

Concerns with Time-Space Based Wireless Security

Richard Lundeen

University of Idaho / Idaho State University
921 South 8th Ave stop 8060, Pocatello, ID, 83209

lundrich@isu.edu

Abstract — Wireless ad-hoc network protocols are a topic of much recent discussion and development. This has prompted many researchers to develop interesting and promising-sounding protocols that should be considered and examined. One such protocol, Authenticated Protocol for Wireless Ad Hoc Networks (APEC), was designed by Robert Hiromoto and Hope Forsmann[1]. APEC has been the subject of an increasing amount of scientific discussion and research around Universities, Laboratories, and professional conferences. In this paper, we examine APEC in depth and discuss many potential problems with the protocol that must be addressed if APEC is to achieve widespread acceptance.

Keywords — Ad Hoc Networks, Networking, Mesh, Sensor Networks, Network Security

1. INTRODUCTION

One problem that has attracted a lot of attention from researchers recently is applying security to ad-hoc wireless mesh networks. Some of the most prominent research in this area has been achieved with the 802.11s proposed standard and subsequent implementations in projects like the One Laptop per Child project. Today, an implementation of 802.11s has been accepted into the official Linux kernel [2].

Despite or perhaps spurred by such rapid progress, there are many proposed competing protocols emerging. One such protocol is an interesting approach called An Authenticated Protocol for Wireless Ad Hoc Networks with Embedded Certificates (APEC) [1].

This paper will examine many problems with APEC. It is outlined as follows: in section 2, current security is discussed to give APEC a baseline for performance and/or security. In sections 3 and 4, the time and space components of APEC are analyzed. In section 5, possible attack scenarios are discussed.

2. CURRENT SECURITY FOR AD-HOC NETWORKS

The IEEE 802.11s draft standard uses Efficient Mesh Security Association (EMSA) to prevent non-authorized devices from sending or receiving traffic on an ad-hoc network. EMSA uses the 802.11i authentication model which includes link level authentication, key distribution, and encryption.

802.11i uses one of the most peer reviewed symmetric block ciphers - AES. For authentication, it makes use of 802.1X. Though 802.1X is extensible and there exists the

option to authenticate to a central authority (eg a RADIUS server) APEC is more comparable to Pre-Shared Key (PSK) authentication since in both cases the key needs to be shared prior to deployment [3].

The biggest difference between multi-hop mesh networks and more common 1 hop networks is that mesh APs must act as both authenticators and supplicants. When a node attempts to join the mesh network, roles between it and additional nodes are established. If a node can reach the authentication server (AS) and the other cannot, the AS connected node becomes the authenticator. If both can reach the AS, one of the nodes is chosen to be the authenticator, the other becomes the supplicant [4]. After the roles have been established, the two nodes perform the 802.11i handshake [5].

To have success, a proposed protocol would have to have sufficient advantage when compared with the current IEEE standards and proposed standards. This advantage must be demonstrable in the form of greater security or increased performance.

3. CLOCK SYNCHRONIZATION

For a time-space protocol like APEC to be functional, it is necessary to have extremely accurate time, and the clocks of the various nodes must be kept in sync. There are several constraints that must be considered if a solution is to be either practical or beneficial.

- Any time synchronization solution must be cost-effective. It must cost less by some metric than the more popular 802.11 protocols.
- Any time synchronization solution must be secure. The system must provide some inherent security that *does not* require public key computation (either authentication or encryption) or uses *less* computation than a more accepted protocol like 802.11i.

In this section, two possible solutions are examined. First, time protocols are examined to be ineffective because although cost-effective they have no inherent security. Second, GPS is examined to be ineffective because it is not cost-effective and also has no inherent security.

3.1 Time Protocols

3.1.1 Network Time Protocol

Network Time Protocol (NTP) is one of the most popular network protocols for distributing accurate time over packet-switched networks. It is based on Marzullo's algorithm, which is particularly resistant to the effects of variable latency (jitter) [6].

The accuracy of NTP varies and is dependent on the predictability of network delays. Over the public Internet it is usually able to maintain time within about 10 milliseconds. Under ideal conditions on a local area network, it can achieve accuracies of 200 microseconds or better.

NTP contains provisions to cryptographically authenticate individual servers. Without this, NTP is insecure and accurate time could be corrupted any number of ways by a malicious node [7].

