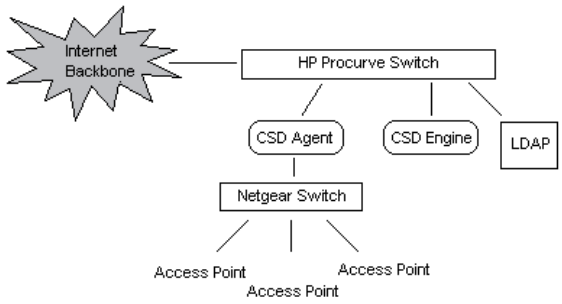

Figure 4.1: Topology of the Testing Network

The first issue we encountered was with the installation: Roving Planet's CSD agent operates as a Layer 2 bridge, thus our initial approachwas to set up the network as shown in Figure 4.2 with the access points connected to one VLAN and the agent acting between that VLAN and the VLAN corresponding to the rest of the network. However, this setting was not operational: no traffic was forwarded due to corrupted ARP tables, there was no mapping between IP addresses and MAC addresses and the network's components could not be recognized. The first thought as what caused this problem was the switch. However, the HP switch that we used supports multiple VLANs and the corresponding configuration was correct, but the problem persisted. The reason for such behavior could not be identified. But since the main objective of our testing was the security aspect, we thought that it was best to modify our setup network and omit the use of VLANs.

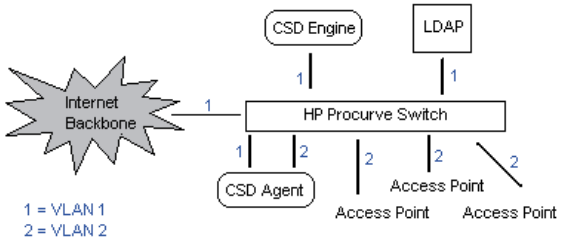

Figure 4.2: First Setup of the Test Network using VLANs

Another setback was that the original access points we had installed were Linksys WAP54G (802.11g). Roving Planet engine relies on SNMP traffic to identify the access points in the network and the users connected to each one. The Linksys access points have a very basic support for SNMP and it is not enough to satisfy Roving Planet's requirements. Nevertheless, Roving Planet will still block traffic until the user authenticates, but the monitoring function will not be functional and all users will go unmonitored. The problems started emerging when we added the Linksys APs to the list maintained by the Agent: excessive traffic was exchanged between the access points and the agent and errors were continuously generated filling up the log tables fast. At one time, the log tables were filled up and caused the CSD engine to become totally irresponsive and the only solution was to reboot the engine. Therefore, to avoid such hassle, it is best to use access points as specified by the Roving Planet documentation (mainly CISCO, Orinoco AP-1000 and 2000, Avaya AP-2 and 3, etc).

One of the issues we considered critical is the termination of a session. In the older version of the agent, the session timeout was the responsibility of the access point the user is associated with. However, our HP AP-2000 access point did not support this feature and hence a user's session was never terminated because Roving Planet did not have any policy regarding session timeout. In the newer version of Roving Planet, the agent can impose a timeout time, during which if the user remains idle, the session is directly terminated. In addition to the time-out issue, the log-off is also worth a mention: there is a way for the user to log-off but it is not intuitive and not easily found by a regular user. When a user first launches a browser, he is prompted to enter his credentials and then authenticated. This same page is used for the log-out. The problem is that the user is directly redirected to the page requested and the log-out page is lost. This is not a major issue since the timeout function is now functioning.

The major part of our testing revolved around the authentication, namely using 802.1x. The Roving Planet specifications identify various suported authentication methods, we tried to test as many of them as possible to check how efficient and simple it is to authenticate. As database servers, we used LDAP (OpenLDAP) with organizational unit hierarchy in addition to the local database, embedded in Roving Planet. Windows Active Directory was not available to us, and thus was not tested. The information concerning Windows AD was provided by Roving Planet Documentation.
The 3 authentication methods that we used were the EAP-TTLS, EAP-PEAP, and EAP-LEAP.

- **EAP-TTLS**: Windows Wireless Client does not support TTLS, so in order to test the EAP-TTLS capabilities, we used a free client software "W2Secure" available online by *Alpha & Ariss*. When tested with the local database, we could authenticate with both PAP and EAP-MD5 challenge as inner protocols. LDAP authentication was also successful.

- **EAP-PEAP**: According to our testings, with the current version of the engine and agent, EAP-PEAP is not supported, but it should be available in future releases. The built-in Windows Wireless Client was used for the testing.

- **EAP-LEAP**:Authentication through EAP-LEAP is possible when the local database is used or RADIUS. With LDAP, the only way to authenticate is when the user password in LDAP is stored in clear text which is not advisable. This test was conducted on CISCO access point and Wireless Client Adapter, since LEAP is a CISCO-specific standard.

According to Roving Planet, future releases of the engine and agent will support more authentication methods and additional compatibility with authentication servers.

Table 4.1 summarizes the authentication methods currently existing and those to be expected soon.

		Authentication Protocols		
		EAP-TTLS	EAP-PEAP	EAP-LEAP
Authentication Servers	Local DB	★	↷	★
	RADIUS	↷	↷	★
	Windows AD	★	↷	
	LDAP	★		

Table 4.1: Authentication methods and Authentication servers currently available (Star) and those to be expected in Future Releases (Arrow)

In Section 4.1.1, we mentioned that one can restrict what applications can be run, by whom, and at what time. Our testing has showed that only applications that are defined explicitly by the administrator can be controlled, all other applications are available to any user that authenticates, regardless of the group of users she/he belongs to. In other terms, applications are not disabled by default. It is however important to note that an unauthenticated user does not have any kind of access and Roving Planet still blocks all traffic until she/he correctly identifies herself/himself to the network. For example, if the application *Telnet* is not explicitly defined, all authenticated users can use it, but if it is added and policies regarding who is allowed to run it are set, then only that group is given access to it, and all others denied access.

One other issue worth mentioning, although very simple and straight forward, is the fact that both engine and agent are vulnerable to PING of death attacks which are harmful to the network as a whole. The solution is to simply shut-down the ping port.

Our testing of the authentication methods have led us inside the engine itself

to monitor its operation closely. It is there that we found a considerable vulnerability that if exploited, the authentication process is easily bypassed. When a user authenticates through a RADIUS (i.e. using 802.1x) or any authentication server, a command is sent to the engine informing him about the successful addition of a new user. This command is a simple http request sent in clear text between the local RADIUS running on the engine machine and engine code itself. Therefore, a duplication of this command, even with fictional username and password can get the user full access. The command can be sent through a browser, where the user can authenticate any MAC address. This command will be interpreted by the engine code as a confirmation of authentication coming from a legal authentication server. This security hole can be prevented by setting up a secured tunnel through which the engine and Radius can communicate or simply encrypting the command in question. We understand that getting hold of this command requires access to the internal system of the engine machine, but, nevertheless, if anyone makes it public, then the whole security of authentication provided by Roving Planet is broken.
This issue however was resolved in the next release of Roving Planet version 2.6. We tried to use the same command as before to gain access to the network without authenticating, but the engine blocked our traffic and did not let us through.

Roving Planet promises the control and management of large-scale wireless Networks. Our objective at the University of New Mexico is to find a solution that is simple yet robust, one that secures a campus wide network while leaving possibility of allowing guest accesses. Roving Planet does satisfy requirements set forth by the university. Table 4.2 summarizes the features of Roving Planet that are of direct interest to our security and management needs.

Requirements	Roving Planet
Authentication	
LDAP Authentication	Supported and Tested
Windows Active Directory	Supported
Secure Tokens Authentication	Supported and Tested (802.1x methods)
Encryption	
3DES, AES encryption	Not provided, but RP supports several encryption standards
Non-proprietary client software for encryption	No client required, unless specific encryption is implemented
Management	
Guest Access	Yes and Controlled
Web-based management	Available via the CSD Console
Bandwidth management	Yes
Traffic prioritization	Yes
Hot Failover	Yes when multiple agents are used (Not tested)
DHCP	Yes
NAT	Yes
"Easy Implementation"	Fairly easy
"Easy Management"	Very simple

Table 4.2: Requirements set forth by the University of New Mexico and solutions offered by Roving Planet

4.2 Bluesocket

Another popular wireless security product on the market today is Bluesocket. Although we have not tested the product in our laboratory, nor did we have direct contact with it, the documentation available on the Bluesocket website ([2]) as well as online evaluation reports ([51]), despite being two years old, provided enough information to formulate an idea of how the product works.

Bluesocket is a hardware solution inserted between the wired network and the access points as illustrated in Figure 4.3. All traffic is intercepted by the box and only authorized users will be allowed access.

Bluesocket is based on Layer 3 security, where "layer" refers to one of the seven layers of the Open Systems Interconnection(OSI) network suite and layer 3 to the network layer, responsible for the opening and maintenance of paths between two

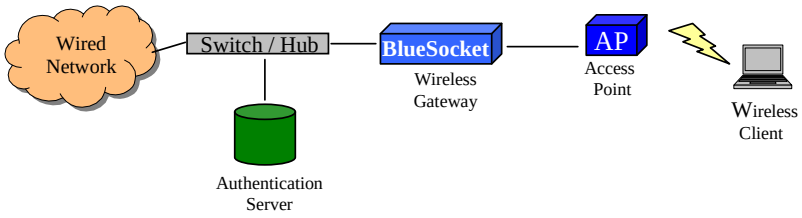

Figure 4.3: Typical Bluesocket setup

systems. Bluesocket acts as a specialized VPN/ Firewall that uses role-based policies to simplify control of the wireless network.

