New Wireless Security Scheme: Preventing WLAN Network Disassociation Attack Using Management Packet Digital Signature

Tarik Guelzim
Department Of Computer Science
Graduate School
Monmouth University, 2008

IEEE 802.11 wireless local area networks provide the ability to build a cost efficient network infrastructure that is flexible and mobile. As we saw in the previous chapters, this technology tried to provide mechanisms by which to secure the network, however, it failed to do so. The technology evolved from Wired Equivalent Privacy (WEP) to Wireless Protected Access (WPA), which introduced more enhancements to the both encryption and authentication. However, both technologies did not succeed to provide the desired level of security. We are going to explore a

new security scheme which plugs in the disassociation hole in the protocol. This vulnerability allows an attacker to shutdown the entire network even for those users that are legal within the network.

1. Introduction

After the widespread deployment of wireless local area networks, many research results and simulated as well as real implemented attacks showed the security flaws that this technology has [1] . Although there is a clear advantage in using WLANs, security is a number one priority for companies who wish to deploy it on a wider scale [2, 3]. We explain the major security issues in wireless local area networks. We will demonstrate our own solution to one these problems as well as a performance evaluation compared to current standard.

The main reason 802.11 network were embraced by industry is their ability the scale in comparison to existing wired solutions such as Ethernet [4]. They also boosted the concept of mobility as any user can roam within the covered area and still get connected. In the same token, for those users that roam between large campuses and building, they have been provided with Mobile IP, this latter permits a wireless node to get associated to other adjacent access points as the user travels from one coverage area to another, thus supporting continuity and non interruption of their connection. As of the writing of this thesis, deployment of wireless technology is cheaper than it is for wired networks. Nevertheless, companies are not making a complete full switch of their infrastructure to it yet because the security issues we are going to discuss.

2. Issues with Wireless Security protocols

2.1. Wired Equivalent Privacy (WEP)

WEP is a protocol developed to provide a secure transmission model in wireless local area networks [5, 6]. The algorithm uses the RC4 cipher, which works as follows:

1- A shared key of 40 bits is given to wireless nodes ahead of time.

2- The data, both sent and received, is encrypted and decrypted using the previously distributed key.

   RC4 Takes data and apply XOR operation to it along with the key such as

   Where K_i the key, D is the data and C is the encrypted text.

3- RC4 then appends the ciphered text to the initialization vector IV used to encrypt the key

   <IV,C> = M

   Here IV is the initialization vector, C is the ciphered text and M is the Message transmitted across the network.

4- The decrypting mechanism takes the algorithm and reverses it.

Initial implementations of the RC4 IV is 24 bits, this implies that the algorithm provides 2^{24} = 16,277,216 unique keys [4]. These unique keys were thought to provide full immunity against attacks. However, the famous FMS attack, proved otherwise. The group of researchers demonstrated that given a moderate sized network, WEP is ultimately forced to reuse previous keys [4, 5, 6, 7]. This key collision enables a hacker to associate keys with packets and conduct a frequency analysis of repeated patterns, keys can be recovered and the network in then compromised. Showed that given the size of the packet 1500 bytes, an attack can be carried out on today’s networks in less than 5 hours given that key reuse start within that time frame in average. Further analysis, in [7, 8], showed that WEP encompass at least 9000 encryption keys that are vulnerable to frequency analysis. Current cracking tools such as aircrack© are programmed ahead of time to benefit from these keys. Another attack defined in [1] requires no effort to inject packets in the network. This scheme works as follows:

1- RC4 cipher encrypts the packet payload using a CRC-32 hash function.

2- CRC-32 algorithm generates a polynomial which represents the payload message.

3- The XOR operation RC4 utilizes enforces the fact that every change in the data load has a one to one map to a certain bit in the CRC-32 hash.

4- An Attacker can then capture a packet, without need of the encryption key, injects new payload, and recomputed the CRC32 hash and resend the packet. In case of UDP applications, this is certainly a very bad situation because the network does not have to spend time waiting for the next sequenced packet to arrive while the attacker conducts this attack.

This flaw in the initial implementation of WEP led to the following proposed scheme.

2.2. Wired Equivalent Privacy (WEP)

In order to fix this issue, TKIP was introduced as a software upgrade to WEP. TKIP still uses RC4 algorithm, however it initiated the following enhancements:

1- The Initialization Vector was increased from 24 bits to 48 bits.

2- Packets are keyed individually, using the IV, the packet sequence number and the user’s Medium Access Control (MAC) address in order to send a one time, unique key.

It is important to mention the fact that RC4 algorithm in itself is not weak, but rather the application it was picked to be used in has other requirements that RC4 was not originally designed for [10]. An alternative method to WEP is the Counter mode scheme; this latter uses the Advanced Encryption Scheme (AES). The way it works is as follows:

1- Before the message is sent to the network, the CRC field is encrypted using a one way function.

2- Upon reception, the integrity of the packet is checked by encoding the payload using the same integrity key as the sender. The receiver accepts the packet if its calculation matches that of the packet, otherwise, the packet is rejected.

Hence, this new technique eliminates the CRC attack since an attack would be unable to create a one to one map between the payload bits and the CRC field bits. Although this improvement introduced noteworthy additions, the counter mode has not essentially plug in the wholes of the WEP because the previously described FMS attack can still be conducted [11].

One of the main drawbacks of WEP as well as TKIP is that both schemes did not provide a method of authenticating users before entering the network. They both implemented encryption schemes to "attempt" to secure the data transmitted. The following section gives an analysis of the authentication mechanism that was introduced in the 802.11i standard.

2.3. Wireless Access Protocol (WPA 1/2)

As described in [11], TKIP was a short term solution that is aimed at providing security with a minor firmware upgrade to the routers in question. In the mean time, finding a solution for the long run began to take place. WPA or wireless access protocol version 2 had some great advantages in terms of encryption over the previous methods. The Advanced Encryption Scheme was chosen to tackle the deficiencies of RC4 [12]. Every user is now provided with a public key to be able to access the network. This public key scheme enabled a more robust authentication model since a wireless node must first authenticate to the network. Once this step is successful, the node can then associate to use the resources available. The one apparent drawback of this scheme is the extra computational overhead that is required in order to encrypt and decrypt the packets [12]. From an access point of view, this implies that the current hardware cannot sustain this kind of operation and take care of the routing mechanism as well at the same time. For this scheme to function properly, a new central processing unit (CPU) must be added to handle the encryption/decryption of packet while the other processor handles packet routing. In addition to this, 802.11i has not addressed the security flaws that the protocol inherently suffers from such as Denial of Service (DoS) Attack and Deauth attacks. The disassociation attack occurs as follows:

1- A malicious user tries to obtain access keys to the network.

2- After a successful attempt, the hacker will inject packets into the network that are flagged as management frames originating from the AP.

