Detecting Man-in-the-Middle and Wormhole Attacks in Wireless Mesh Networks

2009 International Conference on Advanced Information Networking and Applications

Detecting Man-in-the-Middle and Wormhole Attacks in Wireless Mesh Networks

Stephen Glass NICTA
stephen.glass@nicta.com.au
Vallipuram Muthukkumurasamy Griffith University
v.muthu@griffith.edu.au
Marius Portmann The University of Queensland
m.portmann@itee.uq.edu.au

***Abstract*—Wireless networks are being used increasingly in industrial, health care, military and public-safety environments. In these environments security is extremely important because a successful attack against the network may pose a threat to human life. To secure such wireless networks against hostile attack requires both preventative and detective measures.**

In this paper we propose a novel intrusion detection mechanism that identifies man-in-the-middle and wormhole attacks against wireless mesh networks by external adversaries. A simple modification to the wireless MAC protocol is proposed to expose the presence of an adversary conducting a frame-relaying attack. We evaluate the modified MAC protocol experimentally and show the detection mechanism to have a high detection rate, no false positives and a small computational and communication overhead.

I. Introduction

Preventative measures such as encryption and authentication are the principal line of defense against malicious attack. Unfortunately, attackers have proven to be adept at discovering and exploiting flaws in the design, implementation and operation of supposedly secure technologies. Other security threats are present because it is simply not cost-effective to prevent an attack. Intrusion detection forms a second line of defence which will alert users when an attack is under way. This is an effective approach when an attack cannot otherwise be prevented. Warning of an intrusion attempt can spur a change in defensive posture: prompting network operators to collect intelligence or forensic evidence, attempt to locate the adversary physically and take appropriate action.

The man-in-the-middle (or middleperson) attack is one in which legitimate parties communicate via a hostile adversary but without their knowledge or consent. This attack can be devastatingly effective because the adversary enjoys complete control of the communication link and can inspect, inject, delay, delete, modify and re-order traffic to suit their purpose. It may be used, for example, to bypass weak authentication protocols, hijack legitimate sessions, perform active traffic analysis and deny service. The wormhole attack presents a significant threat to the integrity of MANETs and wireless mesh networks (WMNs). A wormhole is a specialized man-in-the-middle attack in which the adversary connects two otherwise distant regions of the network. Stations adjacent to one end of the wormhole appear to be neighbours of stations at the other. The presence of the wormhole subverts the network topology and thus undermines the network routing algorithms. Routes through the wormhole benefit from lower hop-counts than legitimate routes and increase the probability that traffic will be routed via the adversary. Once the wormhole is established the adversary is able to conduct further attacks and do so with a low probability of detection.

This paper proposes a novel MAC-layer intrusion detection mechanism which can be used to detect frame-relaying (man-in-the-middle and wormhole) attacks against wireless networks. A small change to the wireless MAC layer detects the presence of the adversary even before they have proceeded to conduct other, more noticeable, attacks. The proposed mechanism is specific to wireless networks because it exploits the use of positive acknowledgment at the MAC layer.

A. Outline of the paper

The rest of this paper is organized as follows. A brief survey of related work is given in the next section. Section III describes positive acknowledgment and the proposed detection method in detail. Section IV describes the experimental testbed and section V the experiments themselves. In section VI an analysis of the results is presented. We conclude the paper in section VII.

II. Related work

There are a number of intrusion detection methods that are relevant to detection of man-in-the-middle and wormhole attacks. Buttyán, and Hubaux survey several techniques for wormhole detection [1, Chapter 6] and identify the problem underlying these attacks as a failure to ensure the authenticity of neighbouring stations.

A. Fingerprinting

Fingerprinting is an approach for ensuring the authenticity of corresponding stations. It relies on measurement of one or more characteristics of the legitimate stations that cannot be spoofed by an adversary. One particularly interesting approach is to make use of incidental characteristics of the radio transmission to distinguish between different network transceivers [2], [3]. Once recorded, these "transceiverprints"

1550-445X/09 $25.00 © 2009 IEEE
DOI 10.1109/AINA.2009.131

or "signalprints" can then be used to identify impersonation attacks (session hijacking, wormhole or man-in-the-middle attacks). Gill *et al.* suggest a similar approach but avoid the a-priori recording by looking for sudden deviations in received signal strength and round-trip RTS/CTS timing which are presumed to be indicators of a possible session hijacking attack [4]. Korkmaz applies a similar approach to the wormhole problem and uses time-of-flight and signal-power models as part of a neighbour-verification protocol (NVP) [5]. Unfortunately, signal strength can vary significantly in a dynamic mobile environment and time-of-flight maybe hard to determine accurately. Fingerprinting does not, therefore, appear to be suitable for dynamic mobile environments.

B. MAC layer monitoring

Wireless Intrusion Detection Systems (WIDS) make use of rule and signature-based approaches to detect intruders. One set of heuristics identifies attackers using sequence numbers. Many first-generation wireless network interfaces required the use of special modes or race conditions to inject arbitrary traffic. This meant the attacker could not control the sequence number of the injected frames. A scheme proposed by Wright [6] and refined by Guo and Chiueh [7] uses knowledge of this limitation to detect some impersonation attacks. Alternatives to signature-based approaches are also being actively investigated. One approach applies data-mining techniques to wireless network traffic to detect an attack or an anomaly [8]. Identifying events that are mutual outliers may indicate an impending threat.