Bluesocket provides several solutions called wireless gateways, depending on the size of the enterprise. For a small office or floor offices, the WG-1100 is suggested (WG-1100 SOE for small office edition). For medium to large enterprises, the WG-2100 is recommended and for large deployments involving thousands of users, the WG-5000 series is provided. The differences between these solutions is how much throughput they are able to handle, and their respective network interface. Table 4.3 compares the different solutions available from Bluesocket.

	WG 1100/ WG 1100 (SOE)	**WG 2100**	**WG 5000**
Enterprise size	Small	Medium to Large	Large
Throughput			
1) Encrypted	30 Mbps	150 Mbps	400 Mbps
2) Unencrypted	100 Mbps	400 Mbps	1 Gbps
Network Interface	2 10/100 Mbps	2 10/100/1000 Mbps	2 10/100/1000 Mbps
Fail Over Network Interface	10/100 Mpbs	10/100 Mbps	10/100/1000 Mbps
Size of Hardware	1 Rack Unit	2 Rack Units	2 Rack Units

Table 4.3: Different Bluesocket solutions

The product provides two alternatives for wireless usage; encrypted and unencrytped sessions. For guests and regular users, the unencrypted sessions are used to avoid forcing a temporary user into using VPN tunnels, whereas to provide a strong

encryption, it is imperative to establish VPN tunnels. Bluesocket claims to be a non-proprietary solution; any VPN client can be used to setup the tunnel. According to [51], Bluesocket was successfully tested using Microsoft PPTP clients on all Windows Operating Systems, Microsoft IPSec clients on Windows XP and 2000, Safenet SoftRemote Win32 OEMs, SSH Communications Sentinel Win32 IPSec client as well as Certicom movian VPN and unk AdmintOne IPSec clients on PocketPC. This list is relatively old, dating from 2002, which means that it is most probably outdated. The list should have been extended by now to include additional clients and support for supplementary operating systems. Encryption is provided by any of the common algorithms including DES, 3DES and AES.

The authentication process is completed using either credentials (username/ password pair) for unencrypted sessions, or certificates for enhanced security via encryption. When there is need for guest access, the process of getting access to the network is relinquished to a secure web page, where the user is prompted to enter his username and password. Bluesocket supports several authentication servers including RADIUS, LDAP, NT Domain servers, Windows Active Directory, in addition to the local database embedded in the product itself. Bluesocket also supports the 802.1x authentication methods explained in Section 3.3.

When Bluesocket is deployed in large enterprise networks, several wireless gateways are required, and to allow for the possibility of fail-over in case any of them crashes, Bluesocket uses a mesh arrangement of the boxes, where one box is labelled the master and all other configured as slaves. When the master goes down, the slaves continue their regular tasks and take over the master's, however, no policy update can occur until the master has recovered or a new one elected. This setup proves to be very helpful in large environments where at any point a wireless gateway can give away, and clients could lose their connections.

Taking into account the deployment of several wireless gateways, Bluesocket

brings to VPN tunnels the added element of mobility through their "Secure Mobility" feature included in software version 2.0 or later, which allows users to roam across subnets and access points without needing to re-authenticate even if they're using IPSec tunnelling (refer to Section 3.6). This mobility feature assures no degradation of performance. Nevertheless, some issues need to be considered: laptops with personal firewalls cannot roam uninterrupted unless they are configured to trust the wireless gateway. Moreover, Microsoft Windows XP users must update their windows registry to disable "Media Sense" in order to keep the same IP address even if the connection with the access point is lost for few seconds.

The administration of Bluesocket can be conducted using one of three possible interfaces: the SSL-protected GUI interface, SSH v2 protected CLI using serial connections (in case there is need to access the actual box for some reason), and an SNMP for integration with third party systems. Nonetheless, the easiest and simplest form is through the web portal. The Quality of Service (QoS) and bandwidth management are administered through the interfaces. In fact, Bluesocket operates on basis of roles which can be organizational units, job functions, or groups of people. Permissions granted to roles are enforced on all user members of that group. Each group can be assigned a maximum bandwidth and specific rights. Bluesocket groups are not static, and roles are hierarchical , enabling uniform enforcement of basic policies, refined for certain users or groups. Like most firewalls, that which is not explicitly permitted is denied, with one exception: ICMP is permitted unless explicitly denied which according to [51] is used to facilitate debugging but deserves to be mentioned.

The monitoring of wireless activity in Bluesocket is also done via the interface: each user on the network is logged along the time he logged in, what group he belongs to and how much bandwidth he is currently occupying. Information about the system name, VPN type and tunnel status are also included. The only drawback is that

each wireless gateway is monitored independently. The solution to this issue consists in sending each wireless gateway's data to a central log: the SYSLOG server.

Bluesocket's approach to wireless security may be convenient to networks that want enhanced security of the data of its members through VPN tunnels but prefer less cumbersome procedures for their guests. The only issue is that the unencrypted data may be sniffed, and information may be revealed to hackers. Still, the Bluesocket solution has been popular ever since its deployment, since it also provides an IPSec client ready to install that requires very little work from the client. However, this client is for Windows 2000/ XP clients, where the IPSec client already exists but needs configuration, which is taken care of by this downloaded piece of software. The setup may prove to be more complex for clients running other operating systems. Table 4.4 summarizes the different features of Bluesocket.

Requirements	**Bluesocket**
Authentication	
LDAP Authentication	Supported
Windows Active Directory	Supported
Secure Tokens Authentication	Supported via 802.1x
Encryption	
3DES, AES encryption	Yes, both of them supported
Non-proprietary client software for encryption	- Yes, but VPN client needed (Section 4.2) - Possibility for web login only if no encryption needed
Management	
Guest Access	Yes, via web login
Web-based management	Yes
Bandwidth management	Yes
Traffic prioritization	Yes
Hot Failover	Yes, when several Wireless Gateway set in master/slave mode
DHCP	Yes
NAT	Yes
"Easy Implementation"	NOT TESTED
"Easy Management"	NOT TESTED

Table 4.4: Requirements set forth by the University of New Mexico and solutions offered by BlueSocket

4.3 Cranite

Cranite is a solution that promises to provide wireless networks with management, security, and mobility. It is an award-winning solution that prides itself on being certified by the National Institute of standards and Technology (NIST) as meeting the Federal Information Processing Standard (FIPS) 140-2. We did not have the opportunity to test Cranite in our labs, however we were fortunate to get a quick overview of the product, courtesy of Canberra Aquila Inc [3] and most of the information included in this Section was retrieved from the product's documentation at [5].

Cranite offers a software security solution, meaning that the company provides only the software not the hardware; a machine running Linux is usually required for the install. Cranite relies on Layer 2 to provide security of data transmission and secure mobility. Layer 2 refers to the data link layer, in other words, the layer that defines rules for sending and receiving information, encoding and framing data, and detecting and controlling errors. The advantage of using layer 2 security over layer 3 as is the case with Bluesocket (Section 4.2, is the fact that the latter requires the network to run completely open to any potential station which means that IP addresses can be obtained fairly easy. Thus, Cranite offers to hide both the data and the network.

Cranite offers its clients two editions depending on their needs. The Cranite SMB Edition is designed for small to medium businesses accommodating 10 to 250 simultaneous users. For large networks spanning across several subnets, Cranite offers its enterprise edition, which also allows for secure roaming across the subnets.

The Cranite solution comprises three components: the WirelessWall Manager, WirelessWall Access Controller and the WirelessWall Client in a setup similar to the one shown in Figure 4.4 .

WirelessWall Manager This constitutes the browser-based centralized application through which the management, configuration and monitoring is performed.

WirelessWall Access Controller This is the gatekeeper of the network. It enforces the policies created on the manager and performs all session management tasks such as encryption and decryption, firewall filtering and secure mobility. More than one access controller can be deployed in a wireless environment is there is need to.

WirelessWall Client This is a zero-configuration thin client that is mandatory for all stations that need to connect to the wireless network and take advantage of the security provided by Cranite. The client provides a simple interface for login and works with the controller to encrypt and decrypt traffic.

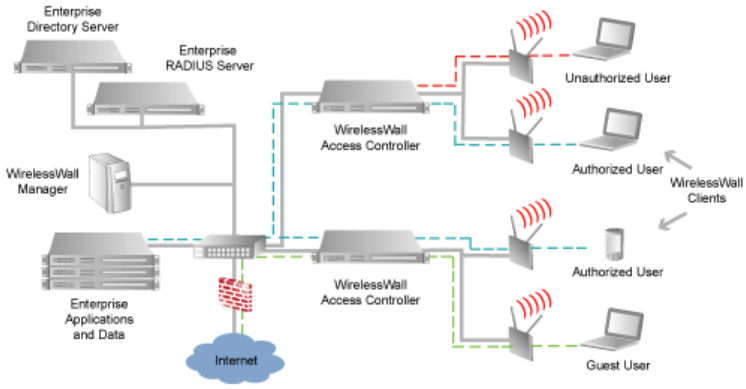

Figure 4.4: Typical Cranite setup showing the different components ([5])

It is to be noted that for the SMB edition of Cranite, the WirelessWall manager and access controller are installed on the same machine whereas for the enterprise edition each is installed on a different machine since there could be several Access Controllers all administered by the same Manager.