3- The injected management frames will be fabricated in such a way they include the MAC address of the AP as well as message that orders the entire network including legitimate and illegitimate users to disassociate themselves from the AP.

4- Upon receiving this packet, which is broadcasted, each node leaves the network.

Another DoS attack that was a variant to the above one occurs when a malicious user, after successfully associating to the network, sends 2 or more packet that fail the integrity check test [12, 13]. The 802.11 protocol states that in such a case, the access point must send disassociation packets to the entire network. We can clearly see how both of these attacks can prevent the network from functioning properly as well as the inconvenience it present to legal users as they are required to re-associate every time to the get access. There is an attempt to solve this problem by modifying the de-authentication mechanism. A suggested solution was to wait a minute after the packet is sent out to ensure that there is no further communication in the network [13]. If the server, however, still receives a packet, from a wireless node, it then knows that the disassociation packet was false and notifies the network admin. This method is clearly not suitable for the wireless network because it does not solve the delay introduced by this scheme.

In the following section, we will propose a new scheme that is aimed at solving this problem. We will also present the performance analysis of the presented method.

3. Proposed Scheme

In this work, we are proposing a scheme that attempts to eliminate the disassociation packet attack in wireless local area networks. The apparent flaw we noticed in the 802.11 stack is in the way management frames are handled. Figure 1 show how a management frame is constructed:

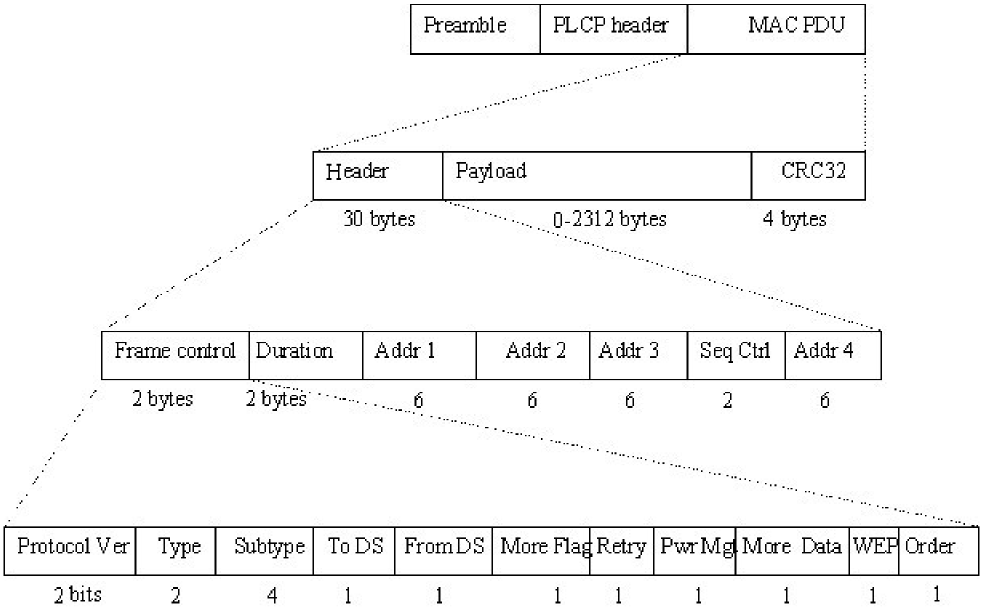

Figure 1: 802.11 MAC packet.

Management frames are two bytes and include information about protocols used as well as control information that need to be transferred from the router to the entire network. These management packets are broadcasted to the network with no encryption or if WEP is enabled, it is transmitted using the Group key. This latter is shared among the entire network users and if an intruder gained access to the network already, it is possible that he can forge these management packets and cause the disassociation of all connected users.

In our work, we saw the benefit from using digital signature as a method to ensure that each management frame is digitally signed by the access point. With this method wireless nodes can trust all management packets that are incoming from the AP. For this technique to work, we need to slightly modify the initialization of the access point.

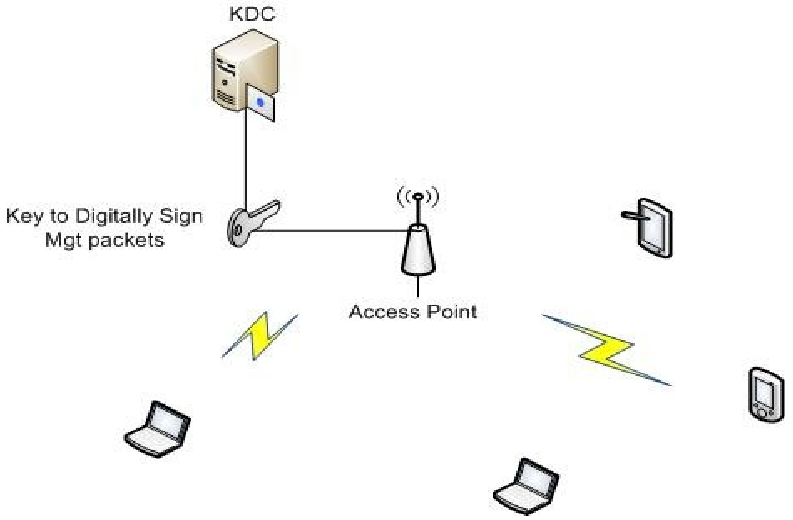

Figure 2: Obtaining MGT digital signature key from KDC

Figure 2 shows the additional functionality the AP must do before obtaining allowing wireless users to connect to it. In phase 1, the AP must authenticate itself to a CA and obtain the DS key from the local Key Distribution Center KDC. Once the AP receives the key, it uses it for management packet digital signature only. It is worth to mention that the wireless users must also connect to the KDC to verify the key of the AP. This process is usually done once. In phase 2, every management packet that leaves the AP must be signed with that key. Wireless users must check the validity of the key before accepting the received management packet as well as executing its command.

In order to check the validity of this method, we implemented a network simulation that extracts the feature of the run network in terms of encryption time, latency and network throughput in order to see how this method affects this overall performance of the network as well as justifying the network delay it might incur.

3.1. **Simulation of a 802.11 wireless network**

Modeling a realistic 802.11 wireless network is a very important aspect of this work. For that reason we created a simulation script in NS2 and a digital signature wrapper in Java. This allowed adding the digital signature capabilities of the simulation. The following are the simulation assumptions that we had:

1- We used a Poisson distribution for arrival time of packets with a Mean λ = 600 packets per second. We have chosen a large number to simulate traffic in a real world scenario.
   In terms of bytes per second or Bps, if we calculate 600 packet * 2 bytes = 1.2KB/s, this implies that we are only generating 1.2 KB of management packets (MGTP) per second.

2- Our simulation assumes that all wireless nodes are indoor only. This assumption is important since it enables us to apply the free propagation model defined by the Fris formula where P is the packet energy, k is a constant usually ¾, r is the distance from the source and n is an exponent to match the environment in which the simulation is running. In free space, n=2.