C. Protocols for detecting intruders

There are several protocols that can expose an attacker. One well-known mechanism for exposing a man-in-the-middle is Rivest and Shamir's interlock protocol [9]. This simple protocol, and derivitives of it, can expose a man-in-the-middle but only if they attempt to subvert the key exchange. Wormhole attacks do not require a breach of authenticity or confidentiality and so this mechanism is of limited use in preventing or detecting this attack.

The problem of wormhole attacks was first discussed in a paper by Hu *et al.* which suggests the use of packet leashes as a remedial measure [10]. Packet leashes restrict the travel of a packet within a tightly-defined geographical area but require either trustworthy geographical data or precisely synchronized clocks. Hu *et al.* suggest that GPS receivers be used satisfy these requirements but this merely exchanges one problem for another — civilian-use GPS receivers make use of unauthenticated signals and are subject to well-known attacks that jam or spoof satellite signals.

Brands and Chaum were the first to suggest distance-bounding protocols which can limit the distance between legitimate parties by using precise timing of a cryptographic challenge/response [11]. In a distance-bounding protocol the round-trip delay time is constrained by the speed of light. An adversary cannot, therefore, appear to be closer than they really are but can delay responses to appear further away. Distance-bounding protocols can constrain the flight of frames within a limited geographical range thus greatly reducing the threat posed by the wormhole attack. One distance-bounding protocol for wireless networks is SECTOR's MAD (Mutual Authentication with Distance-bounding) protocol [12]. MAD can be used as a defence against frame-relaying attacks but requires special hardware support in the form of a fast, low-latency channel. MAD is not suited to the high-latency half-duplex transmission schemes commonly used in commercially available wireless networks.

An alternative approach is advocated by Eriksson in the form of the TrueLink protocol [13]. TrueLink is not a true distance-bounding protocol but it can be used in many situations to establish the authenticity of neighbouring stations and so applies to the general case of frame-relaying attacks. The protocol has two phases. The first phase exchanges RTS/CTS frames which contain a nonce. The timing constraints on the RTS/CTS are such that a hostile adversary cannot relay these frames at the MAC layer. In the second phase nonce values are used to answer periodic authentication challenges which are not time-critical but prove the nonces to be authentic. Detection depends on the frequency of these exchanges. This approach requires only a minor change to the MAC protocol and can be used with widely-used half-duplex wireless hardware. TrueLink has been validated in simulation but not in practice. The requirement for public-key cryptography to validate the nonce exchange imposes a computational overhead that limits the frequency of challenges in applications where computational resources are sparse.

D. Other detection techniques

Statistical approaches may detect and localize a wormhole. The presence of a wormhole always increase number of links of stations adjacent to the wormhole [14] and the hop latency of links traversing the wormhole will be higher than for conventional links [15]. Visualization of the network topology can also be used to identify wormholes [16]. These techniques require complete or non-local knowledge of network topology and so appear better suited to wireless sensor networks than to WMN and MANETs.

III. Proposed detection method

Wireless networks experience much higher error rates than is the case in wired networks. A common design feature of wireless MACs is the use of positive acknowledgment to address this problem. Positive acknowledgment requires that stations must transmit an acknowledgment when it receives a data frame succesfully and must do so within strict time constraints. Our intuition is that positive acknowledgment can be used to expose many frame-relaying attacks.

A. Positive acknowledgment

Positive acknowledgment presents a serious problem for successful frame-relaying attacks. Although the attacker has almost complete control of message traffic they must take care to ensure that ACKs are received before the acknowledgment timeout expires. Failing to do so will alert the sender that the

