Lightweight MAC-Spoof Detection Exploiting Received Signal Power and Median Filtering

Papini, Davide

Published in:
Proceedings of the 5th Nordic Workshop on Dependability and Security (NODES'11)

Publication date:
2011

Lightweight MAC-Spoof Detection Exploiting Received Signal Power and Median Filtering

Davide Papini

DTU Informatics, Danmarks Tekniske Universitet
Richard Petersens Plads, Kgs. Lyngby, DK 2800
dpap@imm.dtu.dk

Abstract. IEEE 802.11 networks are subject to MAC-spoof attacks. An attacker can easily steal the identity of a legitimate station, even Access Points, thus enabling him to take full control over network basic mechanisms or even access restricted resources. In this paper we propose a method to detect this kind of attack based on signal power monitoring. The main contribution of our work is the introduction of a *median filter* that enables the detection of the attack by looking at the variance of the signal power. We take into account two types of references for the samples, time and number of frames, and compare their detection capabilities. Our experimental results show that the spoofing attack is successfully detected with both type of references. Moreover the *median filter* helps to reject false positives.

Keywords: Intrusion Detection Systems, Security, Wireless LANs, IEEE 802.11, signal power, variance, median filter, spoof.

1 Introduction

Although the IEEE 802.11 standard has been modified over the years and extended to include stronger cryptographic mechanisms and security policies, many threats are still present, some of them very severe. These threats are very difficult to mitigate since they are a consequence of protocol basics which for the moment can not be changed due to legacy devices support. Moreover dealing with radio waves is not the same as dealing with wires. Radio waves spread out to uncontrolled areas and are difficult to contain; thus environment control is not feasible. Hence the demand for a technology to monitor malicious activities and detect attacks arises.

There are several threats and attacks that can be carried out against wireless networks. These can be summarized in:

- Message Injection and Active Eavesdropping;
- Message Deletion and Interception;
- Masquerading: Malicious AP and Session Hijacking;
- Man-in-The-Middle;
- Denial of Service (DoS) attacks.

Some of them, such as Malicious AP and session hijacking, are based on MAC-spoof which consists in assuming someone else's identity (usually a legitimate station). This is done by first identifying a suitable target and then changing attacker's own MAC address to the target one after having optionally disabled the target. This is trivial to do [13] and at the same time not easy to detect. Someone could argue that with the introduction of Wi-Fi Protected Access (WPA) and WPA2 , the effectiveness of this attack has been reduced significantly. Unfortunately this is not true since these solutions protect only data frames [2]. Control and management frames still lack any kind of security protection. These types of frame are responsible for important and sensitive functions, critical for correct network operations (e.g. authentication/deauthentication and collision detection and avoidance) so detecting spoofed frames is still a significant problem [10]. 802.11w addresses this vulnerability but it is still in the standardization process, and even when it is issued, requirements for compatibility with legacy devices mean that it will not be fully or widely enforced. In this paper we focus on a method to detect MAC-spoof attacks.

In the past years research on methods to detect MAC-spoof in wireless networks have given a number of different results. Meanwhile WiFi technology has developed to include innovations, such as per-device multiple antennas, that make some of these methods obsolete and useless in practice [11]. The problem has been approached from two different angles, one based on mechanisms or techniques that operate at a logic level [12,9] and one relying on features related to the physical medium, like signal power or Round Trip Time (RTT) [7,8,6,14,4,3,15], which are unspoofable (or nearly impossible to spoof). Signal power analysis is therefore a natural choice for a method to detect MAC-spoof.

In Section 2 we talk about wireless signal characterization and related work in detecting MAC-spoof; then in Section 3 a detection method is proposed. Sections 4 and 5 describe the experimental setup and the results. Finally Section 6 discusses conclusions and future work.