The regular user can login through the client provided by Cranite, the authentication is achieved using EAP-TTLS (Section 3.3), to protect against capture of credentials by an attacker. The client provided by Cranite is compatible with all versions of Microsoft Windows operating systems, Pocket PC 2002, OSX.2 (Jaguar) and Linux Red Hat 7.8. For users with legacy Mac clients, non-encrypted but authorized guest access can be provided. Cranite supports most authentication servers; Windows Active Directory and NT domain server are automatically incorporated, while the use of LDAP requires minor schema integration. The security provided by Cranite is ensured by authenticating every frame exchanged and encrypting full ethernet frames and not just user data using the 128-bit Advanced Encryption Standard (AES). Unique session and message integrity keys are used for every client. To control its connections, Cranite keeps track of all its users and sets two session timeouts. The session length timeout defines the time users are allowed on the wireless network, it is usually set to the duration of the workday. The idle timeout, as its name indicates, defines the time a session stays alive after the absence of all user activity. In both cases, the user is prompted to enter his credentials just before timeout occurs in order to keep the same session alive, or else the connection is dropped and the session has to be re-established. As part of its failover mechanism, the WirelessWall manager is not required for ongoing sessions, this means that if the manager crashes, only creation of new sessions is affected and no new users can join the wireless network until the manager is up and running. Cranite also takes into consideration the possibility of WirelessWall Access controllers to fail, that's why it was designed to optimize system behavior when such incident occurs. Actually, to prevent any failures in the access controller, the administrator can "deploy optional redundant servers to act as hot standby backups for primary servers", without losing any of the connections.

Cranite prides itself with its robust mobility support; it promises seamless uninterrupted sessions for users as they move between subnets. Cranite offers three

options for roaming depending on the entreprise's needs:

Mode 1: Dynamic Home Mode This is also known as the maximum security mode. Each policy is associated with a home subnet; when the client roams to a new subnet, the roamed WirelessWall informs the home WirelessWall of its IP address and an encrypted tunnel is set between the two WirelessWalls to forward any traffic to the client securely. This mode ensures application persistence using a low-latency process with low overhead.

Mode 2: Static Home Mode This mode provides efficient mobility, where the each session is associated with a home subnet. Once the client, roams to another subnet, all traffic is delivered to the designated home WirelessWall, and then forwarded to the new location of the user. This mode provides in addition to application persistence an additional level of management control.

Mode 3: No Home Mode This mode assures that the network traffic is minimized, it also assumes that application persistence is not essential. When the client roams to a new subnet, new sessions keys are derived and the user is provided with a new IP address by the DHCP. Consequently, any application dependent on the maintenance of the IP address must be restarted.

Cranite, in addition to providing tight security for the wireless environment, offers management features. Role-based policies can be defined and real-time monitoring of the connections is also provided via the WirelessWall Manager. Guest access is also possible, however, if the data needs encryption, as it probably will, the temporary user will have to download and install the WirelessWall Client to be granted access. This may present an inconvenience for some users who believe that few hours of connection are just "not worth it".

Cranite does answer most of the requirements of the University of New Mexico for wireless security as summarized in Table 4.5.

Requirements	**Cranite**
Authentication	
LDAP Authentication	Supported with slight schema integration
Windows Active Directory	Supported
Secure Tokens Authentication	Supported (EAP-TTLS)
Encryption	
3DES, AES encryption	128-bit AES is mandatory
Non-proprietary client software for encryption	- WirelessWall Client needs to be installed for encrypted access. - No client needed for unencrypted traffic
Management	
Guest Access	Yes for encrypted (with client) and unencrypted access.
Web-based management	Available via the WirelessWall Manager
Bandwidth management	Yes
Traffic prioritization	Not Tested
Hot Failover	Yes via backup servers
DHCP	Yes
NAT	Unsure
“Easy Implementation”	Not Tested
“Easy Management”	Not Tested

Table 4.5: Requirements set forth by the University of New Mexico and solutions offered by Cranite

4.4 Summary

Although many other products exist on the market, Roving Planet, Bluesocket and Cranite seem to be the most popular. Cranite was adopted as the wireless solution for the United States Military Academy, West point, among others. Several government agencies including the National Security Agency deploy Bluesocket in its network and the University of Wyoming and recently Saint Francis Medical Center in Colorado have reverted to Roving Planet for the security of their wireless network. Each of these products has been used in the educational, governmental, and health care sectors, an indication of their capabilities to ensure privacy and integrity of their networks. The choice between these products relies basically on the specific

requirements.

Chapter 5

Benford's Law

In our quest for securing a network from attackers, our focus turned to an unconventional law known as Benford's Law. Our intent was to use this method as a model of network traffic and accordingly to detect abnormalities in the network. In this chapter, we present Benford's Law and offer an explanation of the law that utilizes the maximum entropy distribution. Although the results we obtained are not an exact match, they are close enough to be intriguing. The purpose of this chapter is to check whether the law can be applied to the area of networking, and more specifically the number of hits on a webserver, the number of bytes exchanged and the time between any two consecutive requests.

5.1 History and Statement

In 1881, Simon Newcomb ([49]) noticed that the first pages of the logarithm books were worn more than the later pages. This observation led him to believe that people using those books looked up numbers that start with small digits more often than those beginning with larger digits. He published his results in the American Journal

of Mathematics stating that the frequency of occurrence of digits in natural numbers does not follow a uniform distribution as one would expect, but rather a logarithmic one. However, his article went unnoticed and his observations were not followed.

More than half a century later, a physicist at General Electric made the same observation about the logarithm books. Frank Benford, who apparently wasn't aware of Newcomb's work, arrived at the same conclusions. Benford ([20]), not only stated the different frequencies of occurrence of the leading digits in naturally occurring numbers, but went on to sample data from several sources ranging from rivers' areas, to population census, to numbers appearing in different magazines. He noticed that almost all of the data gathered followed the same logarithmic distribution of the first significant digit. Moreover, when all the data were assembled together to form a single group, the conformity to the logarithmic law became even more evident. Benford stated that "*numbers that individually are without a relationship are, when considered in large groups, in good agreement with a distribution law*", this made him call his observation the law of "anomalous numbers". After the publication of Benford's article in 1938, the observation became known as Benford's law and stirred the scientific community who was intrigued by this "strange" phenomenon and interested in finding mathematical proofs of its existence.

Mathematically, Benford's law states that in numbers that occur naturally, the probability that the first significant digit is equal to a certain d is given by the following,

$$\text{Prob}(D_1 = d) = log_b(1 + \frac{1}{d}) \tag{5.1}$$

where b is the base used and $d = 1, 2, \ldots, 9$. In addition to accounting for the most significant digit, Benford noticed that all digits tend to be dependent. In other words, the probability that the second digit is a specific d depends on the value of the first digit. In fact, in the general case, for all positive integers k, it follows that,

$$\text{Prob}(D_1 = d_1, \ldots, D_k = d_k) = log_b[1 + (\Sigma_k^{i=1} d_i.10^{k-i})^{-1}] \tag{5.2}$$

where d_1 can assume values from 1 to 9, and d_j, for any $j > 1$ can take on any value 0 through 9, since the digit 0 can be in any position except the first. Table 5.1, taken from [50], shows the different probabilities for up to the fourth significant digits.

	Position in Number			
Digit	1st	2nd	3rd	4th
0	-	0.11968	0.10178	0.10018
1	0.30103	0.11389	0.10138	0.10014
2	0.17609	0.10882	0.10097	0.10010
3	0.12494	0.10433	0.10057	0.10006
4	0.09691	0.10031	0.10018	0.10002
5	0.07918	0.09668	0.09979	0.09998
6	0.06695	0.09337	0.09940	0.09994
7	0.05799	0.09035	0.09902	0.09990
8	0.05115	0.08757	0.09864	0.09986
9	0.04576	0.08500	0.09883	0.09982

Table 5.1: Benford's Law Expected Digital frequencies

An interesting question that arises is how one can identify the data sets that usually conform to Benford's Law. In [50], Nigrini declares that in order for a data set to follow Benford's Law closely, it should satisfy three essential criteria. First, the data should be constituted by positive numbers; this is due to the fact that numbers occurring in nature are generally positive numbers. Second, the data should be free of built-in minimum or maximum values. Actually, by limiting the range of values that a certain variable can assume, the leading digits can be greatly affected; for example, tax deductions claims cannot conform to Benford's law since there is a maximum deductions value, meaning that most of the tax returns will be around that value, and not distributed logarithmically. The data set that would follow the law should not be made up of assigned numbers; telephone numbers and zip codes fall within this group. These assigned numbers are in reality a mere representation of what could be words, add to that a certain geographical region always has the same most significant digit which depends on its location. Benford's law has been applied to a variety of areas as will be seen in Section 5.3.

5.2 Proving Benford's Law

In this section, we will review the different methods that were utilized to prove the existence of Benford's Law. No universal theorems or derivations are available and all the proofs that we present in this section are based on practical assumptions. We start with an intuitive explanation as to why such phenomenon may exist, then offer a brief summary of the different mathematical proofs currently available and finally give our method which involves maximum entropy distribution. Our approach did not result in exact conformity with the law, but we were able to obtain very close match.