3.2. **Digital Encryption Algorithms**

The key component of our new scheme is to compare digital signature algorithms in terms of performance and impact on latency and throughput as well as router processing load. We compared the following algorithms:

1- SHA-DSA: Digital Signature algorithm with SHA family hash function.

2- MD5-RSA: RSA algorithm with MD5 hash function.

3- SHA-1-RSA: RSA with SHA-1 hash function.

We have chosen these algorithms because they are amongst the strongest in the industry.

3.3. **Methodology**

In order to obtain data to evaluate the performance of the new scheme, we ran the network simulation described in section 3.1 for 30 minutes. Every management packet sent was digitally signed and recorded in order to allow further analysis and characteristic extraction from the data obtained. Once the trace file is obtained we user a Perl library we created to determine the latency, throughput, encryption time and central tendency metrics such as the mean of the distribution.

We ran 4 simulation runs:

- The first is a regular 802.11 network without applying our scheme to it.
- The second run has DSA algorithm enabled
- The third simulation run has MD5-RSA algorithm enabled.
- The fourth and last simulation run has SHA-1 RSA algorithm enabled.

All of the above described simulations were processed under the same network infrastructure as well as the same hardware.

4. Performance Evaluation

In this section we are going to show the performance of our scheme using the 4 described algorithms and we are going to compare their performance against each other in order to see which algorithm can lead to a good tradeoff between convenience and digital signature overhead.

4.1. Performance of an 802.11 network with no use of digital signature

In this simulation run, we ran a normal WLAN network with no digital signature enabled. The following are the results we obtained.

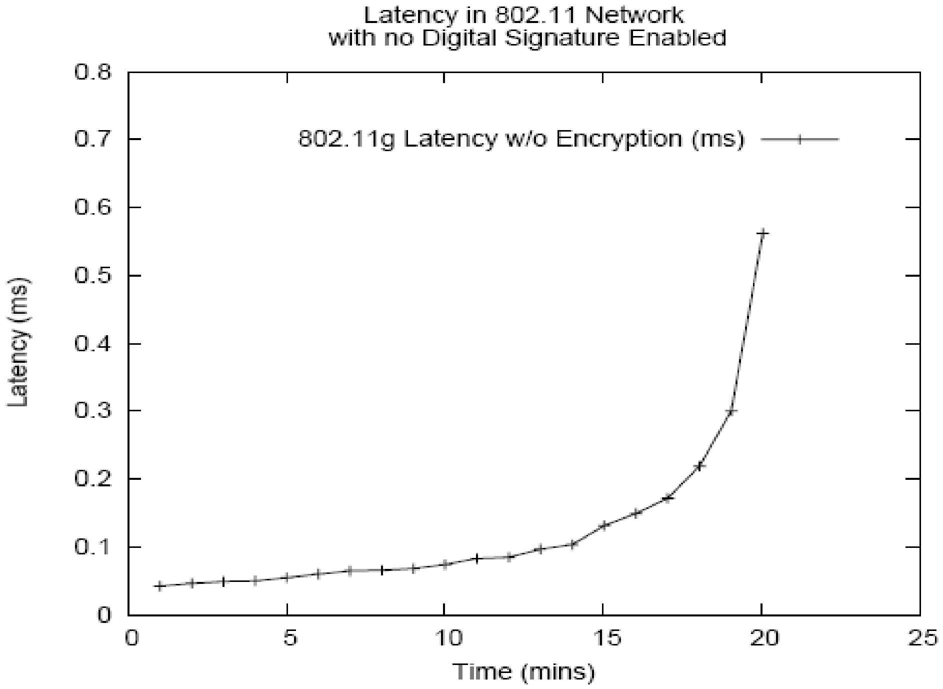

Figure 3: Latency in 802.11 network with digital signature disabled

Figure 3 shows that latency is 1/10000th of a second which is very similar to what would occur in a real time scenario. We calculated the average latency and we obtained 0.12 ms. This figure will help us later see the difference in the latency when enabling digital signature algorithms.

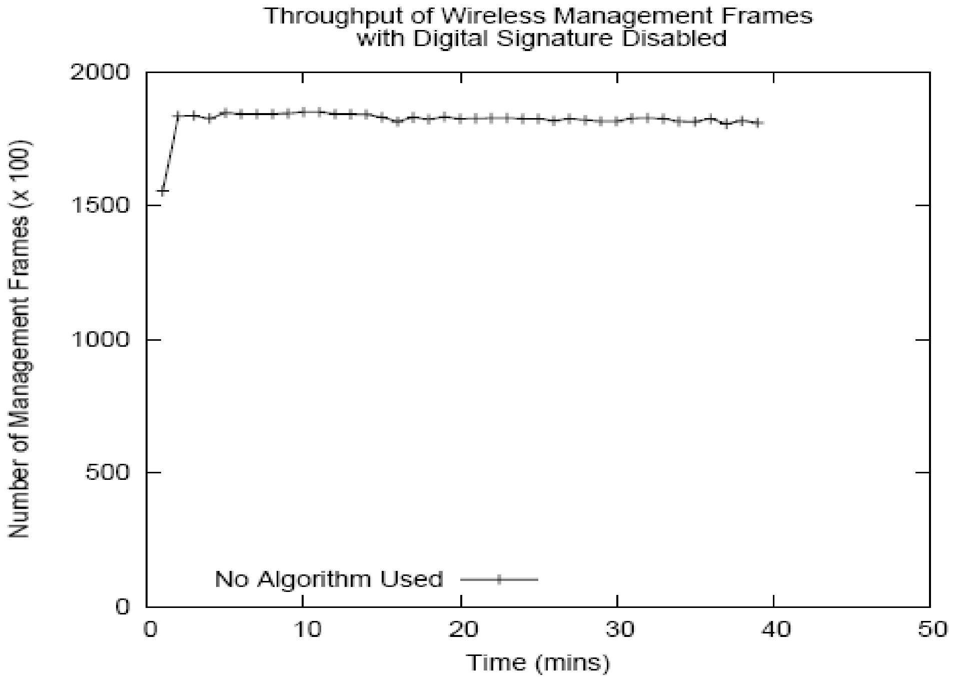

Figure 4: Throughput of MGT frames in 802.11 network with digital signature disabled

In order to understand the effect of latency on any network, it is necessary to examine the throughput of that network. Figure 4 shows an increase in throughput between 2 and 4 minutes of running the simulation, this corresponds to the phase when wireless nodes joined the network. After that phase, the throughput reached a plateau at 180000 MGT while the access point is managing the entire network. The mean throughput of the entire simulation run is 1820.44 packets, which translates to 600 packets/s which sounds a reasonable number for such a network.

4.2. Performance of an 802.11 network with DSA enabled

In this experiment, we enabled digital signature of management packets using the DSA algorithm, the following are the results we obtained.