2 Wireless Signal Characterization and Related Work

Signals in WiFi networks are electromagnetic waves in the ISM band (2.4-2.5 GHz and 5.7-5.8 GHz). Like every radio signal they are subject to attenuation, the power is inversely proportional to the distance. If that were the only attenuation present in the wireless environment, the signal of a static station would be very stable. However wireless networks rely on the so called *multipath propagation*, which allows the signal to spread between adjacent rooms by reflections and refractions. Finally there is also the problem of interference, either between WiFi devices or due to other sources, that needs to be considered. The result is that the signal power (see Figure 1) shows an erratic behaviour with random spikes. A complete model is not within the scope of this paper, this is just to give an idea of the problem. If the reader wishes to read more about the topic a good starting point could be [1,16,5].

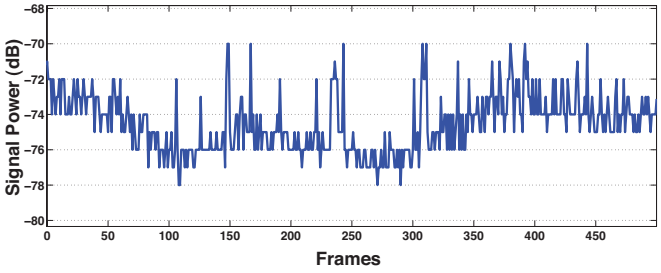

Fig. 1. Signal power of a static station.

Signal power is also dependent on other factors directly related to the device, namely the type of wireless card, antenna, driver etc. So in order to steal an identity undetected (by a perfect detector) an attacker should find a way to forge signals with respect to all these parameters. This task is infeasible in practice.

Signal power is directly related to the relative position of the station and the Access Point (AP) which is considered fixed. If the station moves then the signal will change. The same thing indeed happens when MAC-spoof occurs: the signal will suddenly change since station and attacker are usually in different places (or even if they were not, they would be at least meters apart). This change can be detected, the real problem is to be able to distinguish between false positives (due to signal spikes) and the actual attack. There is also another thing to take into account, wireless devices usually move in the environment (PDAs, Smartphones and on a larger timespan even laptops). This should be dealt with when designing a technique to detect MAC-spoof. In [7,8] a method that monitors subsequent frames power differences is proposed. Here when the difference between two subsequent frames is over a certain threshold an alarm is raised. We have attempted to implement this method but it has proven useless for several reasons: random spikes in legitimate wireless signal patterns, the lack of a time reference that makes it possible to discern how fast the change in the signal pattern occurs. Another proposed method consists in computing the Round Trip Time (RTT) between AP and station. This method is even more unreliable since the time resolution needed for that is in the order of nanoseconds (i.e. the resolution needed to tell users meters apart[1]) which is not that easy to achieve. If you add the fact that frames are variable length, measuring RTT becomes even more difficult. [7,8] propose for this purpose to use fixed length control frames and exploits the RTS-CTS procedure, which is a collision avoidance technique specified in the 802.11 standard. In fact this procedure is almost never used, making every detection scheme based on it of limited use. [6] uses signal power to generate *signalprints* in order to be able to identify different stations. In [11] a different approach is taken. Here the focus is on signal distribution analysis due to antenna diversity. The phenomenon is of most interest for multiple antenna

[1] If we consider a speed of $3 * 10^8 \frac{m}{s}$, a radiowave crosses $1m$ in $3ns$.

devices (typically routers) which generate a multimodal distribution. Data are gathered through a grid of multiple sensors and then processed to fingerprint stations and identify them. A similar approach is taken in [4,3,15,14], where the probability density fuction of the signal power is used to identify and locate single stations in a multiple sensors environment. The last two approaches are indeed innovative but heavier from a computational point of view.

The method we decided to implement relies on signal power and timing data which are gathered from multiple sensors. It uses simple statistical functions (e.g. mean and variance), along with some lightweight filtering, in order to keep computation light. As a matter of fact computation is very relevant since it is not only important to detect the attack, but also to do it in a reasonable time so that countermeasures can be put in place.

3 Detecting MAC-Spoof

This section is divided into three parts. In the first we describe how signal data are gathered, the second how data are processed and why. Finally in the last part we describe how to detect an attack.