3.1.2 Precision Time Protocol

Precision Time Protocol (PTP) defined in [8], can achieve accuracy within a few microseconds on a wireless LAN (several orders of magnitude above NTP) using hardware generated timestamps [9].

The protocol contains rapid configuration, fault tolerance, and both broadcast and singlecast options. Although a fairly new protocol, PTP shows promise and is already implemented in several popular systems [10].

3.1.3 Reference Broadcast Synchronization

Reference Broadcast Synchronization (RBS) is a prototyped time synchronization protocol built specifically for low-power wireless networks. Under ideal conditions, it can synchronize time at the microsecond level [11]. The biggest limitation of RBS is that it requires a broadcast domain. However, with ad-hoc networks this is not an issue.

The Timing Sync Protocol for Sensor Networks (TPSN) also deserves mention, as it is also specifically built for similar networks and has a unique way of synchronizing time. The performance, however, is similar to RBS. The authors claim it has double the the performance of RBS [12].

3.1.4 Conclusions

All of these protocols, and all network protocols in general, share a single important characteristic: they all require cryptography for security.

To use a time protocol for a cryptographic seed is a chicken-egg problem. Although almost all the protocols above have the ability to distribute accurate time over a packet-switched network in a secure manner, this itself requires authentication. Obviously, this defeats much of the purpose if the ultimate goal is to use a time-space protocol to avoid the computational costs of public key authentication.

3.2 GPS

GPS is given as the assumed solution in [1]. Although GPS provides greater accuracy than most network based protocols, GPS has many of the same issues as these protocols that must be addressed. In addition, it can be both cost prohibitive and inaccessible.

3.2.1 GPS Spoofing

Similar to network protocols, civilian GPS receivers are vulnerable to attacks such as blocking, jamming, and spoofing[13]. If time were critical to security, as it is in APEC, the security of using GPS to synchronize time must be examined.

In [14], several cost effective countermeasures are presented to complicate these attacks. However, not only does this paper clearly state that the countermeasures

proposed will not stop spoofing attacks, but a Conrell University led team has recently built a briefcase size GPS spoofing device that the authors are “fairly certain they could spoof all of these, and that's the value of their work.”[15].

Only military GPS signals are encrypted or authenticated. Plans to upgrade do not include adding encryption or authentication to the civilian GPS signal [14].

More importantly, encrypting GPS signals is once again a nearly equivalent problem to key distribution. To securely provide accurate time it must use encryption outside of the APEC security model. Except in the case of military GPS, this encryption is not even publicly available. Obviously, this defeats much of the purpose if the ultimate goal is to use a time-space protocol to avoid the computational costs of public key authentication. Alternatives must be explored.

3.2.2 Cost Constraints

One thing that is not analyzed in [1] is the cost associated with GPS. It is difficult to ascertain the exact cost comparison of GPS versus computation of keys between nodes, but a GPS based time synchronization scheme is almost certainly more expensive in terms of dollars. As the last section demonstrates, authentication is probably necessary when using GPS to synchronize time, so the power consumption may be somewhat comparable.

3.2.3 Inaccessibility

Wireless sensor networks are used in a variety of applications and in a variety of places. There exist military applications, sensors in extreme or isolated areas, etc. There are many places where a GPS signal is unavailable at various times for various reasons.

Additionally, GPS is currently operated by the US government. This may be inappropriate for many countries in many circumstances (military applications for non-US countries, for example).

3.3 Conclusions About Time Synchronization

For the distribution of time to be effective, network time protocols require authentication. Unlike a public encryption scheme, they do not require a key exchange because information is pre-distributed. However, neither do 802.11 protocols necessarily in the case of pre-shared keys (PSK) [16]. In both cases for this to be effective, information must be pre-shared.

The methods analyzed above for securely distributing time provide no advantage to traditional methods. Though using time synchronization may theoretically relieve a protocol from the costs of cryptographic authentication, to synchronize time effectively cryptographic authentication must be used on the time protocols themselves, negating all cost benefits.

4. SPACE

Technology can improve in surprising ways. For example, crystal frequency may improve to the point where accurate time is possible at low cost. Currently, low cost quartz crystals resonate at roughly 32,768 Hz and are accurate to 5-10 seconds per month [17]. If similar low-cost technologies are able to offer better performance, it may be possible to preconfigure these sensor devices with

accurate time that does not need to be updated.