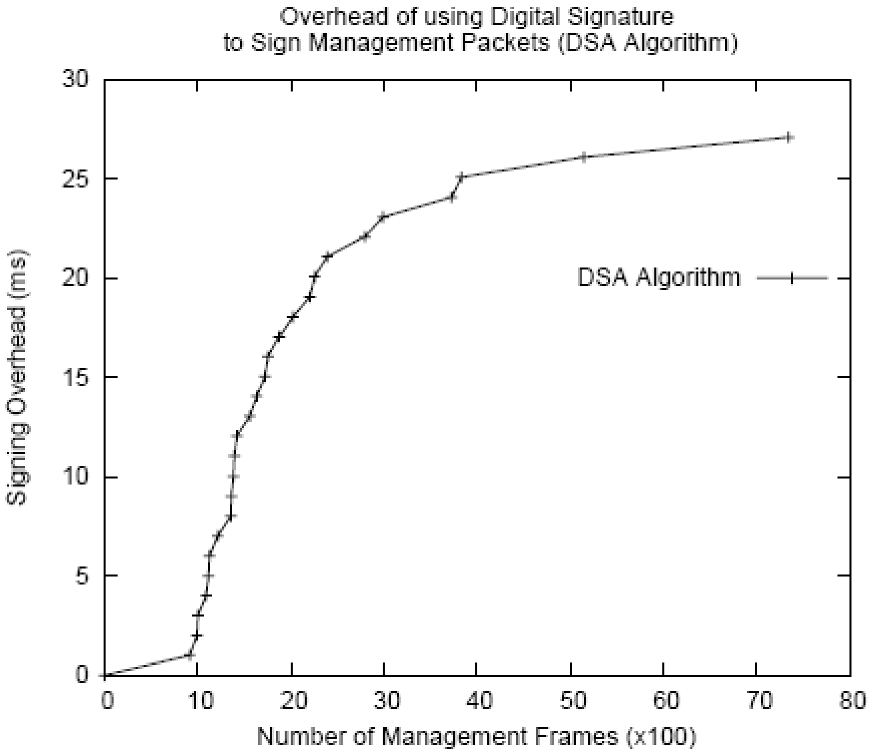

Figure 5: overhead of using DSA algorithm to sign MGT Packets

In figure 5, we notice that the overhead increases as the number of management frames sent across the network grows. The average overhead incurred by digitally signing the packets is 21.36 ms. Although this number might look negligible; we will see how it will affect the overall performance of the network in subsequent sections.

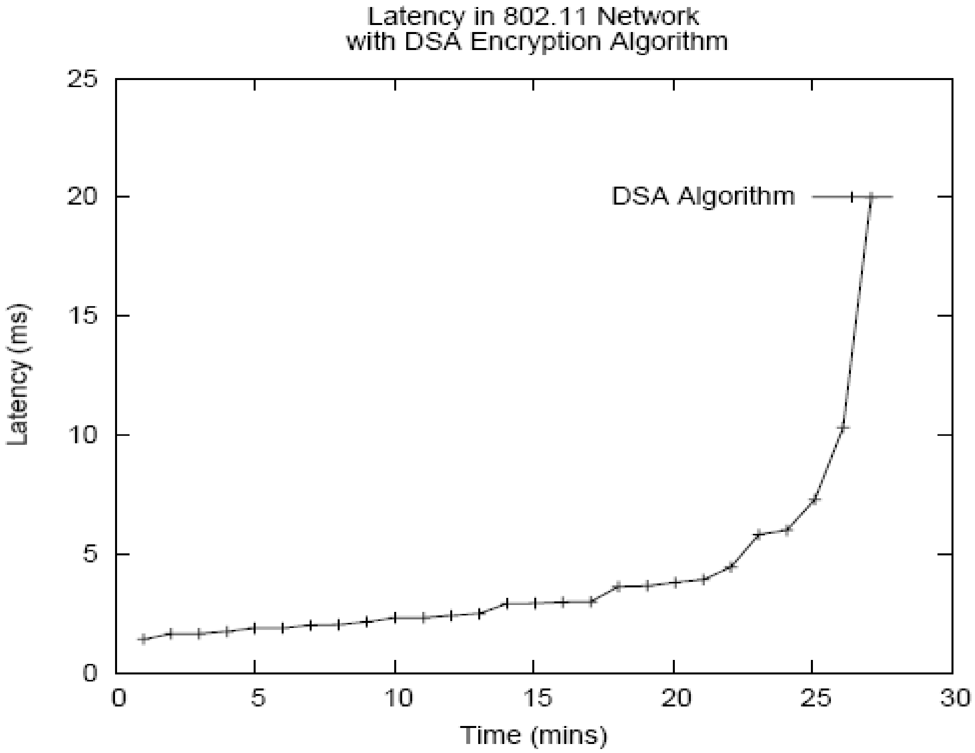

Figure 6: Latency in 802.11 with DSA algorithm enabled

Figure 6 shows the latency of the network when we enabled the DSA algorithm. The latency of the network grew, as we expected, as the simulation progressed. This implies that the AP hardware started to process more packets and encryption is slowing it down. The average latency with DSA enabled is 3.93 ms.

It is worth to mention that the access point is also processing normal packet as well. This might imply that signing MGT might incur a buffer overflow which leads to more dropped packets.

Enabling DSA increased the network latency by 3175%. This is obviously a very big difference in terms of network performance which we will also analyze by computing the throughput of the network under these conditions.