3.1 Gathering Signal Data

All data about signal power are taken directly from the wireless card. There are two types of header from which they can be extracted: the Prism header and the Radiotap Header[2]. The first one reports the signal in terms of Received Signal Strength Indication (RSSI) which is an 8 bit value whose interpretation is card vendor dependent. On the other hand Radiotap Header provides more flexibility and more detailed data namely a 64 bit *Time Synchronization Function Timer* (TSFT)[3] and an 8 bit *Antenna Signal*[4]. Radiotap has therefore been chosen as the best solution both for time resolution and signal power representation.

3.2 Processing Signal Data

As already explained in Section 2 MAC-spoof occurs when an attacker assumes the identity of a legitimate station. In order to detect this by signal monitoring we first need to perform some filtering and statistics on the received signal.

The method we propose first applies a *median filter* to the data. This is done in order to filter out spikes due to the medium. A *median filter* sorts all the values in a window of length n (usually an odd value) from low to high, and then takes the value in the center (see Figure 2). It is used in signal processing to remove salt and pepper noise.

[2] For detailed description: *www.radiotap.org*.
[3] Value in microseconds.
[4] Value in dB from an arbitrary fixed reference.

The lower the window length is, the better the performance and response time. We tried values of 3, 5, 7, 9, 11 and 13. In our experiments a value of 3 was enough to filter out spurious data. It is important to stress that *median filer* has a key role in lightwight spike filtering; moving average or other smoothing techniques do not achieve similar results since spike samples are not excluded from the computation.

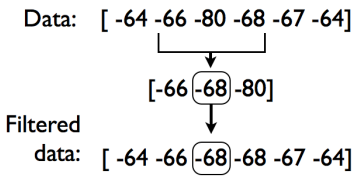

Fig. 2. Median filtering.

After the *median filter*, we compute the mean and variance of samples over a window of a certain size (from now on the term "window" denotes the number of samples of signal power over which the mean and variance are computed). The size is a crucial variable because if we set a value that is too low we risk overlooking MAC-spoof attacks; if it is too high we could mistake attacks for legitimate variation in the signal (i.e. station changes position). The window size can be based on two different measures: number of frames and time. Both present advantages and disadvantages:

- *Frame-based*: variance values are more steady since the computation is always performed on the same number of samples. It catches attacks irrespective of their duration. The problem is that it cannot distinguish between mobile and static stations since it cannot tell how much time it took for the signal power to change, thus raising false positives in the mobile case. Furthermore it cannot allow for the differences between data burst (e.g. FTP sessions) and non-intensive (e.g. WEB surfing)traffic.
- *Time-based*: can distinguish between mobile station and MAC-spoof attacks. If an attacker waits enough time for the window to pass between the disabling of the station and the very first data transmitted, the attack would not be detected. TSFT is taken as time reference[5].

We performed experiments with both. Results are shown in Section 5.

3.3 Detecting the Attack

The variable we use to detect the attack is the variance of the signal power. The longer the signal power differs from its mean the more the variance grows through time. Mean and variance are computed within the defined window size, the bigger the size the slower the variance grows. Moreover if the window is too big an attack could be neglected when the window passes from the legitimate station samples to the attacker ones (i.e. all the samples in the window are from the attacker only). If the variance changes significantly then an attack is taking place.

[5] This is indeed more precise then CPU time since it is taken directly from the wireless card at the first frame's received bit.

4 Experimental Setup

The setup is composed of 4 sensors (2 laptops and 2 workstations), a station (a workstation) and an attacker (a laptop). The hardware configuration is shown in Table 1.

Device ID	CPU/RAM/OS	card/chipset
Sensor (W) I	core i7/12GB Fedora 11	Orinoco 11abg PCI Atheros AR5001X+
Sensor (W) II	Celeron@400MHz 512MB/Fedora 13	Orinoco 11abg PCI Atheros AR5001X+
Sensor (L) III	MacBook Pro 4GB/Snow Leopard	Airport Extreme BCM43xx
Sensor (L) IV	Latitude D410 1GB/Ubuntu 9.10	Orinoco PCMCIA Atheros AR5001X+
Station (W) V	i7/768MB Backtrack 4	Linksys WUSB54GC Ralink rt73
Attacker (L) VI	Latitude D410 1GB/Ubuntu 9.10	Intel BG2200

(W)=Workstation (L)=Laptop

Table 1. Hardware configuration.