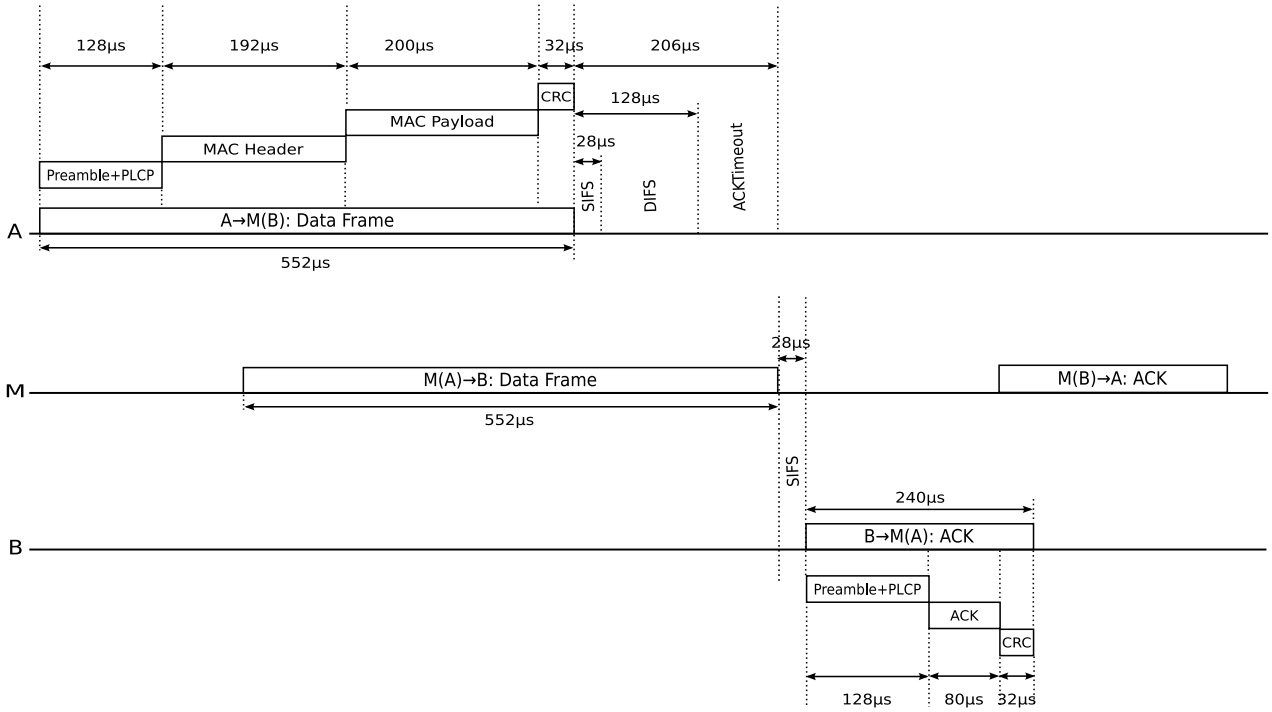


Fig. 1. Timing diagram for message exchange in IEEE 58802.11b

frame has not been received causing the sender to retransmit and, eventually, signal an error to the upper layers. The problem for the adversary is that it is often impractical to meet the acknowledgment timing constraints by relaying frames between the legitimate stations. When the round-trip latency introduced by the attacker exceeds the acknowledgment time-out it becomes impossible to do so.

Figure 1 illustrates the adversary's problem with a timing diagram for a single message transfer in an IEEE 802.11b network. In this example A sends a 25 octet message to B using FHSS modulation at the base rate of 1Mb/s. The transfer is actually accomplished by the adversary M which relays the message and then the acknowledgment in turn. M waits only until the first ten octets of the MAC header have been seen (i.e. enough to determine the frame type and destination) before deciding to forward the frame and this entails a necessary delay of $208\mu s$ for each frame. An additional delay is caused by the SIFS introduced by B and so the resulting ACK is not received at A until $238\mu s$ after the ACKTimeout has expired. Relaying acknowledgments in this manner cannot, therefore, meet the protocol requirements for timely acknowledgment.

In practice the situation is actually more difficult for the adversary. The IEEE 802.11 standard requires that the ACKTimeout should be calculated according to the following equation [17, §9.2.8]:

$$ACKTimeout = t_{SIFS} + t_{Slot} + t_{RXStartDelay} \quad (1)$$

which, for the example used above, gives a value of $206\mu s$. Many wireless interfaces fail to comply with this requirement and instead use a fixed value. MadWifi-NG, for example, normally fixes its ACKTimeout value at $48\mu s$.

A successful attack must, therefore, ensure that the ACK is received within the ACKTimeout period. In practical attacks this is achieved by having M itself send an ACK whenever it successfully receives a frame for A or B. All that needs to be done is to modify the interface address and BSSID mask. Reconfiguring the wireless interface in this way allows the adversary to meet the timing constraints but the adversary sends an ACK *before* it has received an ACK from the final destination. This property can be used to expose the presence of a frame-relaying attack.

B. Detection strategy

If sender and receiver secretly agree that certain frames must not be acknowledged on their first transmission it becomes possible to detect a frame-relay attack which acknowledges when a legitimate station would have remained silent. The following paragraphs describe how this can be implemented within a modified IEEE 802.11 MAC layer. The scheme can, in principle, be applied to other wireless network technologies that use positive acknowledgment.

C. MAC layer changes

The first MAC layer change extends the procedure responsible for sending encrypted, unicast data frames. The 802.11 security protocols guarantee the integrity, authenticity and replay-protection only for encrypted data frames. The intrusion detection measure is, therefore, applied only for these

frame types in order to prevent an adversary using an active attack to discover the shared secret. The modified SEND-DATA procedure is shown in figure 2 which adds a test (at lines 8+9) to detect inauthentic acknowledgments.

```
SEND-DATA(frame)
 1  n ← 1
 2  CLEAR-RETRANSMIT-FLAG(frame)
 3  ack ← TRANSMIT(frame, ACKTIMEOUT)
 4  while ack = NIL and n ≤ max-attempts
 5        do n ← n + 1
 6           SET-RETRANSMIT-FLAG(frame)
 7           ack ← TRANSMIT(frame, ACKTIMEOUT)
 8  if ack ≠ NIL and SUPPRESS-ACK?(frame)
 9     then error   ▷ ACK is not authentic
10     else ...
```