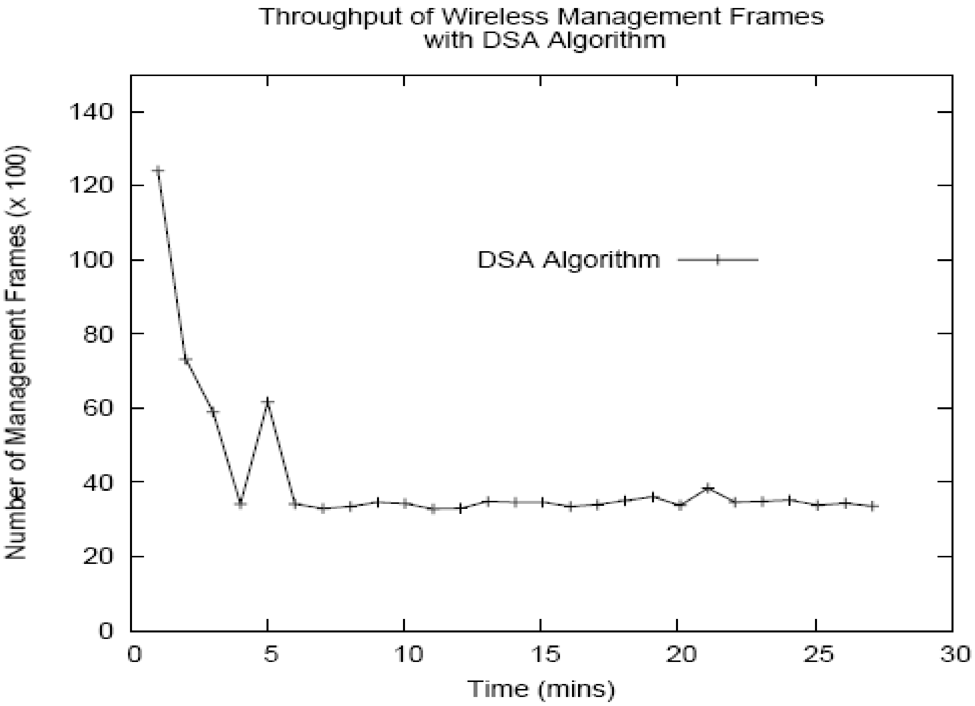

Figure 7: Throughput of 802.11 network with DSA algorithm enabled

Figure 7 shows how the network throughput decreases due to the latency incurred by the digital signature algorithm. The throughput started high, but as more nodes joined the network, the access point spent more time signing packet. As a result, the number of packets sent across the network medium decreased. The average throughput is 136.82 packets per second. This scheme affected the throughput of the network by 77%. Since this is unreasonable in most of the setups, we wanted to see the effect of other digital encryption algorithms on the network.

4.3. **Performance of an 802.11 network with RSA-MD5 enabled**

As mentioned above, DSA is not the right choice for digital packet signature since it introduces a very high performance limitation on the network. In this section, we used the RSA algorithm and the MD5 hash function to sign packets.

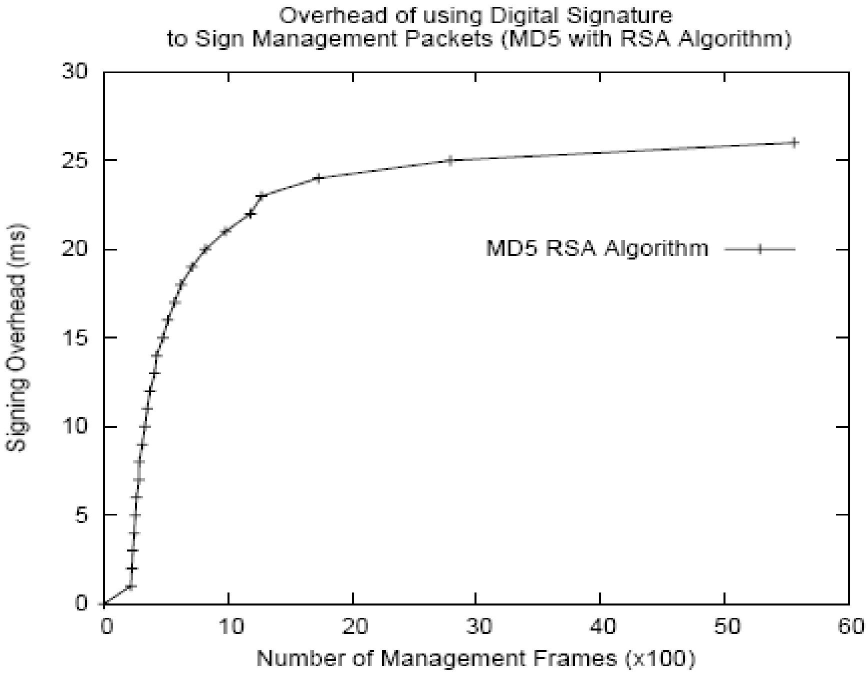

Figure 8: Overhead incurred by MD5-RSA to sign management packets

Figure 8 demonstrate the impact of the MD5-RSA algorithm on the network. As expected, the overhead increased with the increase of the management packets that have to be handled by access point. On average we each packet required an additional 7.94 ms to sign each packet before sending it. This is a significant decrease from 21.36 ms that was introduced by DSA. This is a 169% decrease in the process time the access point has to dedicate to sign those packets.

As in the previous experiments, we study the impact of MD5-RSA on the latency of the entire network.

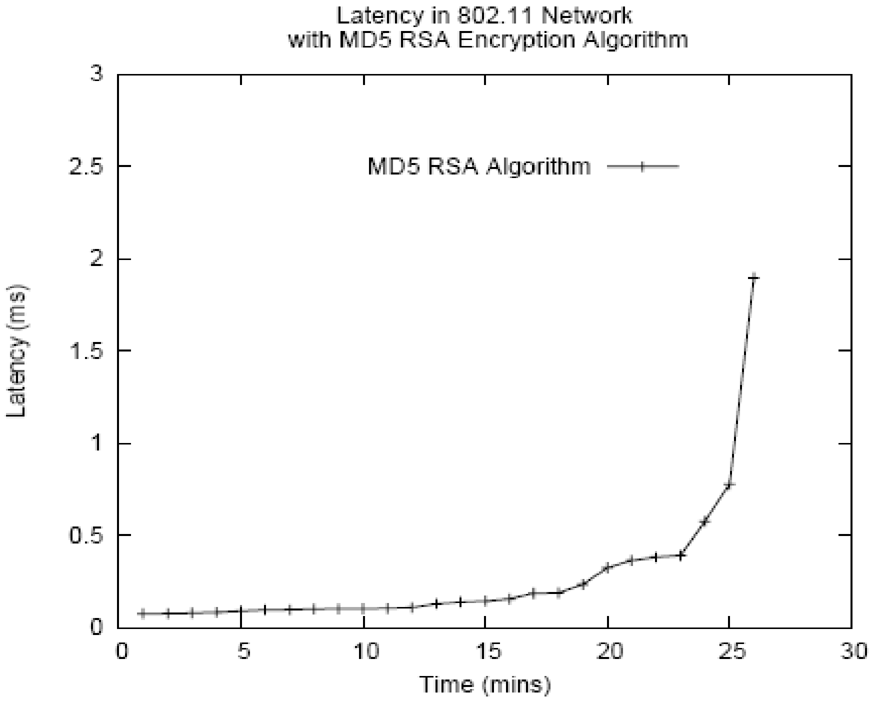

Figure 9: Latency of 802.11 network using MD5 RSA algorithm

Figure 9 shows the latency of the network when we enabled the MD5-RSA algorithm. The latency of the network grew, as we expected, as the simulation progressed. This implies that the AP hardware started to process more packets and encryption is slowing it down. The average latency with MD5-RSA enabled is 0.27 ms. This a significant decrease of 1355% compared to the latency introduced by DSA which was 3.95 ms.

In comparison to the simulation run with disabled digital signature we only incurred 125% increase in latency versus 3192% that DSA introduced.