The setup comprises two rooms on the same floor (see Figure 3). The rooms are roughly 10 meters apart, separated by another room. Room #1 measures 3.10×7 m while room #2 15×7 m. We deployed 2 sensors in each room, the attacker in room #2 and the station in room #1. This is a public dynamic environment, so other stations and users were present and people were coming and going (i.e. doors opened and closed). This makes the results more interesting and significant for a real deployment scenario.

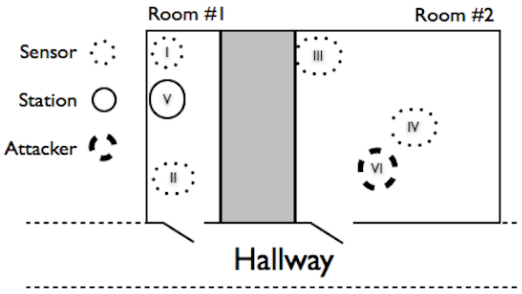

Fig. 3. Experimental setup.

All measures regarding unknown stations were not taken into consideration for the experiments, but they passively contribute to the interference and channel collision phenomena.

Three kinds of measurements were performed:

1. Normal data session "static": a static station transmits and receives data.
2. Normal data session "mobile": a mobile station transmits and receives data.
3. MAC-spoof session: an attacker performs a MAC-spoof attack.

The first two were performed in order to investigate the behaviour of normal traffic, the last for the attack traffic. We performed about 200 experiments both for normal and MAC-spoof sessions and all experiments showed similar results wich are outlined in the next section.

5 Results

This section is divided into four parts. The first shows the effect of a *median filter*. In the second and third we consider the variance of the signal power in normal data sessions and in MAC-spoof sessions. We then discuss further considerations on median filter and finally talk about the performance of the sensors.

5.1 The Effect of Median Filtering

In Section 3.2 we have already explained how a *median filter* works.

Figure 4 shows its effects, presenting first raw data and then data filtered with a *median filter* of size 3 and 5; all data where taken from the same FTP session. The graph shows signal power differences in dB between subsequent frames. Here spikes are points with an high absolute value (e.g. > 8 dB).

Noticeably, with a filter of size 3, rejection of spikes is quite effective: raw data show spikes up to 25 dB while after the first filtering we see just spikes of 4 dB at most. In experiments we found out that in very noisy situations some spikes still survive to a size 3 median filter, while with a size 5 they are filtered out. Nevertheless spikes that survived were isolated and did not influenced the variance of the signal power at all. Therefore we decided to apply a *median filter* of size 3 to the samples.

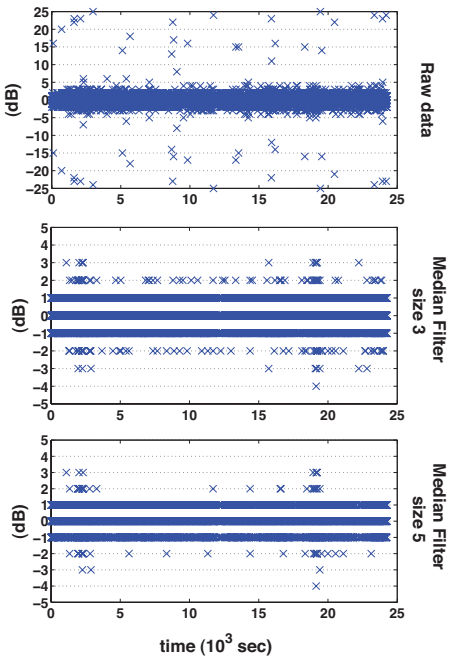

Fig. 4. Median filter effect on signal power differences between subsequent frames.

5.2 Normal Data Session

Before performing experiments on MAC-spoof detection we investigated the behaviour of signal power for legitimate stations. We performed two sets of measurement:

1. "mobile": a station moves from within rooms up to 25 m far while sending data.
2. "static": a station is in a fixed position while sending data.

For each set we performed two measurements, *frame-based* and *time-based*. We used values ranging from 200 $frames$ to 10000 $frames$ for the first from 0.5 s to 16 s for the second. We choose to show results with a window length of 200 and 4000 in the first case, and a length of 2 s and 8 s for the second one since these significantly represent results.