Fig. 2. Modified SEND-DATA procedure

The second change is made to the corresponding receive procedure and suppresses the initial ACK when required. To do this an additional test is introduced (lines 1+2) as shown in figure 3.

```
RECV-DATA(frame)
1  if SUPPRESS-ACK?(frame)
2     then ▷ No acknowledgment
3     else TRANSMIT(ACK)
4          ...
```

Fig. 3. Modified RECV-DATA procedure

The SUPPRESS-ACK? function is introduced so that sender and receiver can agree as to whether the initial ACK is suppressed. The function must do so in a manner that cannot be predicted by an adversary who is actively interfering with communication. We assume that sender and receiver share a secret that is unknown to the adversary and this is used as shown in figure 4.

```
SUPPRESS-ACK?(frame)
1  suppress ← false
2  if RETRANSMIT-FLAG-CLEAR?(frame)
3     then h ← HMAC(SHA-1, frame, shared-secret)
4          x ← h ∧ bitmask
5          if x = 0
6             then suppress ← true
7  return suppress
```

Fig. 4. The SUPPRESS-ACK? function

This function checks the frame header to ensure that the retransmit flag is clear before computing a keyed hash value[1] $h \in \{0,1\}^m$ for the frame contents. A subset $x \in \{0,1\}^n$ of h is taken and only if each bit in x is zero is the ACK be suppressed. Thus, if a uniform distribution of hash values is assumed, the probability of a packet having its initial ACK suppressed is $P(\frac{1}{2^n})$. Without knowledge of the key the adversary cannot predict which frames should not acknowledged.

[1]Throughout our discussion HMAC(SHA-1) is used but any secure symmetric message authentication code may be used instead.

IV. Equipment and Preparation

The following paragraphs describe the test network that is established to test the performance of the modified MAC protocol under experimental conditions.

A. Test equipment

The modified MAC protocol works on a hop-by-hop basis. Each station is equiped with a single wireless network interface and routing between the stations is statically configured. To control for the effects of routing strategy and other stations a single link between two network stations A and B is tested.

The adversary M conducts a frame-relay attack and uses a dedicated laptop computer equipped with two wireless network interfaces. Frames are forwarded from one interface to the other under program control. This attack can be considered to be a wormhole with an optimal high speed link between its endpoints. The adversary faithfully forwards traffic and implements no overtly hostile behaviours by which its presence may be detected.

Experimental data is collected at a separate monitoring station using a single wireless network interface and are logged to file using the `tcpdump` network sniffer. The monitor uses an Intersil PRISM2-based 802.11b wireless interface because it records the time of arrival of each frame with a resolution of $1\mu s$.

All of the computers used in the experiments use the Debian GNU/Linux operating system and Linux kernel version 2.6.24. The wireless network interfaces used for all stations (except the monitor) are 802.11a/b/g multi-standard devices based on the Atheros AR5213 chipset. These are used with a modified version of the 0.9.4 release of the MadWifi-NG device driver. This chipset has been selected because it allows for direct program control of many wireless interface functions.

B. MAC implementation

The implementation of the modified MAC protocol presents one of the most difficult challenges of the experimental setup. In commodity 802.11 wireless equipment the behaviour is implemented in hardware or interface firmware and is not under program control. The implemntation requires that the hardware mechanism is disabled and that the ACK is generated by software in response to a received frame.

A modified MadWifi-NG device driver is used[2] that exposes an `iwpriv` command to allow the hardware or software acknowledgment to be selected from the user level. Hardware acknowledgment is disabled by writing a bit into the wireless interface PCU control register. Generating ACKs in software requires careful attention to minimize latency. The

[2]The modified MadWifi-NG driver is available from the principal author.

driver has a modified interrupt service routine which issues an acknowledgment in response to the receipt of a unicast frame. A dedicated transmit queue internal to the wireless network interface hardware is used to ensure ACKs are injected after waiting only for one SIFS period and without the usual exponential back-off behaviour.

C. Attack implementation

A user-mode program is used to transfer frames between interfaces used by the attacker. Not all frames need to be forwarded — frames to/from stations other than those under attack should not be forwarded and control frames such as RTS/CTS exchanges need not be forwarded. A Berkeley Packet Filter program is installed in the kernel and used to discard such frames without the intervention of the user-mode program. Fixing the transmit rate for the experiments avoids having to adopt a rate-control control algorithm within the attack implementation in addition to minimizing the effects of implementing the modified MAC acknowledgment protocol in software.

The frame relay strategy adopted for this experiment is not as efficient as the one outlined in section III. That example was intended to show the theoretical necessity for an adversary to generate acknowledgments for frame relaying attacks at the MAC layer. The only supported mode of operation for commonly available wireless network interfaces is to wait until the whole frame has been received before the adversary can forward it to the final destination. This introduces additional latency into the exchange. If the 25 octet message example outlined above has to be received in full before forwarding then the ACK would not begin to arrive at A until $614\mu s$ after the ACKTimeout has expired. This underscores the practical necessity for the adversary to generate the ACK themselves.