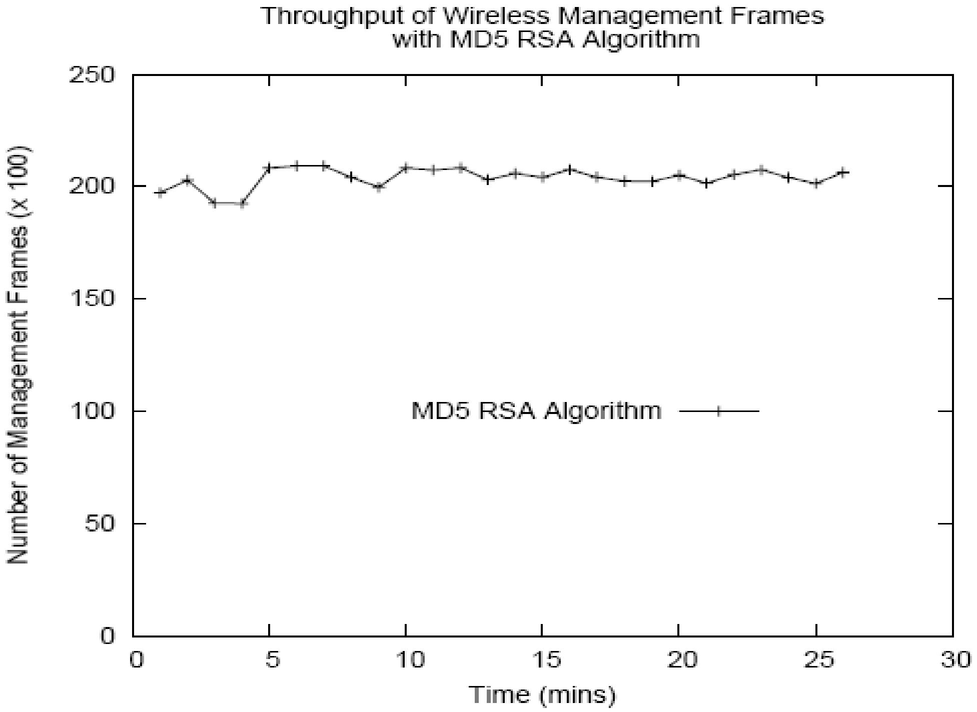

Figure 10: Throughput of 802.11 network using MD5 RSA algorithm

Figure 10 shows the throughput of the network with MD5-RSA enabled. Unlike DSA, the throughput did not drop drastically due to the use of this algorithm because the computation overhead of MD5 is very small. This can be further explained by looking at the latency as well since we only went from 0.12 ms without digital signature to 0.27 ms. The average number of packets sent was 231 with an 40% increase over DSA and 60% decrease over no signature versus 77% decrease when DSA was enabled.

As we can see, this is a very good enhancement in terms of throughput, without any decrease in security. Up to this point, MD5 RSA is the choice as far as digital encryption algorithm. However we need to compare this performance using the SHA-1 RSA algorithm.

4.4. Performance of an 802.11 network with SHA-1 RSA enabled

In this section we show the performance of using SHA-1 RSA as an algorithm to digitally sign management packets. We first start by examining the overhead introduced by such a scheme.

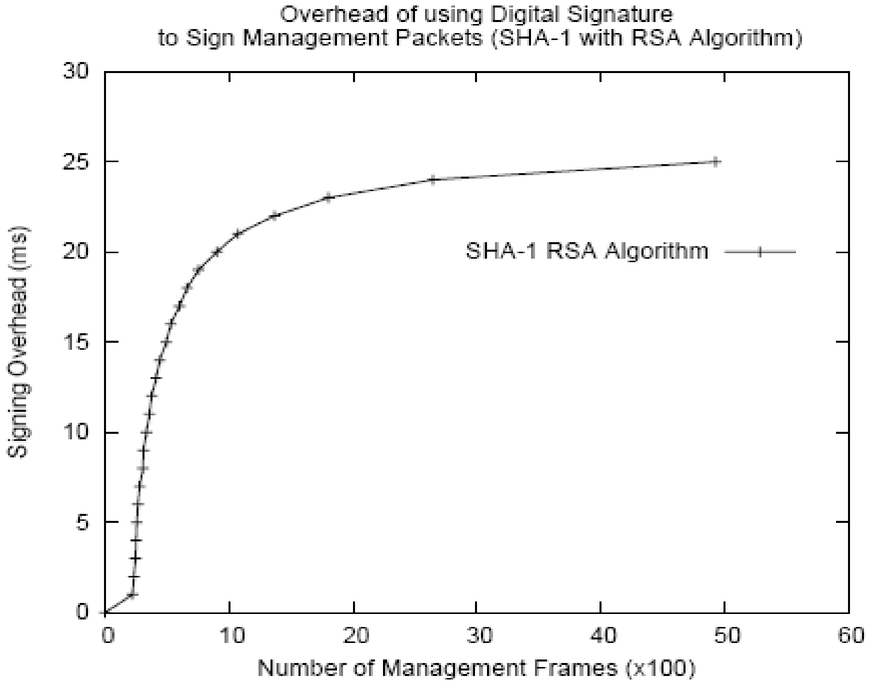

Figure 11: Overhead of 802.11 network using SHA-1 RSA algorithm

Figure 11 shows the overhead incurred by the SHA-1 RSA algorithm, which tend to increase as the number of management packets increased. This reflects the fact that the AP is busy doing the calculation to sign packets. On average an additional 7.70 ms was added to the delay of the network due to the use of this scheme. This is a decrease of 3% compared to MD5 RSA and also a decrease of 177.4% compared to DSA.