Figure 5 shows results for the mobile case and Figure 6 for the static one.

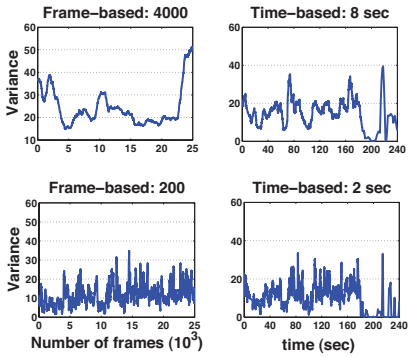

Fig. 5. Mobile station: *frame-based* and *time-based* signal power variance.

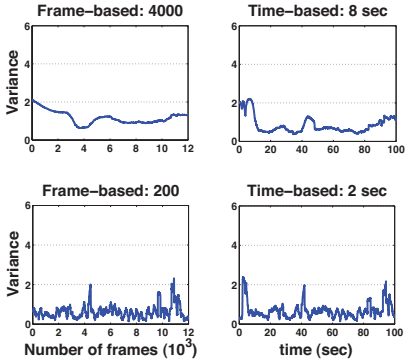

Fig. 6. Static station: *frame-based* and *time-based* signal power variance.

In the mobile case we can see that with a longer window the maximum value of the variance is higher, both for *time-based* and *frame-based*. This is expected since a longer window captures a larger movement of the station, resulting in signal power larger variations. It is interesting to notice that *frame-based* measures never present a value close to zero, while for *time-based* (Figure 5) this happens noticeably (e.g. at 200 s). We call this *return to zero phenomenon*. This is due to the fact that the station does not transmit continuously but it has period of inactivity. This is registered by the *time-based* method while it cannot happen with the *frame-based* one. Moreover, due to this fact, the variance shows higher values in the *frame-based* (over 50).

In the static case variance values are lower (below 6). This is due to the fact that the station does not move. We observe a similar behaviour with respect

to the *return to zero phenomenon* (see Figure 6). Furthermore *time-based* measurements return to zero frequently (still due to the fact that stations are often inactive), while *frame-based* ones hold on around the same value (this shows that the station does not really move between periods of inactivity). Particularly in this case the stability of variance depends on the length of the window.

Summing up, mobile stations shows higher and more erratic values than in the static case. This behaviour can be used to distinguish mobile stations from static ones.

5.3 MAC-Spoof Session

Figure 7 shows variance data related to a MAC-spoof attack. There are 4 graphs, 2 for time based measures and 2 for frame based ones. Results shows that when MAC-spoof occurs the sensor registers a big spike in the variance up to values of 80 and 300 for *time-based* and over 400 for *frame-based*. Both methods therefore successfully detect the attack. Moreover this behaviour differs noticeably from normal variance values to a point that it is very difficult to mistake an attack for legitimate traffic, even when this is mobile. It is also interesting to notice that the variance peak increases as the window becomes shorter (e.g. from 80 to 280 for *time-based* and from 400 to 500 for *frame-based*). This behavior is the opposite with respect to normal data sessions. This different behaviour is an additional indication of a MAC-spoof attack.

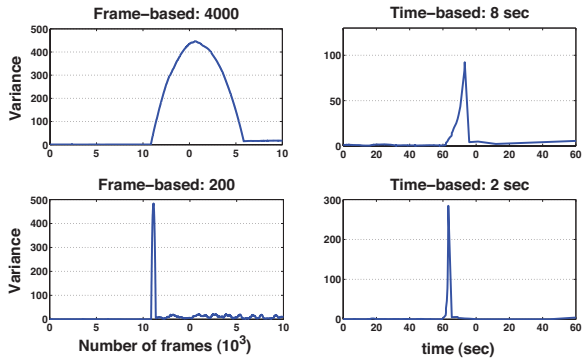

Fig. 7. MAC-spoof signal power variance.