V. Experiments

The experimental investigation is intended to provide a basis for comparison betwen the modified and original MAC protocols and to determine the cost and effectiveness of the modified MAC protocol.

1) Performance comparison: Software-generated acknowledgment has the potential to incur a substantial performance penalty. To establish a basis for comparison The `iperf` program is used to generate a ten-second burst of UDP traffic, this repeated five times. The average of the five tests is used to account for environmental variations. The results collected for three configurations:

- using the hardware ACK and the standard MAC protocol,
- using a software-generated ACK with the standard MAC protocol, and
- using a software-generated ACK with the modified MAC protocol and $n = 63$.

These three configurations allow for the overhead of implementing acknowledgment in software to be separated from the cost of the modified MAC protocol. Each test is repeated with the data rate locked at 1Mb/s, 5.5Mb/s and 11Mb/s and the ACKTimeout fixed at the hardware maximum of $744\mu s$. The latency introduced by the software implementation is expected to be relatively stable and as the rate increases the overhead it imposes on bandwidth should increase.

2) Detection rate: To identify how closely the modified MAC protocol comes to the theoretical detection rate stations A and B are taken outside their mutual radio range and communicate via the adversary M. In this case M filters control frames and generates its own ACK frames. The modified MAC protocol allows for an ACK to be suppressed with a probability of $P(\frac{1}{2^n})$. The `iperf` program is used to generate a fifteen-minute burst of UDP traffic and the procedure is repeated for values of n at 4, 5, 6, 7 and 8. Detection of the adversary is achieved whenever the adversary acknowledges a frame when they should have remained silent.

3) Frame-relay strategies: To identify the effects of modified acknowledgment strategy a simple experiment is used. The `iperf` program is used to generate a sixty-second burst of UDP traffic and the results collected for three configurations:

- a simple configuration of legitimate stations A and B with no man-in-the-middle,
- station A and B are taken outside their mutual radio range and communicate via the adversary M employing simple frame relay strategy and
- stations A, B and M as above but where the adversary filters control frames and generates their own ACK frames.

For this experiment all traffic is exchanged at a maximum data rate of 1Mb/s to control for the effects of the rate control algorithm. From the explanation given in section III it is expected that the simple frame-relay strategy will not be able to return ACKs inside the ACKTimeout time window. This should demonstrate the necessity for the adversary to generate acknowledgments for a frame-relay attack to succeed.

VI. Analysis of Results

The two key measures of the effectiveness of the modified MAC protocol are the detection rate and the cost of the modified MAC protocol in terms of reduced bandwidth. An ideal detection scheme will produce no false positives, detect all attacks and impose no additional computational or bandwidth overhead. The following paragraphs consider these factors and other important aspects of the modified MAC protocol.

A. Cost of the modified MAC protocol

The modified MAC protocol introduces a new source of frame loss into the channel and will result in an increased number of re-transmissions. The reduction in throughput is summarized by the results in table I.

The results show a clear drop in throughput that is attributable to the implementing the MAC protocol in software. The software implementations reduce throughput to between 83% and 90% of that achieved by the hardware. A much smaller loss can be attributed to the modified MAC protocol which only reduced the bandwith to 98% of that achieved by the software implementation of the standard 802.11 MAC.

An anomoly is that at 11Mb/s the modified MAC protocol at times appears to outperform the standard 802.11 protocol. The modified MAC protocol has to do more computational work than the conventional protocol and so it is to be assumed

TABLE I
EXPERIMENT THROUGHPUT RESULTS

	Rate (Mb/s)	Client (Kb/s)	Server (Kb/s)	Transfer (KB)
Hardware 802.11	1	923.0	885.2	1112.0
	5.5	4066.0	4028.0	4874.0
	11	7316.0	7270.0	8736.0
Software 802.11	1	799.2	762.6	992.8
	5.5	3692.0	3650.0	4430.0
	11	6120.0	6080.0	7312.0
Software Modified MAC	1	794.6	760.4	991.6
	5.5	3630.0	3594.0	4342.0
	11	6252.0	6210.0	7462.0

that this will usually take longer than doing no computation. Two of the five tests for the software implementation of the standard 802.11 protocol performed more slowly than for the modified MAC protocol and thus brought down the average. The radio environment in which the tests took place is uncontrolled and there are other IEEE 802.11 networks that are also contending for the channel and the presence of such traffic is correlated with these slower results. This suggests that the additional overhead of the modified protocol is negligible and that other factors (such as channel-access contention) are more significant. Further testing using co-axial cabling and signal attenuators is planned to control for environmental interference.

One performance-related consideration is the effect of suppressing acknowledgments on higher layers. `iperf` reported no packets lost for any of the experimental tests. The retransmissions incurred at the MAC layer were completely sufficient to avoid packet loss at higher layers.