It is possible to assume a best-case scenario where time is somehow synchronized and accurate in a cost-effective and secure method. This may be feasible through methods that are currently unknown.

Despite this leap of faith, there are still many issues with an APEC-like protocol that must be addressed.

In [1], the assumption is made that the radio frequencies are concealed. It states, "Property 2: It is desirable to conceal the selection of the radio frequency channel pair assigned to only one node within the network." This is fundamentally a very difficult problem to solve.

It is reasonable to assume that APEC uses an 802.11 protocol. Though the exact protocol is not mentioned, there would be some serious constraints if the frequencies were arbitrary. First, if the frequencies used were outside of the 2.4GHz or 5GHz spectrum, there would be the issue of licensing these (since 2.4GHz and 5GHz are some of the few frequencies available for public use and additional frequencies would have to be approved by the FDA for use in the United States). Secondly, there would be the issue of outfitting these supposedly low-cost sensor devices with a broad range of antennas and transmission equipment that would be capable of transmitting across these frequencies. Because of these tremendous difficulties, the below discussion assumes the range of frequencies used is within the 802.11 spectrums. However, as long as a small, finite spectrum is used then the below arguments should apply.

4.1 Encryption

One of the most serious limitations of a space based encryption scheme is the fact all frequencies can be monitored in an effectively simultaneous manner.

Because 802.11b/g has only 11 channels in the US [18], one method to do this is to simply buy 11 wireless cards and to simultaneously monitor all 11 frequencies. USB wireless cards that are capable of monitor mode are relatively inexpensive and multiple USB cards can work on a single notebook.

The disadvantages of using 11 wireless cards are cost and power consumption (though neither of these constraints make the idea unusable). Most wireless cards are able to perform frequency hopping and are capable of monitoring multiple frequencies. Initially, it was thought that a single card could effectively monitor all frequencies simultaneously using this technique, and initial tests seemed to support this theory. However, in subsequent testing the initial results were not repeatable.

Alice, monitoring 11 channels on a frequency hopping device



Laptop, sending packets at a random time on frequency A



Wireless Router

Figure 1: An experiment to ascertain how effective channel hopping is for 802.11 networks

The program for sending packets at a random time from the laptop to a server has some nuances that should be mentioned. Delta is a function determining how often the packets are randomly sent. For example, a delta 1 would send out a new packet at a random time between 0 and 1 second. A new TCP connection is established for every data packet sent. Below is the main program loop:

```
for i in range(0,3000, 2):
    s = socket.socket(AF_INET, SOCK_STREAM)
    s.connect((sys.argv[1], 7700))
    #sleep for a random 0-1s * delta
    time.sleep(float(random() * delta))
    #send even numbers 0 through 65000
    s.sendall(str(i))
```

The receiver program simply accepts the connection and logs the data from the packet to a file.

```
while 1:
    clientsock, clientaddr = s.accept()
    #in reality it should be much shorter - only an integer
    thisconn = clientsock.recv(500)
    f.write(thisconn + "\n")
    clientsock.close()
    if counter % 100 == 0:
        f.flush()
```

This simulates the operation of APEC.

To capture the data, kismet was used with an ALFA wireless card containing a Realtek rtl8xxx chipset. Channel hopping was forcibly enabled with the -x flag. To count the data, tcpdump was used with the following filter:

```
ip.addr == <address> and tcp.port == 7700
and data.data
```

This filter ignores all traffic besides our own that is captured, and the number of data packets captured is printed.

Below are the results of the experiment.

Channel	Delta	Sx	Rx
1	2	1500	39
1	1	1500	35
1	1	1500	36
1	.5	1500	32
1	.25	1500	32
1	.1	1500	9
1	.05	1500	22
1	0	1500	8
5	2	1500	47
5	1	1500	44
5	.5	1500	40
5	.25	1500	25
5	.05	1500	22
5	0	1500	12
10	2	1500	31
10	1	1500	21
10	.5	1500	16
10	.25	1500	36
10	.05	1500	23
10	0	1500	10

Figure 2: Sample Experimental Results with Channel hopping enabled.