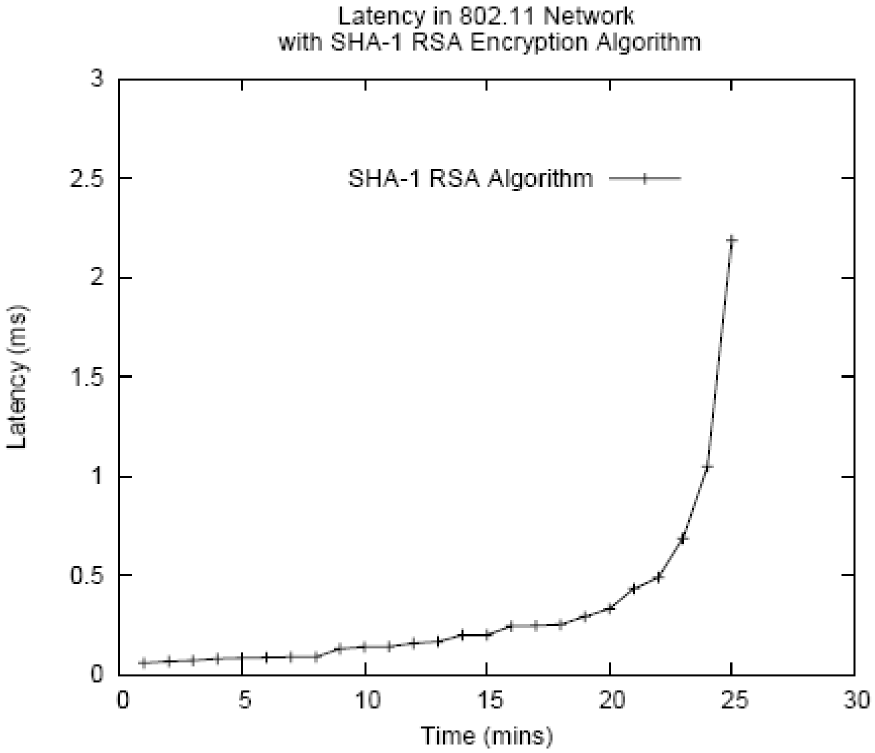

Figure 12: Latency of 802.11 network using SHA-1 RSA algorithm

Figure 12 demonstrates the latency of the network when the SHA-1 RSA algorithm was used. The latency of the network grew, as we expected, as the simulation progressed. This implies that the AP hardware started to process more packets and encryption is slowing it down. The average latency with SHA-1 RSA enabled is 0.19 ms. This a significant decrease of 29.62% compared to the latency introduced by MD5-RSA which was 0.27 ms and 1978% decrease in latency in comparison with DSA versus 1355% decrease when MD5-RSA was used.

In comparison to the simulation run with disabled digital signature we only incurred 36.8 ms or 4% increase in latency versus 3192% that DSA introduced and 125% for MD5-RSA.

In terms of throughput, we obtained the following:

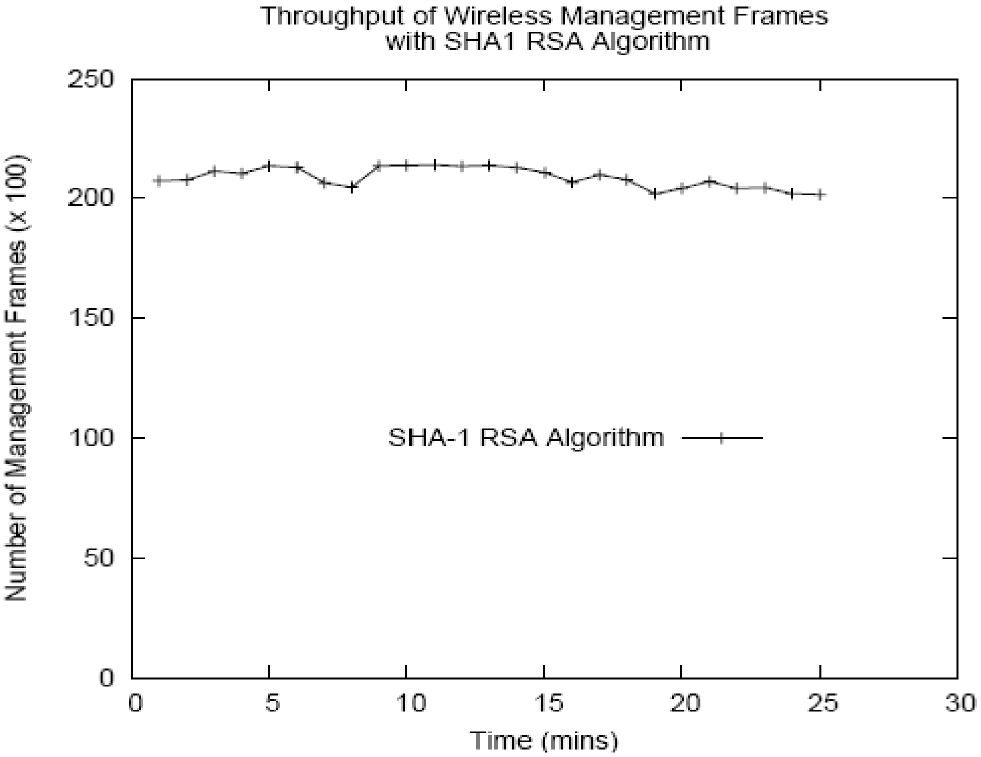

Figure 13: Throughput of 802.11 network using SHA-1 RSA algorithm

Figure 13 depicts the throughput of the network with SHA-1 RSA enabled. Unlike DSA, the throughput did not drop drastically due to the use of this algorithm because the computation overhead of SHA-1 along with MD5 is very small. This can be further explained by looking at the latency as well since we only went from 0.12 ms without digital signature to 0.19 ms. The average number of packets sent was 239 with a 46% increase over DSA and 57% decrease over no signature versus 77% decrease when DSA was enabled and 60% with MD5-RSA.

4.5. **Performance Evaluation of all the above techniques**

In the previous sections we evaluated the performance of each algorithm used on to digitally sign management packets. In this section we will compare those performances along with each other in order to determine which one is the most efficient to use without deteriorating the performance of the overall network.

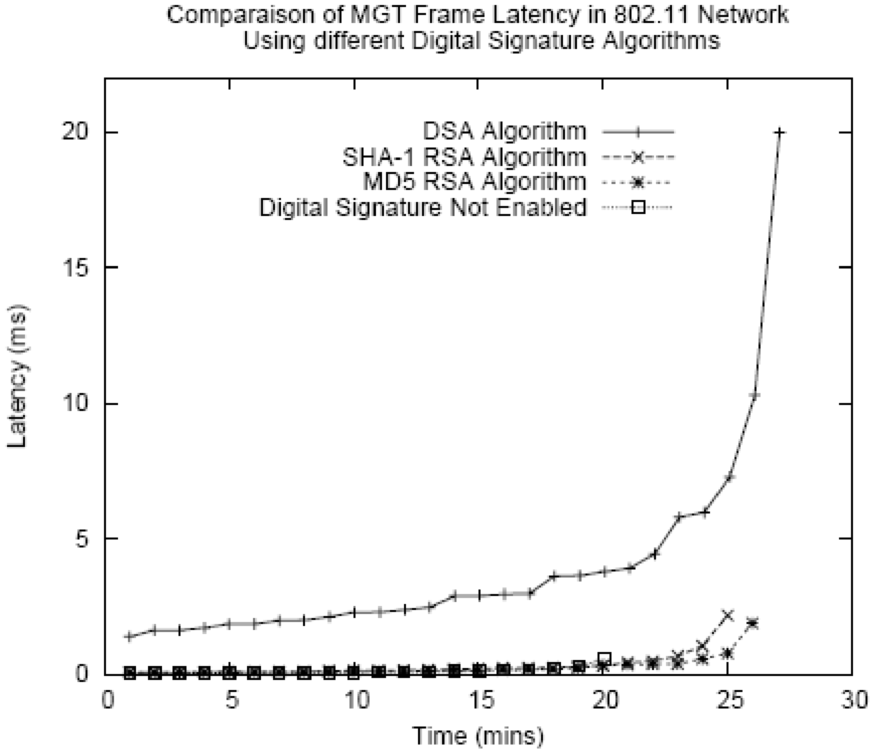

Figure 14: Latency comparison using DSA, MD5-RSA and SHA-1 RSA

Figure 14 shows the latency that is incurred by all 3 algorithms used. As we can notice with no latency we have the smallest latency possible of close to 0.12 ms. Using DSA as a signature algorithm, the top most curve, introduced an average of 3.95 ms or an increase of 3192%. MD5 RSA and SHA-1 RSA on the other hand have a 0.27ms and 0.19 ms latency respectively.