Looking at Benford's Law from an intuitive approach ([50]), the law seems very logical: assume a city's population is 10,000 at a certain moment of time. The number will have '1' as a most significant digit until the population reaches 20,000. This constitutes a 100% increase. On the other hand, '2' will be the leading digit up until the populating attains 30,000, which is a 50% increase. Intuitively, one would assume that it takes more time for a population to increase by 100%, then it is to increase by 50%, therefore '1' will remain as the most significant digit longer than digit '2'. The same logic can be applied to the rest of the digits 3 through 9.

In a more formal approach, the laws of mathematics are used to prove the existence of Benford's Law. The first explanation of the law was offered by Benford himself in [20]; he starts with the intuitive approach and then specifies that sets of numerical data tend to form a geometric series if the first digits follow a logarithmic behavior. In [30], Cohen uses the natural density of any set of integers S to show that through mapping of different subsets of S it is possible to obtain $d(1) = log_{10}2$, and then goes on to generalize for $d(k)$. Pinkham in [52], proves that Benford's law is the only law that possess the scale invariance property, and then tries to approximate the distribution that results in the proportion of the population with leading digit

d. Flehinger ([33]) assumes that the explanation of the law lies in the properties of the set of integers as represented in a radix number system; in order to get to the probability measure of most significant digits, he uses a limiting process where he establishes that the superior and inferior limits of a density function over the set of integers is the same and is equal to the probability given by Benford's Law. On the other hand, Raimi ([54]), presents his "unorthodox" approach using finitely but not countably-additive measures. A new statistical derivation is presented by Hill in [39] which uses a central-limit like theorem for significant digits. In this paper, Hill also surveyed several papers aiming to prove Benford's law, in addition to establishing that this distribution is unique in satisfying both scale and base invariance.

In an attempt to try to prove the law from a novel approach, we used the maximum entropy distribution concept to examine whether the obtained results would be a good match to those given by Benford. Nevertheless, our method only allowed us to approximate the law without totally agreeing with it.

Entropy, as defined in [31], is the measure of uncertainty of a random variable. In general, the entropy is maximized when the uniform distribution is used. However, when there is a constraint on the numbers, the former statement no longer holds. In fact, for the set of positive numbers, assuming the mean is fixed, the distribution that yields maximum entropy is the exponential distribution, i.e. for $X \in S = [0, \infty)$, $p(X) = \frac{1}{\lambda} e^{-\frac{x}{\lambda}}$ is the distribution maximizing the entropy. This approach was inspired by the example of the distribution of the air particles in the atmosphere subject to the gravitational force and the altitude from the surface of the earth and where the potential energy is a fixed entity; the distribution maximizing the entropy proved to be the exponential distribution. Figure 5.1 illustrates the distribution in question.

In order to calculate the probabilities that the leading digit is equal to d, (assume $d = 1$ for simplicity), we need to integrate the areas under the curve that correspond to ranges of numbers with d as the leading digit. In other words, for $d = 1$, we

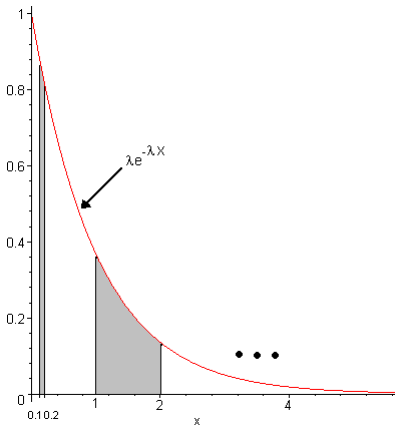

Figure 5.1: Exponential Distribution, and the areas to be integrated for $d = 1$ and base $b = 10$

need to integrate over all ranges $[1.10^i, 2.10^i)$, for all positive and negative integers i. Hence,

$$\begin{aligned}\text{Prob}(D_1 = 1) = \ldots + \int_{0.1}^{0.2} \lambda e^{-\lambda X} dX + \int_{1}^{2} \lambda e^{-\lambda X} dX \\ + \int_{10}^{20} \lambda e^{-\lambda X} dX + \ldots \end{aligned} \tag{5.3}$$

Generally, for any leading digit d and base b, we have the following resulting equation,

$$\text{Prob}(D_1 = d) = \sum_{i=-\infty}^{\infty} e^{-\lambda d b^i}(1 - e^{-\lambda d b^i}) \tag{5.4}$$

The following step is to try to determine what values of λ will result in probabilities that are close to those obtained using Benford's law.

We start by assuming that λ is a fixed number. To give an example, we assume that the leading digit $d = 1$ and the base we are using is $b = 10$, (it is also possible to use any value for d and b). Table 5.2 shows the different results obtained when λ takes on values between 1 and 9.

λ	Probability	Theoretic	Error
1	0.329656978	0.301029996	9.50968%
2	0.287055759	0.301029996	-4.64214%
3	0.271315629	0.301029996	-9.87090%
4	0.281064628	0.301029996	-6.63235%
5	0.297253594	0.301029996	-1.25449%
6	0.311546502	0.301029996	3.49351%
7	0.321638909	0.301029996	6.84613%
8	0.327532868	0.301029996	8.80406%
9	0.32993326	0.301029996	9.60146%
For d = 1; b = 10			

Table 5.2: Results for a fixed value of λ

The interesting point is that any value of λ that is a multiple of 10 of a number leads to the same result. In fact, for $\lambda = 0.001$ or 0.1 or 1 or 10 the resulting probability is the same. Therefore, examining values of λ between 1 and 9, is sufficient to determine the value that gives the closest approximation to the theoretical value of $log_{10}2 = 0.301029996$. From Table 5.2, we notice that when $\lambda = 5.10^i$ for any positive or negative integer value of i, the resulting probability is the closest to the theoretical one but the margin of error is still unacceptable.

Another approach would be to choose λ itself exponentially distributed over the interval $[0, \infty)$ with fixed mean m. Again, to give an example, we consider $d = 1$ and $b = 10$. Then, equation 5.4 will become,

$$\text{Prob}(D_1 = 1) = \sum_{i=-\infty}^{\infty} e^{-10^i\lambda}(1 - e^{-10^i\lambda}) \tag{5.5}$$

The summation of equation 5.5, results in a oscillation around the theoretical value of $log_{10}2$ as seen in Figure 5.2.

Calculating the average of the oscillations using

$$\text{Prob}(D_1 = 1) = \int_0^{\infty} \frac{1}{m} e^{\frac{-\lambda}{m}} \sum_{i=-\infty}^{\infty} e^{-10^i\lambda}(1 - e^{-10^i\lambda}) \tag{5.6}$$

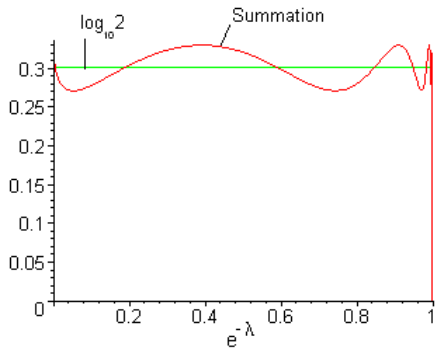

Figure 5.2: Summation of Equation (5.5) and $\log_{10} 2$ for $d = 1$ and base $b = 10$

where $\frac{1}{m}e^{\frac{-\lambda}{m}}$ represents the pdf of λ and m its mean. We then get,

$$\text{Prob}(D_1 = 1) = \frac{1}{m} \sum_{i=-\infty}^{\infty} \frac{10^i}{(\frac{1}{m} + 10^i)(\frac{1}{m} + 2.10^i)} \tag{5.7}$$

Generally, for any leading digit d,

$$\text{Prob}(D_1 = d) = \frac{1}{m} \sum_{i=-\infty}^{\infty} \frac{b^i}{(\frac{1}{m} + db^i)(\frac{1}{m} + (d+1)b^i)} \tag{5.8}$$

Using equation 5.8, we tried varying the values of the base b and the mean m to find out what would yield a better approximation. The results are summarized in Table 5.3.

The table shows that there is no single value of the mean m that minimizes the difference between the obtained values and the theoretical ones. In fact, for base 3, there are two values of the mean that give almost $10^{-5}\%$ error, values 5 and 8, whereas for base 10, the difference is minimized with mean $m = 4$. For base 16, the best value would be 6. A general rule is hard to find, and the numbers don't seem to follow recognizable pattern through which one might be able to deduce some kind of general law that would render the task easier. It is important to keep in mind that when we refer to value d, we mean all values $d.10^i$, for all negative and positive values of i. Still, we did not get to the exact values we were hoping for.