B. Detection rate

The detection rate of the modified MAC protocol is controlled by the choice of value for n. Assuming a uniform distribution of values from the secure hash function an adversary will, on average, be detected after 2^{n-1} frames. Figure 5 plots the frequency distribution of distance between suppressed ACKs for each of the experimental values of n. The main aim of this is to ensure that the adversary cannot easily determine if an ACK needs to be suppressed or not. Our proposed algorithm given in Fig 5 is sufficient to guarantee this. As can be seen from figure 5 (a) to (e), the selection of the value of n also makes the distribution of the suppression of ACK vary. The full impact of the selection of value of n in terms of security and link throughput needs to be examined further in detail.

C. False negatives

An adversary might avoid detection if they do not acknowledge a frame for which the ACK would be supressed. In noisy environments it is possible probably that a station will miss the first transmission of a frame or its ACK may not be successfully received. In this case, the adversary may escape detection for longer because interference causes it to fail to receive the initial transmission or because it fails to acknowledge a packet due to interference. It does, however, require the detection of only a single ACK in violation of the protocol to expose an attack.

An adversary may turn this to their advantage and choose not to reply to the first transmission of any frame. The presence of the retry bit in the frame header provides sufficient information as to whether they should reply or not. Although this evades the modified MAC protocol's principal detection measure it is also extremely noticeable in and of itself. An alarm condition for stations exhibiting such behaviour is, therefore, a necessary adjunct to the principal alarm.

D. False positives

The false alarm rate is a measure of the expected rate of false positives for a given intrusion detection mechanism. An intrusion detection system that generates false positives will reduce the users' confidence in the alarms and lead to them being ignored. The false alarm rate, therefore, provide a critical measure of the real-world usefulness of an intrusion detection method. The modified MAC protocol cannot generate any false positives by design. Legitimate stations decide which acknowledgments must be suppressed based only on the contents of the frame and of the shared secret. The reception of an acknowledgment that should have been surpressed can *only* occur because it was generated by a station other than the legitimate one.