We have also performed the attack in a setup where the attacker and the station were positioned closely (to see if positioning devices nearby could somehow avoid detection from the sensor). Results showed that even though devices were near it was still possible to detect the attack. This is because signal power depends also on other factors than merely position (see Section 2).

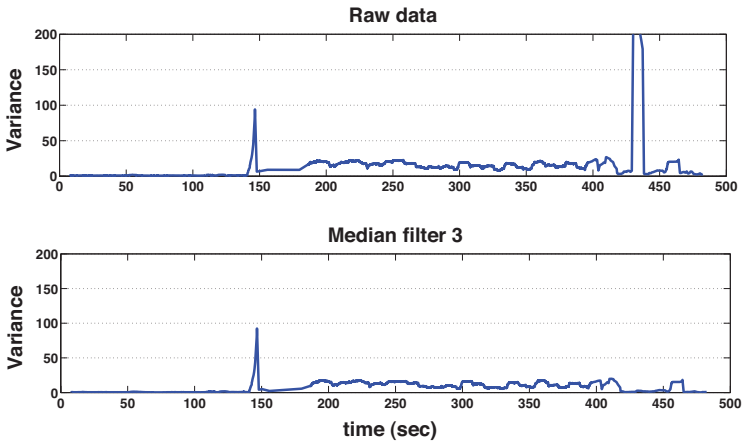

Fig. 8. Median filter false positive rejection.

5.4 Further Considerations on Median Filter

The effect of *median filtering* has already been discussed in Section 5.1. Now, after having discussed MAC-spoof detection, we can also show another important effect: *median filtering* helps to reject false positives. Figure 8 shows the variance during a MAC-spoof attack, computed first on raw data and then on filtered data. In the first case the variance presents two spikes, while in the filtered data only one. The disappeared spike is a false positive which would have raised a false alarm if it had not been filtered out. Furthermore the second spike is even bigger than the first one. *Median filtering* has therefore two functions: for normal traffic it limits the variance of the signal power while for MAC-spoof attacks it helps to reject false positives.

5.5 Performance Evaluation

We implemented and tested the system with different sensor configurations (see Table 1) in order to check both consistency between measurements taken from different platform/hardware and performance across different configurations. Results show that measurements are consistent across different platforms and hardware. This means that potentially it can be implemented on any system without invalidating the method. With regard to the performance, the only thing to stress is that Sensor II was unable to catch the same traffic, in terms of number of frames, as the other ones. This is mainly due to the processor that simply does not have the computational power to process high frame rate of data intensive sessions.

There is also another aspect to take into the performance evaluation: that is the difference between *frame-based* and *time-based* methods. We found out that the *time-based* is almost 30-40 times faster than the first one. As you can see

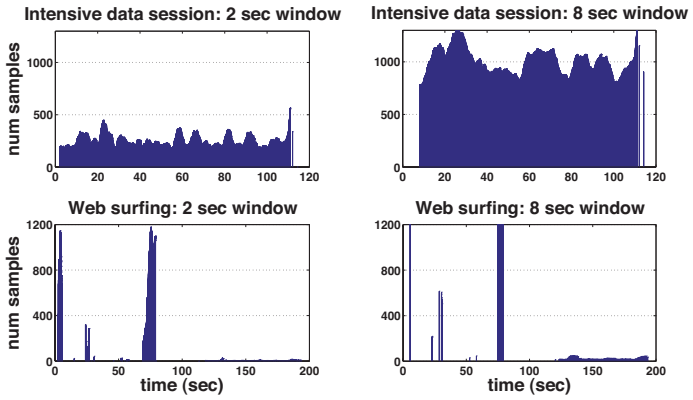

Fig. 9. Frame processed per *time-based* window.

from Figure 9, this is because the number of frames processed in a window differ for each window. Moreover it changes significantly between normal sessions (e.g. web surfing) and data intensive sessions. Therefore the sensor does not have to compute the variance on windows of a constant size (e.g. 2000 or 4000) and sometimes windows are even zero or few hundreds frames big.