Base	Mean m	Probability	Theoretic	Error
3	1	0.630929622	0.630929754	0.00002%
	2	0.630930065	0.630929754	-0.00005%
	3	0.630929622	0.630929754	0.00002%
	4	0.630929462	0.630929754	0.00005%
	5	0.630929816	0.630929754	-0.00001%
	6	0.630930065	0.630929754	-0.00005%
	7	0.630930036	0.630929754	-0.00004%
	8	0.630929839	0.630929754	-0.00001%
	9	0.630929622	0.630929754	0.00002%
10	1	0.302009891	0.301029996	-0.32551%
	2	0.299432556	0.301029996	0.53066%
	3	0.299865524	0.301029996	0.38683%
	4	0.30105672	0.301029996	-0.00888%
	5	0.302009891	0.301029996	-0.32551%
	6	0.30253957	0.301029996	-0.50147%
	7	0.302704494	0.301029996	-0.55626%
	8	0.302610951	0.301029996	-0.52518%
	9	0.302354975	0.301029996	-0.44015%
16	1	0.253667768	0.25	-1.46711%
	2	0.246332232	0.25	1.46711%
	3	0.244864837	0.25	2.05407%
	4	0.246332232	0.25	1.46711%
	5	0.248562934	0.25	0.57483%
	6	0.250684472	0.25	-0.27379%
	7	0.252404709	0.25	-0.96188%
	8	0.253667768	0.25	-1.46711%
	9	0.254508319	0.25	-1.80333%

Table 5.3: For $d = 1$, The results given by Equation (5.8) for different values of λ and m compared to the results given by Benford's Law

Thus, we moved to a third method that we thought could result in better approximations. The idea is to find the distribution $p(X)$ knowing that its conditional distribution with respect to the mean $\frac{1}{\lambda}$ is exponential. In other words,

$$p(X|\lambda) = \frac{1}{\lambda} e^{\frac{-x}{\lambda}} \tag{5.9}$$

We assume that λ itself is exponentially distributed, with fixed mean m; i.e.,

$$p(\lambda) = \frac{1}{m} e^{\frac{-\lambda}{m}} \tag{5.10}$$

We then calculate the probability distribution of X,

$$p(X) = \int_0^\infty p(X|\lambda).p(\lambda)d\lambda \tag{5.11}$$

$$= \int_0^\infty \frac{1}{m\lambda} e^{(-\frac{x}{\lambda}+\frac{\lambda}{m})} \tag{5.12}$$

Resulting in a distribution $p(x)$ shown in Figure 5.3,

$$p(x) = \frac{2}{m} BesselK(0, 2.\sqrt{\frac{x}{m}}) \tag{5.13}$$

where BesselK is the Bessel function of the second kind.

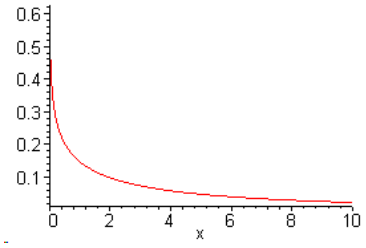

Figure 5.3: Distribution of $p(X)$ as given by equation 5.13

The following step would involve integrating the areas under the curve as was shown in Figure 5.1. Using a decimal base and a most significant digit $d = 1$, we get the probabilities,

$$P(D_1 = 1) = \ldots + \int_{0.1}^{0.2} p(x)dx + \int_{0.1}^{0.2} p(x)dx + \int_{0.1}^{0.2} p(x)dx + \ldots \tag{5.14}$$

Each term in the above expression results in the following expression,

$$\int_a^b p(x)dx = -\frac{2(\sqrt{b}.BesselK(1, 2\sqrt{\frac{b}{m}}) - \sqrt{a}.BesselK(1, 2\sqrt{\frac{a}{m}}))}{\sqrt{m}} \tag{5.15}$$

Using the above equation, we tried to find how close we can get to the theoretic value by varying the mean. Table 5.4 summarizes our results. We notice that we still cannot reach the exact theoretical probability, but are getting closer, when the mean m is chosen to be around 4 and 5, (including all 4.10^i and 5.10^i for all positive and negative values of i).

In order to pursue this approach further, we tried to assume that the probability distribution of λ is actually a conditional distribution with respect to an exponentially

Mean m	Probability	Theoretic	Error
1	0.302426750	0.301029996	-0.46399%
2	0.299712598	0.301029996	0.43763%
3	0.299515714	0.301029996	0.50303%
4	0.300463621	0.301029996	0.18815%
5	0.301466776	0.301029996	-0.14510%
6	0.302185602	0.301029996	-0.38388%
7	0.302580115	0.301029996	-0.51494%
8	0.302704642	0.301029996	-0.55631%
9	0.302631703	0.301029996	-0.53208%
For d = 1; b = 10			

Table 5.4: For $d = 1$, The results given by Equation (5.14) for different values of the mean m compared to the results given by Benford's Law

distributed mean m with $E[m] = \alpha$. But when we integrated using the same tactic, we ended up with a complex expression involving the polygamma function. Hence, we refrained from continuing with the same path and settled with the results that we got so far.

Our maximum entropy approach using the exponential distribution provided us with results that are close to the probabilities given by Benford's Law without really matching them. Actually in [43], the authors examine several distributions and their conformance to the logarithmic law and found that the exponential distribution never really matches the law but is very close. They also test additional distributions and found that the distribution that best approximates the law with an 11-digit agreement is the log-logistic distribution. In addition to that, the authors propose distributions that result in exact accordance to Benford's Law.

5.3 Current Applications

Benford's Law may be a strange phenomenon and although it is sometimes considered counter-intuitive, it can be found in many of the data sets that one can think about. The criteria in Section 5.1, shed some light on what group of numbers would follow the law, however, conformity is even observed in some data that do not exactly abide

by these criteria.

The first applications to Benford's Law were presented by Benford himself in his 1938 article ([20]). The goal of including these diverse data sets was to show that the observation applies to a wide range of data sets from population census to river areas, and form molecular weights to random numbers picked out from magazine pages. In total, Benford's tables included more than 20,000 numbers. Benford noticed that not all sets follow his law perfectly: in fact, the numbers that are of random nature such as numbers extracted from magazines and newspapers tend to better conform to the law than the molecular weights and data that are closely knit. However, when all the data were grouped into one large collection, the numbers tend to follow the law very closely with little error. In Figure 5.4, both the theoretical probabilities and those of the large cluster are shown.

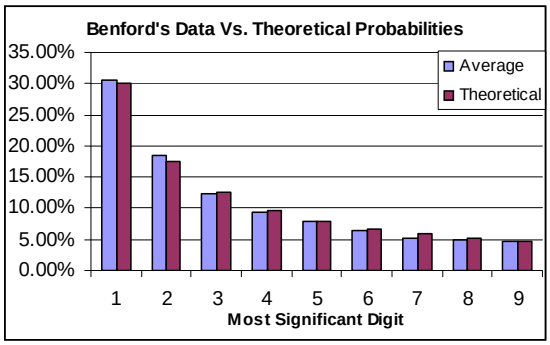

Figure 5.4: Frequency of most significant digits in Benford's original data table vs. theoretical values

To test how available these data sets that conform to Benford's Law really are, we tested some of our own sets to find whether such compliance could be found and to what extent. In our quest, we found that the 5000 Fibonacci numbers follow the law closely, as well as the first 500 factorial numbers as can be seen in Figure 5.5. This was surprising given the fact that data resulting from mathematical formulas do not generally comply with Benford's Law.

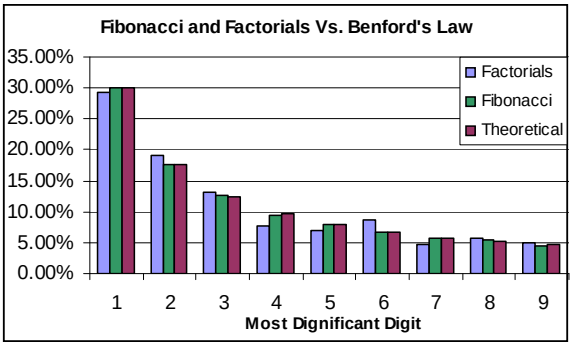

Figure 5.5: Frequency of most significant digits in the first 5000 Fibonacci numbers and the first 500 numbers the factorials. vs. theoretical values

Many researchers have tried to exploit the Benford phenomenon in their areas of interest. What follows discusses some of the interesting work that dealt with the subject in diverse areas ranging from economics and statistics to digital imaging.

In a 1991 note, Burke and Kincanon ([24]) observed that the physical constants tend to have a first digit distribution that approaches Benford's law. The physical constants were extracted from the inside cover of an introductory physics book, and include values such as the gravitational force, the speed of light, etc. When the leading digit was examined, it was found that the digit '1'occurs the most, whether the mks (metric) or English systems were used. The exact frequency however did not follow the exact probabilities given by Benford mainly because the data set in question is very limited: only 20 constants were examined, which could hardly give rise to the exact probabilities since larger data sets are needed for the law to manifest itself.

On a more interesting note, Nigrini ([50]) was able to use Benford's Law successfully in detecting fraud in tax returns. By studying several tax returns and taxation documents, Nigrini was able to prove that the leading digits of these numbers were indeed conforming to the Benford's Law, which implies that any discrepancy from the law would raise a question on the validity of the data presented in the tax forms.

A reason why this could actually work is presented in ([40]); Hill explains that the nature of human beings prevents them from faking data that usually conforms to the law: when people engage in fabricating data, they tend to choose entities that mostly start with the digits 5 and 6, and rarely use numbers beginning with digit 1, which is the mostly used according to Benford's Law. Currently, several software are available to detect how close a-fit a set of numbers included in a tax return form are to the Benford frequency of occurrence tables. In fact, by searching for "Benford's Law software" on Google ([6]), a list of hits containing several products available for download for a fee are returned. These software are not only useful for general use but also for government use; in effect, as was reported in the *Wall Street Journal* on July 10, 1995, the district attorney of Brooklyn discovered fraud in seven companies using software relying on Benford's Law.