Channel	Delta	Sx	Rx
1	2	1500	1372
1	1	1500	1192
1	1	1500	1238
1	.5	1500	1450
1	.25	1500	656
1	.1	1500	451
1	.05	1500	1455
1	0	1500	1250
5	2	1500	1482
5	1	1500	1240
5	.5	1500	1489
5	.25	1500	1481
5	.05	1500	1447
5	0	1500	1138
10	2	1500	1405
10	1	1500	1400
10	.5	1500	1382
10	.25	1500	1440
10	.05	1500	1421
10	0	1500	1240

Figure 4: Control Experiment Results with Channel hopping disabled.

In an additional experiment, channel hopping was disabled and the test repeated. Channel hopping can be disabled with the -X flag in kismet, and a channel can be manually set with -I. An overview of the experiment and the results are shown in Figures 3 and 4.

Obviously, when channel hopping, the number of packets captured decreases significantly. The average packet capture when channel hopping is enabled is only about 1.8%, whereas with channel hopping this increases to about 85%.

However, although channel hopping was not shown to be a feasible attack vector in these experiments as was originally thought, it should not be ruled out as a possibility. Perhaps an RF Spectrum analyzer or different wireless cards would perform better. Regardless, the cost of purchasing several wireless cards is not prohibitive, and the point remains.

Alice, monitoring only frequency A



Laptop, sending packets at a random time on frequency A



Wireless Router

Figure 3: A control experiment to ascertain how many packets are captured with the same experiment as figure 1, but without channel hopping.

4.2 Authentication

In many situations involving wireless sensor networks, authentication is more important than encryption. While some applications, such as military networks, may require encryption, there are many others, such as environmental sensor networks, that may only require authentication.

There are several issues that should be addressed.

The first, since APEC requires pre-shared data for authentication, it should be compared to 802.11x PSK, and not a public key protocol. When compared to a public key protocol, both PSK and APEC have the disadvantage of including the “key” on every single node. Both have the advantage of being computationally efficient. Since no public implementations of APEC exist yet, the efficiencies of the two are difficult to compare.

The second and more serious issue is the security of an APEC. Although it is theoretically extremely difficult to

predict the next time slot and frequency, attacks are still likely feasible. See section 5.1.

5. ATTACK SCENARIOS

5.1 Bypassing Authentication

This is addressed in section 3 of [1], where the paper states, “attacks can be detected if two or more distinct data packets arrive during the same time-slot.” Although there are many issues of APEC that are not yet well-defined, the following is likely to succeed.

1. The attacker has multiple wireless cards, one for each frequency. This is possible because the number of frequencies is likely small – probably 11 if it uses 802.11b/g/n.

2. Because there is effectively no encryption (see section 4.1) a single node is isolated and monitored using a directional antenna. Information such as the MAC address, the next-hop node, and length of time slots are recorded.

3. Using a directional antenna and an amplifier, the isolated node is jammed.

4. For every time slot, the false message is sent to the next-hop node on every frequency. Depending on the implementation, the next-hop node will only detect the message when it is listening on the given frequency for the sending node.

It may be possible to mitigate this attack if next-hop nodes routinely listen on various channels for extra data sent, or listen for nodes that send data out of turn. It should be noted that this most likely makes it difficult to put nodes in a hibernation state which would adversely affect power consumption.

Though the exact nature of the time slots and protocols are unknown, another attack may exist that offers a significantly better likelihood of success and non-detection. The time slot is probably bigger than an average RTT and most likely includes a margin of error for network communication to be practical. An attacker can then detect that the sending node is trying to send a message on a specific channel (as he jams this channel) then immediately send packets over the channel detected. This is a very computationally fast operation, and it is likely that some falsified packets can be sent during a valid time slot. Even if it's not *always* possible to send false messages within a valid slot, if there is a practical probability that the spoofed messages sometimes make it through, then the authentication fails. This behavior may generate anomalies in the network, but unless the time slots are extremely tight then these could be extremely difficult to detect.

In APEC, authentication can most likely be bypassed unless additional constraints are imposed.

5.2 Denial of Service

It would be unfair to discount the fact that most wireless networks are particularly susceptible to denial of service attacks. However, APEC is particularly vulnerable to very precise Denial of Service attacks that should be examined. The following attack would prevent any chosen particular node from communicating while allowing the rest of the network to communicate.