We also need to see the performance of the above algorithms in terms of throughput.

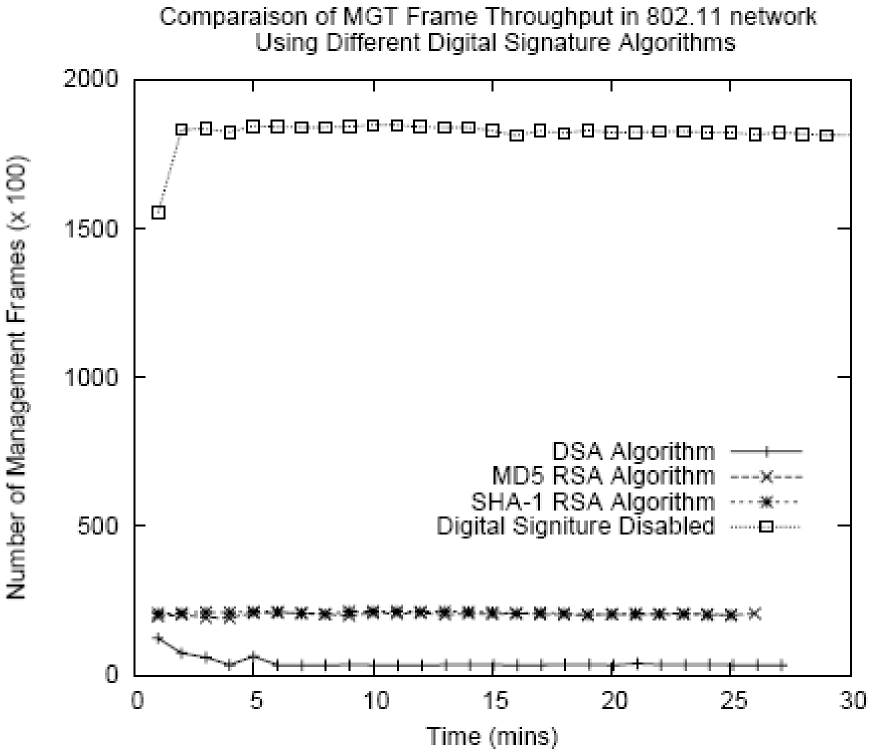

Figure 15: Network Throughput comparison using DSA, MD5-RSA and SHA-1 RSA

Figure 15 shows the throughput of the algorithms used to sign the packets. Disabling signature capability showed the network under normal conditions with an average of 600 packets per second. MD5 and SHA-1 RSA algorithms produced very similar characteristics in terms of throughput because there latency and encryption was about the same. The throughputs were 231 and 239 packets/s for MD5 and SHA1 respectively. DSA again ranked last with 136.82 packets/s, which is over 1/2 the throughput of the other 2 algorithms, and 1/5 the performance of the 802.11 network with no signature involved.

5. Conclusion

The IEEE 802.11 standard allowed wireless local area network to proliferate at a very high rate. Nevertheless, it failed to attain the security levels that it promised to deliver at design stage.

Many security flaws were rectified in the incremental versions of the standard however some security holes are still an issue. In this work, we presented a new scheme to correct the disassociation packet attack flaw, which hackers use to de-authenticate all users in a network. Our scheme relies on the digital signature method. This latter allows the access point to sign management frames with its private key. Mobile users can then verify the signature and accept the packets if the key is valid. In our work we implemented and compared three algorithms: DSA, MD5-RSA and SHA-1 RSA. After evaluation, we concluded that DSA is not a very good candidate for this scheme because it introduces very high latency as well as low throughput. The RSA schemes, however, produced close results and allowed to make this scheme very applicable without negatively affecting neither latency nor throughput.

References

[1] Se Hyun Park, Aura Ganz, Zvi Ganz, "Security Protocol for IEEE 802.11 Wireless Local Area Network", *ACM mobile Networks and Application,* Vol.12, issue3, pp. 237-246, Sept. 1998.

[2] Toshiya Okabe, Takayuki Shizuno, Tsutomu Kitamura, "Wireless LAN Access Network System for Moving Vehicles," *IEEE Symposium on Computers and Communications (ISCC'05)*, pp. 211-216, June 2005.

[3] Edith C. H. Ngai, Michael R. Lyu, "Trust- and Clustering-Based Authentication Services in Mobile Ad Hoc Networks*", International Conference on Distributed Computing Systems Workshops,* pp. 582-587, March 2004.

[4] Michell, S. and Srinivasan, "State based key hop protocol: a lightweight security protocol for wireless networks", *Proceedings of the 1st ACM international Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, pp. 112-118, Oct. 2004.

[5] He, C. and Mitchell, J. C., "Analysis of the 802.11i 4-way handshake", *ACM Workshop on Wireless Security,* pp. 43-50, Oct. 2004.

[6] Jin-Cherng Lin, Yu-Hsin Kao, Chen-Wei Yang, "Secure Enhanced Wireless Transfer Protocol,", *First International Conference on Availability, Reliability and Security (ARES'06)*, pp. 536-543, April 2006.

[7] Arbaugh, W. A., Shankar, N., and Wang, J. "Your 802.11 Network has no Clothes" *IEEE International Conference on Wireless LANs and Home Networks*, pp. 131-144, Dec. 2001.

[8] Bellardo, J., and Savage, S., "802.11 Denial-of-Service attacks: real vulnerabilities and practical solutions", In Proceedings of the USENIX Security Symposium, pp. 15-28, August 2003.

[9] Walker J., 802.11 Security Series - Part II: The Temporal Key Integrity Protocol (TKIP), Intel Corporation, 2002.

[10] J. Kohl and B. Neuman, “The Kerberos network authentication service (version 5),” RFC-1510, June 1991.
[11] L. Gong, “Increasing Availability and Security of an Authentication Service,” *IEEE Journal on Selected Areasin Communications*, vol. 11, no. 5, June 1993.
[12] J. Li and W. Jia, “Traffic Analysis in Ad Hoc Networks Based on Location-Aware clustering,” *Proceeding of the 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW ’03),* 2003.
[13] J. Li and W. Jia, “Traffic Analysis in Ad Hoc Networks Based on Location-Aware Clustering,” *Proceeding of the 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW ’03),* 2003.