Again in the area of economics, Ley ([44]) has found that the series of 1-day returns on the Dow-Jones Industrial Average Index (DJIA) and the Standard & Poor's Index (S&P) (former NASDAQ) reasonably agree with Benford's Law. The one-day return on the index is defined as,

$$r_t = \frac{\ln p_{t+1} - \ln p_t}{d_t} * 100 \tag{5.16}$$

where p_t is the closing value of the stock index at time t, and d_t the number of days between trading days t and $t+1$. Figure 5.6 illustrates the distributions of the DJIA between 1900 and 1993 and the S&P's between 1926 and 1993.

Another area where Benford's Law has proven to have an application is Digital Imaging. In a 2001 article, Jolion, ([41]), shows that the frequency of digits in the magnitude of the two-dimensional gradient of an image obeys Benford's law. Moreover, it was shown that the Benford's Law is well appropriated for the frequency occurrence of any level of the Laplacian transformation of an image. Knowing that the Laplacian pyramid is a compact image code, Jolion suggest using it in entropy

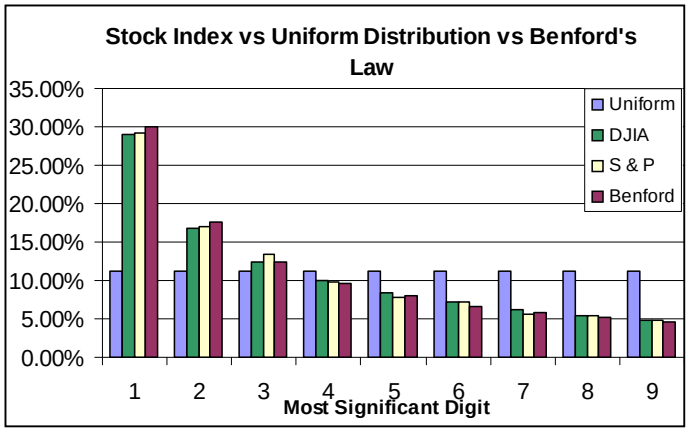

Figure 5.6: Frequency of most significant digits of the Dow Jones and Standard & Poor's Index compared to the previously expected uniform distribution and Benford's Law

based coding which takes into account the distribution of the values in order to optimize their coding. In fact, for the probabilities obtained from Benford's Law, the entropy H_0 is found to be 2.876. Using Huffman coding with the probabilities yields an average length of 2.921, with an efficiency of coding equal to 0.9846. Therefore, the coding scheme is efficient enough to be used in data transmission. An interesting problem that the author presents at the end of the paper is how to use such results in order to efficiently embed data in images.

In [23], Buck et al apply the concept of logarithmic distribution of first significant digits to the half lives of 477 nuclei of radioactive particles. The research focuses on calculations of alpha decay half lives, where atomic nuclei decay by emitting alpha particles spontaneously. By inspecting the half lives, the authors noticed that this data set could be considered as naturally occurring and went on to study the similarity with Benford's Law. Both the measured half life and the calculated half lives are considered and both were found to exhibit a leading digit distribution close to the law but still not an exact fit: digit '1' appeared in the most significant position more often than any other digit, followed by digit '2', the last 3 digits (7, 8 and 9)

did not perfectly conform to the law. However, it is greatly intriguing why Benford distributions would be present in half lives of radioactive material.

Additional interesting applications are included in [36], where Gent and Walsh consider two factors: the space occupied by files on a UNIX computer and the runtime of hard optimization and hard decision problems. First, the file size in number of bytes on a UNIX machine is studied for both one single user and for all users. The results showed great similarity with Benford's Law. Even if the unit of measurement is changed (i.e. if instead of 1 byte, a block of 512 Bytes or 1KB is considered). The paper examines next the runtime of NP-hard problems; specifically the Travelling salesman Problem (TSP) and the Satisfiability of propositional formulae problem (SAT). Runtime in both experiments provided data that appeared to follow Benford's Law closely.

Although Benford's Law has been addressed from many perspectives and applied in different fields, to our knowledge, it has yet to be applied in the area of networking and network traffic modelling. In the subsequent section, we present our approach in modelling traffic and more precisely webserver traffic using Benford's Law.

5.4 Using Benford's Law in the Area of networking

Our interest in Benford's Law extends beyond proving the law from a novel approach and surveying the different areas where it applies; our goal was to find a way where this particular distribution could be applied to networking. More accurately, examine whether it possible to use the law to detect abnormalities in a network such as Denial-of-Service attacks. Our idea was to add to the security measures described in Chapter 3 a fresh element that would offer a new perspective in network security through the

processing of ongoing traffic and detecting variations from the model. The first step was to establish a model using Benford's Law; i.e. try to apply the logarithmic distribution to network traffic characteristics.

The initial parameters that we examined were the number of hits of webserver and the number of bytes transmitted in the response. The goal was to test whether the leading digits in these entities follow Benford's law and how closely. Our testing was conducted using five different data sets retrieved from the Internet Traffic Archives ([7]). The data sets, the period of time when they were samples as well as the total number of requests serviced within that interval of time are shown in Table 5.5. A detailed analysis of the data is available in [16] and [17].

Data Sets	Period	# of Requests
World Cup 98	4/30/98 - 7/26/98	1,352,804,107
NASA	7/1/95 - 7/31/95	1,891,714
University of Calgary	10/24/94 - 10/11/95	726,739
ClarkNet	8/28/95 - 9/10/95	3,328,587
University of Saskatchewan	6/1/95 - 12/31/95	2,408,625

Table 5.5: Data sets examined

The world cup servers were heavily used during the period of data sampling; therefore the corresponding data sets were examined by themselves and not grouped with other data. As for the remaining four data sets, NASA, University of Calgary, ClarkNet and University of Saskatchewan, the retrieved data for each was not sufficient by itself, thus they were merged into one single set upon which we based our work. This data set will be referred to as the NCCS data set. Each of the sets we studied had the form of a log file with timestamps of 1 second resolution, which made our experiments flexible since the intervals of time could be easily modified.

For each of our data sets, the world cup and the NCCS, the data was divided into several groups according to a pre-set interval. The study was first conducted on a 24-hour basis, the repeated for 12, 8, 6, 4, 3, 2, 1 hour intervals an so forth until we reached a point where by further dividing the data more than of the intervals

contained no requests. In the world cup data set, it was possible to reach 36-second intervals while still maintaining a decent number of requests, an indication of how busy the server actually was. At each observation, the most significant digit of both the number of hits and the number of bytes transferred were recorded to finally construct the frequency of occurrence of each digit d, where $d = 1, \ldots, 9$. We start with 24-hour intervals, i.e. per day. Figure 5.7 depicts the world cup data's number of requests and number of bytes recorded in one day for a total of 92 days (i.e. 92 data points).

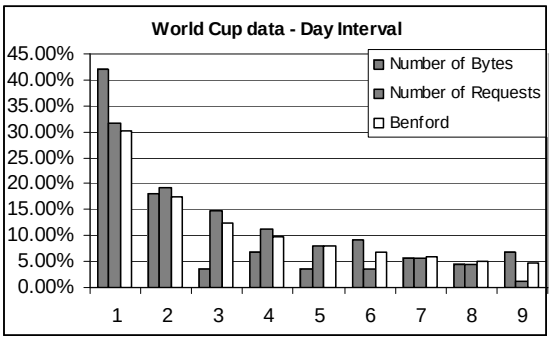

Figure 5.7: Number of hits and Number of Bytes of the World Cup data for a 24-hour period

As can be inferred from the graph, the number of hits per interval of time follows Benford closely and although the number of bytes doesn't seem quite as matching, the probabilities of '1' and '2' being the most significant digits are greater than all others, lead us to think that there is Benford-like behavior without exact match. Figure 5.8 also illustrates a 24-hour interval for the NCCS group. This time, a close match was obtained in the number of bytes transferred but not in the number of hits although the digit '1' was prevailing as the most significant digit.

If the data was divided further into intervals of 1-minute for the World cup data, in other terms 92 days divided into 960 intervals of 1 minute each then grouped together, the conformity was much better as can be seen in Figure 5.9 for both the

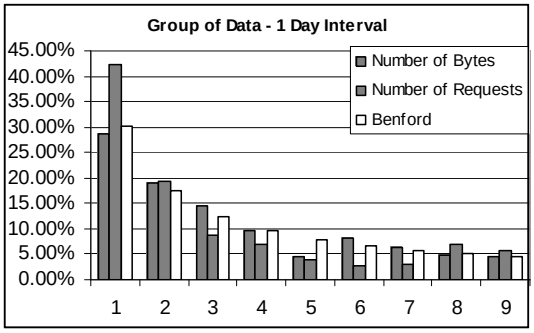

Figure 5.8: Number of hits and Number of Bytes of the NCCS group data for a 24-hour period

number of requests and the number of bytes.

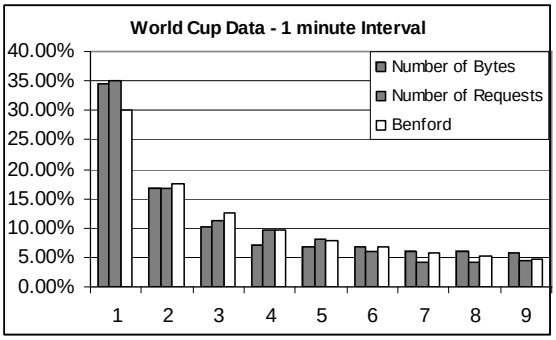

Figure 5.9: Number of hits and Number of Bytes of the World Cup data for 1-minute intervals

For the NCCS group, the smallest achievable interval was the 1-hour interval, where, as can be deduced from Figure 5.10, the match is almost exact for both the number of bytes and the number of requests. In fact, it is only logical that with a large number of data points, the approximation will be as tight as possible.