1. A single node is monitored. Information such as the

MAC address, the next-hop node, and length of time slots are recorded. However, unlike the previous attack, the time slots do not need to be accurate, and this attack would most likely work without any time slot information.

2. Using multiple wireless cards for every frequency, packets are sent at an interval smaller than the likely time slots to the next-hop node. Because the protocol assumes an attack if more than one packet is received, both the valid packet and the forged packet are discarded and it is impossible to know which one was valid.

Although the protocol provides possible detection for this type of attack, prevention is not accounted for.

A similar attack that is unique to this protocol would be to disrupt the time synchronization, which would render the nodes unable to communicate. Also, if the time synchronization is not authenticated cryptographically, more sinister attack possibilities exist.

5.3 Node Capture

Similar to PSK implementations of 802.11x, there are a host of problems with pre-shared information. For example, there are no certificates to revoke if a node is compromised. Worse, unless the PRNG seed information is encrypted locally, this information can potentially be taken from a captured node and used to compromise all security.

6. CONCLUSION

APEC is a novel protocol with many interesting aspects. However, there are some serious issues that must be addressed if this protocol is to be considered as advantageous. There is no encryption, and the authentication can most likely be circumvented if additional precautions are not introduced into the protocol. Additionally, cryptographic authentication may be required outside of the protocol to accurately update time in a secure manner, defeating many of the possible cost savings. These types of concerns should be addressed before any serious attempt to implement an APEC-like protocol. As it is now, there is no obvious advantage over existing protocols.

7. ACKNOWLEDGMENTS

Thanks to Robert Hiromoto and Hope Forsmann for developing this interesting protocol to analyze, and to the OLPC project for all their work and documentation on 802.11s. Thanks to Brigitte Hodson for editing and proofreading this paper.

8. REFERENCES

- [1] R. Hiromoto, J. Forsmann, “An Authentication Protocol for Wireless Ad Hoc Networks with Embedded Certificates,” submitted for publication.
- [2] Linux 2.6.26 Changes. Available: <http://kernel.org>
- [3] RFC 4017.
- [4] J. Camp, E. Knightly, “The IEEE 802.11s Extended Service Set Mesh Networking Standard,” in *IEEE Communications Magazine*, 46(8), pp. 120-126, Aug 2008.

- [5] IEEE, "Draft amendment: ESS mesh networking," IEEE P802.11s Draft 1.00, November 2006.
- [6] RFC 1305.
- [7] D.L. Mills, "Network Time Protocol (Version 3) Specification, Implementation, and Analysis," RFC-1305, March 1992. Available: <http://www.eecis.udel.edu/~mills/database/reports/stime1/stime.pdf>
- [8] IEEE 1588-2002.
- [9] J. Kannisto, Contact Information, T. Vanhatupa, Contact Information, M. Hännikäinen, and T.D. Hämmäläinen, "Precision Time Protocol Prototype on Wireless LAN," Springer Berlin, 2004.
- [10] J. Eidson, "Measurement, Control and Communication Using IEEE 1588," Springer, April 2006.
- [11] J. Elson, L. Girod, and D. Estrin, "Fine-Grained Network Time Synchronization using Reference Broadcasts," UCLA CS Department in *First Workshop on Hot Topics in Networks (HotNets-I)*, Oct. 2002.
- [12] S. Ganeriwal, R. Kumar, M. Srivastava, "Timing-sync protocol for sensor networks," in *ACM Conference On Embedded Networked Sensor Systems*, 2003.
- [13] L. Brutt, "NS/EP Implication of GPS Timing," Office of the Manager, National Communications System, Technical Notes, Technology and Standards Division, Volume 6, Number 2, Aug. 1999.
- [14] J. Warner, R. Johnston, "GPS Spoofing Countermeasures," Los Alamos National Laboratory research paper LAUR-03-6163, Dec. 2003.
- [15] A. Ju, "Researchers raise uncomfortable questions by showing how GPS navigation devices can be duped," *Cornell Chronicle Online*, Sept. 2008, Available: <http://www.news.cornell.edu/stories/Sept08/GPSSpoofing.aj.html>
- [16] IEEE 802.11x-2004
- [17] D.B. Sullivan, "Time and frequency measurement at NIST: The first 100 years," NIST, 2001.
- [18] IEEE 802.11-2007