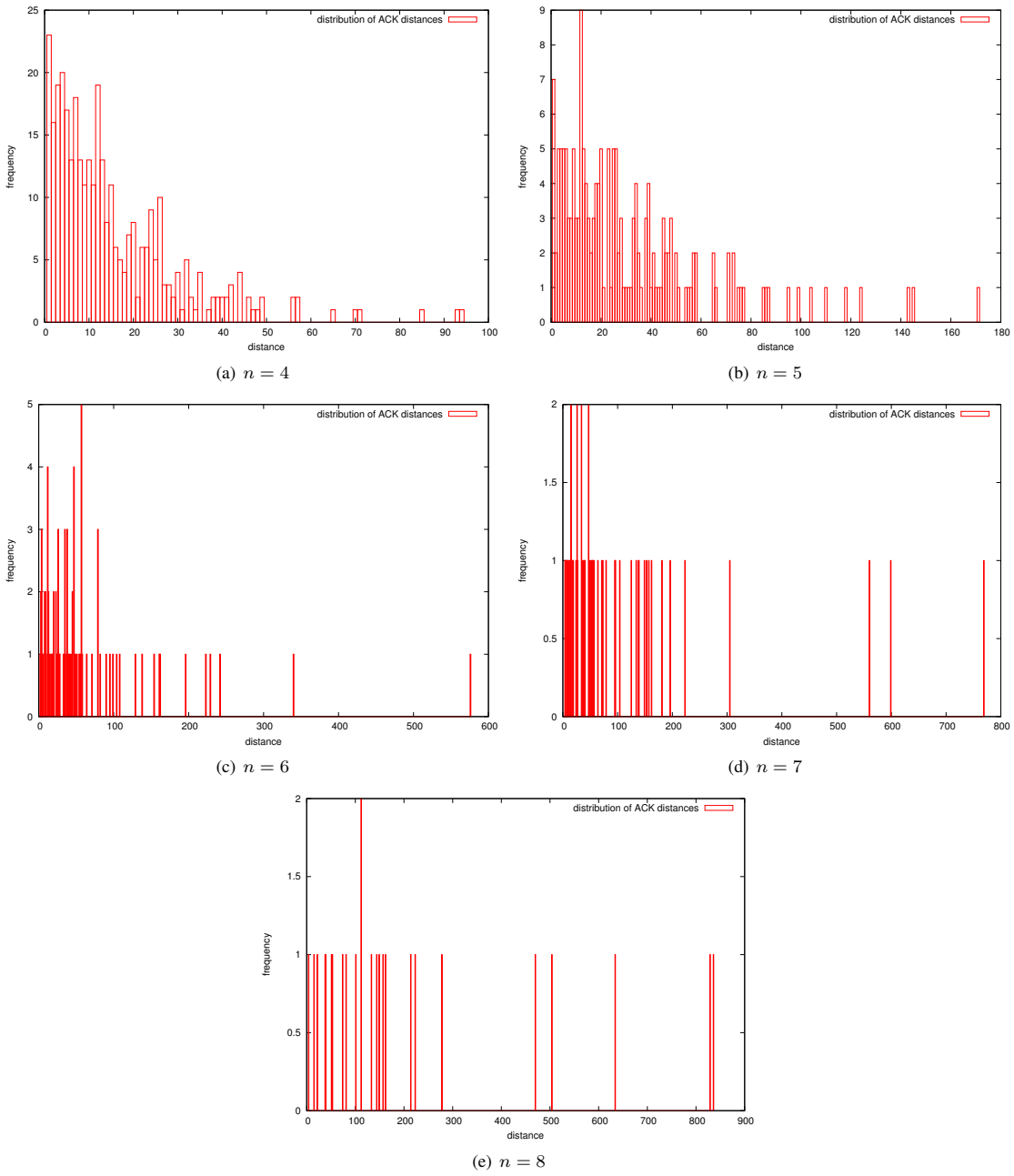
E. Cryptographic attacks

The modified MAC protocol is an extension of the security protocol that depends on a secret shared by the legitimate parties. This key can be established by the security protocol and exchanged as part of the key distribution process (in IEEE 802.11 this is the four-way handshake). The `suppress-ACK?` function is called only for unicast, encrypted traffic in order to ensure the authenticity of the frame and prevent replay attacks. In this paper the algorithm used to compute the keyed hash value is HMAC(SHA-1) and there is good reason to doubt the continuing security of SHA-1[18]. The use of HMAC permits the easy selection of a secure replacement and the selection of an appropriate hash function is simply an engineering decision based on ease of computation for the defenders versus the work effort for an adversary.

F. Impact on routing

Re-transmissions caused by the modified MAC protocol are essentially invisible to the upper layers (including routing) but the impact of discovering an attack can also be devastating. Once legitimate stations have become aware that a frame-relay attack is being conducted on a given link they can invalidate routes that use that link and issue a new route request (this time ignoring any routes that use the adversary-provided link).

The problem is that the wormhole will forward the frames encapsulating the route request packets and these may reach the destination more quickly via the wormhole than by traversing legitimate links. Many reactive routing protocols suppress duplicate routing requests and so discard those that arrive after



(a) $n = 4$ (b) $n = 5$

(c) $n = 6$ (d) $n = 7$

(e) $n = 8$

Fig. 5. Frequency distribution of suppressed-ACK distance

the one via the wormhole. This is the same as would happen in a rushing attack [19] and can prevent the establishment of an alternative route.

G. Impact on QoS

IEEE 802.11e introduced the concept of different access classes and block acknowledgment into 802.11. Access classes separate and prioritizes traffic into voice, video, best-effort and background types. In themselves these present no problem for the modified MAC protocol but the concept of block ACK does. A block ACK allows a number of frames separated only by a SIFS and with an ACK for the whole group. The modified MAC protocol is not suitable for use with block ACK because it relies on low-cost re-transmission.

H. Impact on maximum link distance

An essential security constraint of the modified MAC protocol is that the ACKTimeout must be short enough to deny an adversary sufficient time to relay acknowledgments between stations. For long-distance links the ACKTimeout must also

be large enough to allow for the transmission time of the data frame and its acknowledgment and for any protocol-mandated delays at the receiver.

In IEEE 802.11-based WMNs the ACKTimeout is governed by equation 1 in which the term t_{Slot} is the slot time and has the effect of governing the maximum link distance. In 802.11 the slot time value is fixed at either $9\mu s$ or $20\mu s$ and allows a maximum theoretical link distance of approximately 1350m and 3000m respectively. The slot time values are much smaller than the transmission times and the security constraint can be satisfied but the modified MAC protocol described here is unsuitable for situations in which there are link distances of 1km or more between stations.

I. Physical layer attacks

An objection to the proposed MAC protocol is the assumption that an adversary conducting a frame-relay attack will take special measures to handle positive acknowledgment. Table II summarizes the result of using the simple frame-relay strategy to one in which the adversary generates acknowledgments. In this case the strategy makes all the difference between a successful attack and failure. In the case of the simple frame relay strategy the late arrival of the ACKs results in a complete failure to establish a channel across which testing could take place.

TABLE II
COMPARISON OF ACK STRATEGIES

	Rate (Mb/s)	Client (Kb/s)	Server (Kb/s)	Transfer (Mb)
Relayed ACK	—	—	—	—
Self-Generated	1	889	583	6.38

An alternative approach is to mount the attack at the physical layer. In this case the adversary uses a radio repeater - receiving the radio waveform, amplifying and retransmitting it onwards to its legitimate destination. The problem with such a physical layer attack is that the frame must contend for access to the channel twice: once at the originating station and again when departing the adversary. If the adversary finds the onward channel busy then they are obliged to go through the exponential back-off procedure. Should the channel be busy this requires that the frame waits for at least a DIFS and p slots (where CWMIN $\leq p \leq$ CWMAX) before re-transmission. Links subject to a frame-relay attack will, therefore, experience more contention and a higher number of retransmissions than ordinary links.

The simplest way for an adversary to preserve link quality and evade detection would be to buffer the waveform and generate their own acknowledgments. Thus, even some physical re-transmission attacks might be discovered by the modified MAC protocol.