The above data demonstrated that the number of requests on a webserver and the number of Bytes transferred follow Benford's Law closely when there is enough data to work with. One possible explanation to such observation could be that the number of hits on a server occurs naturally and in scattered instants in time and not following

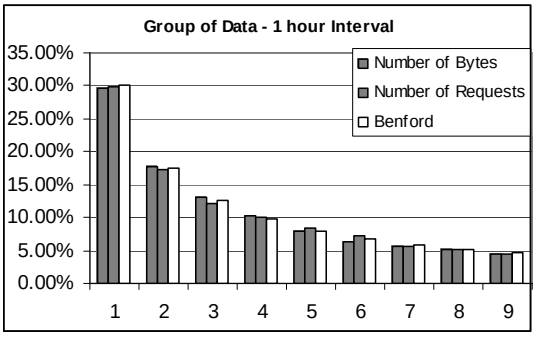

Figure 5.10: Number of hits and Number of Bytes of the NCCS Group data for 1-hour intervals

a certain pattern of occurrence. The same goes for the number of Bytes transferred, where, depending on the requested document, the response is constructed. Thus, we can state that Benford's law constitutes a model for the number of hits on a web server and the number of bytes exchanged.

Nevertheless, when taken over one single interval and not the totality of intervals, the compliance to Benford's Law was lost completely for both the number of hits and the number of bytes. The digit '1' no longer manifested itself as the most frequent leading digit. Even when there is conformity to the law, when the group of all intervals is considered, the results cannot be really used as a security measure since the data required extends to a 100-day period or even more; which is totally useless, since we desire notification of what's wrong as soon as possible and not after the harm is done.

That said, we headed towards our second objective: test whether the inter-arrival time between any two consecutive requests on the web server adhere to Benford's Law. The data we studied was also retrieved from the Internet Traffic Archive [7], is constituted of 18 days' worth of HTTP traces gathered from the Home IP service provided by the University of California at Berkeley and contains around 9 million requests. For analysis of the data, refer to [37]. The data has a Microsecond

resolution; the log files include the time at which the client made the request up to the Microsecond value, which is why we chose this data set to conduct our testing since more than one request can occur in the same second, but it is almost impossible for them to occur in the same Microsecond.

Contrary to the previous experiment where the number of requests and number of bytes were examined, this experiment studied the time elapsed between requests during the entire 18 days, a single day, a single interval of 12, 6, 3 and 1 hour. Moreover, in most days, it was possible to go further to 5 minutes intervals while still having enough data to work with.

The first experiment was for 24-hour intervals. When taken over all 18 days of data, the results, shown in Figure 5.11, are a very close match to Benford's Law.

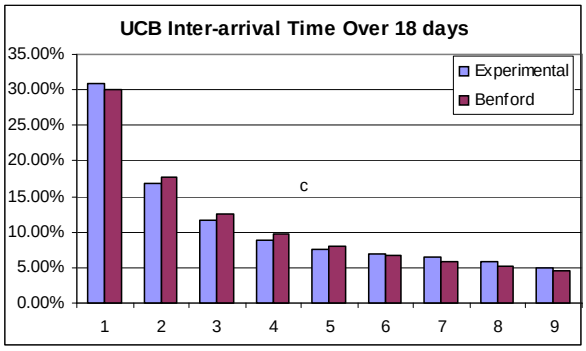

Figure 5.11: Inter-arrival Time Distribution over all 18 days for a 24-hour interval

But as was explained previously, this is not very helpful by itself. That is why we considered a single 24-hour period (i.e. data over one single day), and as can be inferred from Figure 5.12, the match was almost perfect to Benford distributions. Although Figure 5.12 shows the inter-arrival times of requests on Day 3, when we examined the 18 different days, all exhibited the same behavior.

The same behavior was also detected when the data was divided into smaller intervals. In fact, the leading digit of the inter-arrival times over a single 1-hour

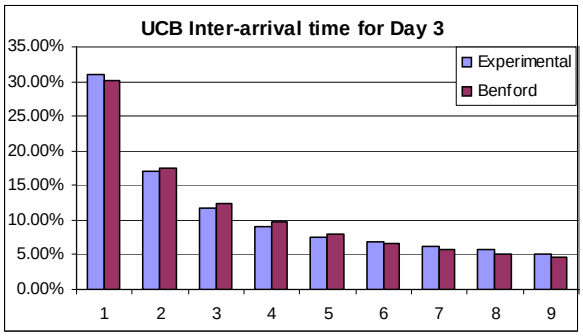

Figure 5.12: Inter-arrival Time Distribution for Day 3

interval (Figure 5.13), regardless of what interval is chosen, were conforming to Benford's law. The compliance with Benford even extends to 5-minute intervals (Figure 5.14).

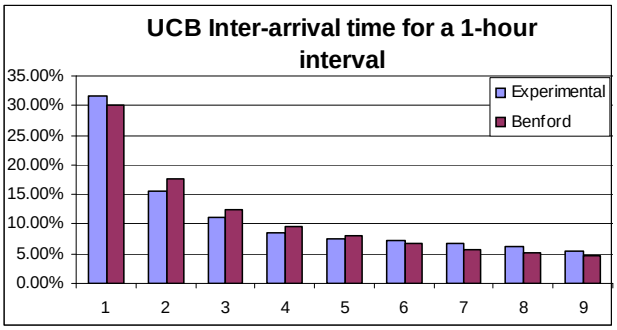

Figure 5.13: Inter-arrival Time Distribution for a single 1-hour interval

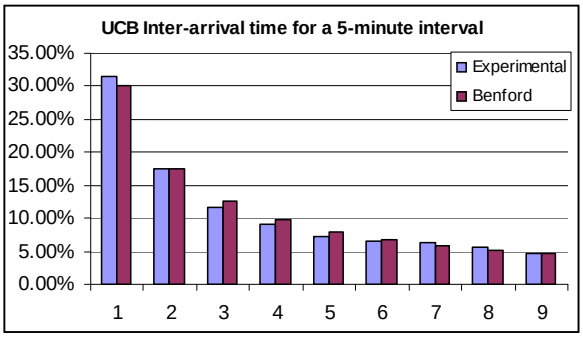

Figure 5.14: Inter-arrival Time Distribution for a single 5-minute interval

The above results confirm our hypothesis that the inter-arrival times obey a logarithmic law. Moreover, at all resolutions, the same behavior is detected.

It has long been suggested that the arrival time of requests or hits on a web server follows a poisson rate. Our discoveries in no way contradict that fact since we are taking into account the leading digit of the inter-arrival time rather than the number as a whole. Therefore, the two models do not interfere with each other and can be viewed as totally independent.

5.5 Summary

Now that we have established that Benford's Law can efficiently be used as a model for network traffic and more precisely the inter-arrival time of requests at a webserver, there is a possibility to take advantage of this property in detecting variations in the traffic, that could be a sign of something wrong in the network. In fact, one can set a 15-minute interval during which the traffic is continuously sampled and the distributions of the leading digits are calculated and the compliance with the Benford's Law is tested. If the distributions match within a certain error margin, then the network is labelled as "safe" for that time. If on the other hand there is a considerable discrepancy between the values, then a warning message is sent to the administrator or person responsible of the network, inviting them to check the status of the network because it may be under attack. The warning is not a sure indication of abnormalities, but a simple alarm that could be false.

The idea described above is simply one of the different possibilities where the relation between network traffic characteristics and Benford's law can be exploited. Other usages, with diverse goals, could also be employed. However, since the main flavor of this document is security, wireless security specifically, the risk of abnormality in the network was examined.

Chapter 6

Conclusions

In this thesis, we reviewed the different methods that provide security to one's wireless network. Whether the need is to have an open network, where anyone can get connected, or a tight wireless environment where only very limited access is granted, the choice is left to the administrator. Open authentication, SSID broadcast, MAC Filtering are all rudimentary ways to provide "some kind" of privacy, which is hardly sufficient. WEP is vulnerable to several attacks but an unskilled attacker will have some difficulties cracking the security since the operation does require some knowledge and technical skills not readily available. WPA is "so-far" a solid solution which prepares for the 802.11i standard that should bring a universal approach to enuring the security of wireless networks. In addition to these measures and standards, several off-the-shelf products can be installed to control access to the network as well as manage it, a feature that none of the above mentioned measures presents. This makes such products appealing to administrators who wish to have an all-in-one solution to their wireless tasks.

An interesting topic that we covered in this thesis is Benford's Law, its justifications, applications, and our way of utilizing it in modeling a webserver's traffic.

Benford's Law was used as a model for the number of hits, inter-arrival times between requests, and the number of bytes exchanged during a connection with the webserver. This model may be used to detect abnormalities, such that whether the server is under attack.

Interesting future work will comprise in simulating an attack on a webserver and checking how feasible it is to detect that something is wrong for a certain interval of time. We believe that this topic deserves much more study, since it could be useful in keeping a network safe from attacks such as Denial-of-Service where detecting a great amount of hits in a short time could be an indication of such attack. An interesting application, that deserves our attention, is using Benford's Law as a model for requests on an access point. This way, one can detect if anything suspicious is happening on the wireless network. The only drawback in applying such logic is the fact that large amounts of data are required in order to successfully establish the model, which would be impractical for small networks with a limited number of users.