6 Conclusions

In this work we have been investigating the problem of MAC-spoof detection and developed a method to detect it, based solely on the signal power of the wireless signal. The contribution of our work is the introduction of a *median filter* applied to the signal samples, and the detection mechanism, which is based on the variance of them along with the use of a timing reference. An attack is detected when the variance presents very high values. The purpose of the filter is twofold: it keeps variance from growing too much due to signal spikes and it helps to reject false positives. The variance has been computed over two different window types: time and number of frames. Experiments show that both give similar results, although the non-time based method cannot distinguish between mobile and static stations, and between data bursts and non-intensive traffic. Moreover *time-based* measures present lower variance values for normal traffic while MAC-spoof is still detected. Furthermore the method used is lightweight compared to the ones in the literature since it uses only simple functions namely the *median filter* and variance.

Acknowledgments

We gratefully appreciate the help and precious insights given by Prof. Robin Sharp from DTU Informatics.

References

1. P. Barsocchi, G. Oligeri, and F. Potortı. Validation for 802.11b wireless channel measurements. ISTI-CNR, 2006.
2. H. Bulbul, I. Batmaz, and M. Ozel. Wireless network security: Comparison of WEP (wired equivalent privacy) mechanism, WPA (wi-fi protected access) and RSN (robust security network) security protocols. In *Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop*, pages 1–6. ICST, 2008.
3. Y. Chen, W. Trappe, and R. Martin. Detecting and localizing wireless spoofing attacks. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON'07. 4th Annual IEEE Communications Society Conference on*, pages 193–202. IEEE, 2007.
4. Y. Chen, J. Yang, W. Trappe, and R. Martin. Detecting and localizing identity-based attacks in wireless and sensor networks. *Vehicular Technology, IEEE Transactions on*, 59(5):2418–2434, 2010.
5. D. Faria. Modeling signal attenuation in IEEE 802.11 wireless lans-vol. 1. *Computer Science Department, Stanford University*.
6. D. Faria and D. Cheriton. Detecting identity-based attacks in wireless networks using signalprints. In *Proceedings of the 5th ACM Workshop on Wireless Security*, pages 43–52. ACM, 2006.
7. R. Gill, J. Smith, and A. Clark. Experiences in passively detecting session hijacking attacks in IEEE 802.11 networks. In *Proceedings of the 2006 Australasian workshops on Grid computing and e-research-Volume 54*, pages 221–230. Australian Computer Society, Inc., 2006.
8. R. Gill, J. Smith, M. Looi, and A. Clark. Passive techniques for detecting session hijacking attacks in IEEE 802.11 wireless networks. *Proceedings of AusCERT*, pages 26–38, 2005.
9. F. Guo and T.-C. Chiueh. Sequence number-based MAC address spoof detection. *Lecture Notes in Computer Science : Recent Advances in Intrusion Detection*, pages 309–329, 2006.
10. A. Mishra, N. Petroni Jr, W. Arbaugh, and T. Fraser. Security issues in IEEE 802.11 wireless local area networks: a survey. *Wireless Communications and Mobile Computing*, 4(8):821–833, 2004.
11. Y. Sheng, K. Tan, G. Chen, D. Kotz, and A. Campbell. Detecting 802.11 MAC layer spoofing using received signal strength. In *INFOCOM 2008. The 27th Conference on Computer Communications*, pages 1768–1776. IEEE, 2008.
12. A. Tsakountakis, G. Kambourakis, and S. Gritzalis. Towards effective wireless intrusion detection in IEEE 802.11i. *Security, Privacy and Trust in Pervasive and Ubiquitous Computing. SECPerU 2007*, pages 37–42.
13. J. Wright. Detecting wireless LAN MAC address spoofing. *White Paper, January*, 2003.
14. J. Yang, Y. Chen, and W. Trappe. Detecting spoofing attacks in mobile wireless environments. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON'09. 6th Annual IEEE Communications Society Conference on*, pages 1–9. IEEE, 2009.
15. J. Yang, Y. Chen, W. Trappe, and J. Cheng. Determining the number of attackers and localizing multiple adversaries in wireless spoofing attacks. In *INFOCOM 2009, IEEE*, pages 666–674. IEEE, 2009.
16. J. Yeo, S. Banerjee, and A. Agrawala. Measuring traffic on the wireless medium: Experience and pitfalls, 2002.