VII. CONCLUSIONS

We have presented a novel modified MAC-layer protocol for the detection of frame-relaying (man-in-the-middle and wormhole) attacks in wireless networks. This method exploits the positive acknowledgment property often used in wireless networks to expose the presence of an adversary conducting a MAC layer man-in-the-middle or wormhole attack. The proposed modification has general applicability and is suitable for MANETs, wireless mesh and infrastructure networks. Our analysis suggests that this method enjoys a very high detection rate with no false positives at the cost of a very small loss of bandwidth.

VIII. ACKNOWLEDGMENTS

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program; and the Queensland Government.

REFERENCES

[1] Levente Buttyán and Jean-Pierre Hubaux. *Security and cooperation in wireless networks*. Cambridge University Press, Cambridge, England, 2008.

[2] Michel Barbeau, Jeyanthi Hall, and Evangelos Kranakis. Detecting impersonation attacks in future wireless and mobile networks. In Mike Burmester and Alec Yasinsac, editors, *MADNES*, volume 4074 of *Lecture Notes in Computer Science*, pages 80–95. Springer, 2005.

[3] Daniel B. Faria and David R. Cheriton. Detecting identity-based attacks in wireless networks using signalprints. In *WiSe '06: Proceedings of the 5th ACM workshop on Wireless security*, pages 43–52, New York, NY, USA, 2006. ACM Press.

[4] Rupinder Gill, Jason Smith, and Andrew Clark. Experiences in passively detecting session hijacking attacks in IEEE 802.11 networks. In *ACSW Frontiers '06: Proceedings of the 2006 Australian workshops on Gri Computing*, pages 221–230, Darlinghurst, Australia, 2006. Australian Computer Society, Inc.

[5] Turgay Korkmaz. Verifying physical presence of neighbors against replay-based attacks in wireless ad hoc networks. In *ITCC '05: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II*, pages 704–709, Washington, DC, USA, 2005. IEEE Computer Society.

[6] Joshua Wright. Detecting wireless LAN MAC address spoofing. Available from http://home.jwu.edu/jwright/papers/wlan-mac-spoof.pdf, January 2003.

[7] Fanglu Guo and Tzi-cker Chiueh. Sequence number-based MAC address spoof detection. In Alfonso Valdes and Diego Zamboni, editors, *8th International Symposium on Recent Advances in Intrusion Detection (RAID 2005)*, volume 3858 of *Lecture Notes in Computer Science*, pages 309–329. Springer, September 7–9 2005.

[8] Elankayer Sithirasenan and Vallipuram Muthukkumarasamy. Detecting security threats in wireless LANs using timing and behavioral anomalies. In *15th IEEE International Conference on Networks*, pages 66–71. IEEE, November 2007.

[9] Ronald L. Rivest and Adi Shamir. How to expose an eavesdropper. *Communications of the ACM*, 27(4):393–394, 1984.

[10] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Wormhole attacks in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(2):370–380, February 2006.

[11] Stefan Brands and David Chaum. Distance-bounding protocols. In *EUROCRYPT '93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, pages 344–359, Secaucus, NJ, USA, 1993. Springer-Verlag New York, Inc.

[12] Srđan Čapkun, Levente Buttyán, and Jean-Pierre Hubaux. SECTOR: secure tracking of node encounters in multi-hop wireless networks. In *SASN '03: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, pages 21–32, New York, NY, USA, 2003. ACM Press.

[13] Jakob Eriksson, Srikanth V. Krishnamurthy, and Michalis Faloutsos. TrueLink: a practical countermeasure to the wormhole attack in wireless networks. In *14th Annual IEEE conference on network protocols (ICNP 2006)*, pages 75–84. IEEE Computer Society, 2006.

[14] Levente Buttyán, László Dóra, and István Vajda. Statistical wormhole detection in sensor networks. In *Security and Privacy in Ad-hoc and Sensor Networks*, volume 381 3/2005 of *LNCS*, pages 128–141. Springer Berlin/Heidelberg, 2005.

[15] Phuong Van Tran, Le Xuan Hung, Young-Koo Lee, Sungyoung Lee, and Heejo Lee. TTM: An efficient mechanism to detect wormhole attacks in wireless ad-hoc networks. In *4th IEEE Consumer Communications and Networking Conference (CCNC 2007)*, pages 593–598. Institution of Electrical and Electronics Engineers, Institution of Electrical and Electronics Engineers, January 2007.

[16] Weichao Wang and Bharat Bhargava. Visualization of wormholes in sensor networks. In *WiSe '04: Proceedings of the 3rd ACM workshop on Wireless security*, pages 51–60, New York, NY, USA, 2004. ACM.

[17] LAN/MAN Standard Committee of the IEEE Computer Society. *IEEE Standard for Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. Institution of Electrical and Electronics Engineers, 802.11-2007 edition, June 2007.

[18] Xiaoyun Wang, Yiqun L. Yin, and Hongbo Yu. Finding collisions in the full SHA-1. In *Advances in Cryptology CRYPTO 2005*, volume 3621 of *LNCS*, pages 17–36. Springer Berlin/Heidelberg, November 2005.

[19] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In *WiSe '03: Proceedings of the 2003 ACM workshop on Wireless security*, pages 30–40, New York, NY, USA, 2003. ACM Press.