References

[1] Airsnort. http://airsnort.shmoo.com/.

[2] Bluesocket website. http://www.bluesocket.com.

[3] Canberra aquila inc. http://www.aquilagroup.com.

[4] Cisco. http://www.cisco.com/.

[5] Cranite website. http://www.cranite.com.

[6] Google, search engine. http://www.google.com.

[7] The internet traffic archives. http://ita.ee.lbl.gov/.

[8] Kismet. http://www.kismetwireless.net/.

[9] Netstumbler. http://www.netstumbler.com/.

[10] Roving planet website. http://www.rovingplanet.com.

[11] Wepcrack. http://wepcrack.sourceforge.net/.

[12] J. Allen and J. Wilson. Securing a wireless network. *Proc. Of the 30th Annual ACM SIGUCCS Conference on User Services*, 1:213–215, 2002.

[13] Wi-Fi Alliance. Wi-Fi Protected Access: Strong, standards-based, interoperable security for today's Wi-Fi networks. White Paper, 2003.

[14] W. A. Arbaugh, N. Shankar, and Y.C.J. Wan. Your 802.11 wireless network has no clothes. http://www.cs.umd.edu/ waa/wireless.pdf, 2001.

[15] W.A. Arbaugh. Wireless security is different. *IEEE Computer Society*, 36(8):99–101, August 2003.

[16] M. Arlitt and T. Jin. 1998 world cup web site access logs. Available at http://www.acm.org/sigcomm/ITA, August 1998.

[17] M. Arlitt and C. Williamson. Web server workload characterization: The search for invariants. In *Proceedings of the 1996 ACM SIGMETRICS Conference on the Measurement and Modeling of Computer Systems*, Philadelphia, PA, 1996.

[18] N. Asokan, V. Niemi, and K. Nyberg. Man-in-the-middle in tunnelled authentication protocols. Technical Report Technical Report, Nokia Research Center, Finland, 2002.

[19] L. Barken. *How Secure is you Wireless Network: Safeguarding Your Wi-Fi LAN.* Prentice Hall PTR, 2003.

[20] F. Benford. The law of anomalous numbers. *Proceedings of the American Philosophical Society*, 78:551–572, 1938.

[21] D. Benson and M. Gold. Survey of selected 802.xx wireless standards. SRI Consulting Business Intelligence, 2003.

[22] N. Borisov, I. Goldberg, and D. Wagner. Intercepting mobile communications: the insecurity of 802.11. *Proceedings of the 7th annual international conference on Mobile computing and networking*, 1:180 – 189, 2001.

[23] B. Buck, A.C. Merchant, and S.M. Perez. An illustration of benford's first digit law using alpha decay half lives. *European Journal of Physics*, 14:59–63, 1993.

[24] J. Burke and E. Kincanon. Benford's law and physical constants: the distribution of initial digits. *American Journal of Physics*, 59(10):952, 1991.

[25] N. Cam-Winget, D. McGrew, J. Salowey, and H. Zhou. EAP flexible authentication via secure tunneling (EAP-FAST). Internet Draft: draft-cam-winget-eap-fast-00.txt, 2004.

[26] N. Cam-Winget, T. Moore, D. Stanley, and J. Walker. IEEE 802.11i overview. NIST 802.11 Wireless LAN Security Workshop, Fall Church, Virginia, December, 2002.

[27] B. Carter and R. Shumway. *Wireless Security: End to End.* Wiley Publishing, Inc, Indianapolis, Indiana, 2002.

[28] 2004 Cisco Release. Dictionary attack on Cisco LEAP vulnerability. http://www.cisco.com/en/US/tech/tk722/tk809/technologies_security_ notices09186a00801aa80f.html.

[29] 2004 Cisco Release. EAP-FAST. http://www.cisco.com/en/US/products/hw/wireless/ps430/products_qanda_item09186a00802030dc.shtml, 2004.

[30] D.I.A. Cohen. An explanation of the first digit phenomenon. *Journal of Combinatorial theory (A)*, 20:367–370, 1976.

[31] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

[32] D. Eaton. Diving into the 802.11i spec: A tutorial. CommsDesign, http://www.commsdesign.com/design_library/cd/hn/OEG20021126S0003, November, 2002.

[33] B.J. Flehinger. On the probability that a random integer has initial digit A. *The American Mathematical Monthly*, 73(10):1056–1061, 1966.

[34] S. Fluhrer, I. Mantin, and A. Shamir. Weakness in the key scheduling algorithm of RC4. *Eighth Annual Workshop on Selected Areas in Cryptography*, 2001.

[35] M.S. Gast. *802.11 Wireless Networks: The Definitive Guide*. O'Reilly, Sebastopol, California, 2002.

[36] I. Gent and T. Walsh. Benford's law. Technical Report apes-25-2001, Algorithms, Problems and Empirical Studies Group, 2001.

[37] S.D. Gribble. UC Berkeley home IP HTTP traces. Available at http://www.acm.org/sigcomm/ITA, July 1997.

[38] D. Halasz. IEEE 802.11i draft & call for interest on link security for IEEE 802 networks. IEEE 802 Security CFI Plenary, November, 2002.

[39] T.P. Hill. A statistical derivation of the significant-digit law. *Statistical Science*, 10:354–363, 1996.

[40] T.P. Hill. The difficulty of faking data. *Chance*, 26:8–13, 1999.

[41] J.M. Jolion. Image and the benford's law. *Journal of Mathematical Imaging and Vision*, 14:73–81, 2001.

[42] R. Jordan and C.T. Abdallah. Wireless communications and networking: An overview. *EEE Antennas and Propagation Magazine*, 44(1):185–193, February 2002.

[43] L.M. Leemis, B.W. Schmeiser, and D.L. Evans. Survival distributions satisfying benford's law. *The American Statistician*, 54(3), 2000.

[44] E. Ley. On the peculiar distribution of the U.S. stock indexes' digits. *The American Statistician*, 50(4):311–313, 1996.

[45] L. Loeb. Roaming charges: Are you ready for 802.11i? IBM Article, http://www-106.ibm.com/developerworks/wireless/library/wi-roam19.html, February, 2004.

[46] S.S. Miller. *WiFi Security*. McGraw-Hill Networking, 2003.

[47] A. Mishra and W.A. Arbaugh. An initial security analysis of the IEEE 802.1x standard. Technical Report CS-TR-4328, University of Maryland, 2002.

[48] R. Moskowitz. Weakness in passphrase Choice in WPA interface. Wi-Fi Networking News: http://wifinetnews.com/archives/002452.html, 2003.

[49] S. Newcomb. Note on the frequency of use of the different digits in natural numbers. *American Journal of Mathematics*, 4(1):39–40, 1881.

[50] M.J. Nigrini. *Digital Analysis Using Benford's Law*. Statistical Science, Global Audit Publications, 2000.

[51] L. Phifer. Taming wireless security blues with Bluesocket. Available at http://isp-planet.com/fixed_wireless/technology/2002/bluesocket.html, October 2002.

[52] R.S. Pinkham. On the distribution of fisrt significant digits. *The Annuals of Mathematical Statistics*, 32(4):1223–1230, 1961.

[53] B. Potter. Wireless security's future. *IEEE Security and Privacy*, 1(4):68–72, July- August 2003.

[54] R.A. Raimi. On the distribution of first significant figures. *The American Mathematical Monthly*, 76(4):342–348, 1969.

[55] F. Robinson. Examining 802.11i and WPA; the new standards – up close. Network Computing, http://www.nwc.com/showitem.jhtml?docid=1506ws1, April, 2004.

[56] P. Roshan. A comprehensive review of 802.11 wireless LAN security and the Cisco wireless security suite. Cisco white paper, 2002.

[57] W. Stallings. *Network and Internetwork Security*. IEEE Press, New Jersey, 1995.

[58] A. Stubblefield, J. Loannidis, and A.D. Rubin. Using the Fluhrer, Mantin, and Shamir attack to break WEP. Technical Report Technical Report, AT&T Labs, 2001.

[59] M. Sutton. Hacking the invisible network: Insecurities in 802.11x. iDEFENCE White Paper, 2002.

[60] T.M. Swaminatha and C.R. Elden. *Wireless Securityand Privacy: Best Practices and Design Techniques*. Addison-Wesley, Boston, Massachusetts, 2003.

[61] R.D. Vines. *Wireless Security Essentials: Defending Mobile Systems from Data Piracy*. John Wiley & Sons, 2002.

[62] J.R. Walker. Unsafe at any size; an analysis of the WEP encapsulation. Technical Report IEEE document 802.11-00/362, Intel, 2000.

[63] D.J. Welch and S.D. Lathrop. A survey of 802.11a wireless security threats and security mechanisms. Technical Report ITOC-TR-2003-101, United States Military Academy, West Point, 2003.

[64] J. Wright. Detecting wireless LAN MAC address spoofing. White Paper, 2003.

[65] J. Wright. Weaknesses in LEAP challenge/response. Presentation, http://home.jwu.edu/jwright/presentations/asleap-defcon.pdf, 2003.

[66] Y. Zahur and T.A. Yang. Wireless LAN security and laboratory designs. *The Journal of Computing in Small Colleges*, 19:44 – 60, January 2004.