Advanced Security Issues in Wireless Networks

Alexander Krenhuber[1] and Andreas Niederschick[1]

Johannes Kepler Universität Linz,
Institut für Informationsverarbeitung und Mikroprozessortechnik
alex.krenhuber@liwest.at, andi_niederschick@gmx.at

Abstract. The intention of this paper is to give an overview of security mechanisms in wireless LAN technology and possible attacks and abuses affecting them.

1 Introduction

Wireless LANs are very useful to enrich organizational or private LANs with mobility and easy access. As the costs for wireless devices are continuously decreasing, WLANs are employed more and more often. But they come at a higher risk than wired LANs due to the usage of a shared medium, which every device, capable of receiving wireless signals, can decode the sent information. Therefore various security mechanisms have to be applied to encounter threats emerging from using WLANs. This paper shows a few of these threats and how to overcome them starting with commonly known encryption techniques and ending up in more specialized scenarios.

2 WEP

To ensure security, WLAN-networks need data encryption and authentication. A first approach has been provided using "Wired Equivalent Privacy", or short WEP.

2.1 Basics of the WEP-Encryption

WEP uses the RC4 stream cypher to encrypt data. Initially a root key (Rk) is chosen which is shared among participants of the network. This key remains unchanged during the WLAN exists. To increase the possible keys in a network, the root key is extended by an initialisation vector (IV). The IV is chosen for every packet transmitted. Using || as concatenation symbol, the key K to encrypt a packet in WEP can be written as IV||Rk. Out of this packet key the RC4 key scheduling algorithm (KSA) generates a keystream, which is a permutation of the numbers 0 to 255. Then the Pseudo-Random Generation Algorithm (PRGA) generates a stream of numbers using this permutation. This stream is used for encryption by XORing it with the plain text. The plain text consists of the data and an integrity check value (ICV). The resulting cyphertext is sent over the network together with the unencrypted IV.[1]

2.2 Attacks on WEP

An early attack on WEP was developed by Fluhrer, Mantin and Shamir and is therefore called FSM-Attack.[2] Their main point are that the first bytes of data is quite easy to predict because they often provide meta information like the type of header used. Through this knowledge one can calculate the first bytes of the keystream. An attacker also knows the IV of the key because it is transmitted unencrypted. Now the attacker can simulate some steps of the KSA. As the key is part of the algorithm, just a few conditions have to hold to guess the next byte of the key. The probability to guess correct is about $(\frac{1}{e})^3 \approx 5\%$.[2]

This probability results in about 4.000.000 to 6.000.000 needed packets to recover the key at a 50% chance. Improved variants are the KoreK attack, which defines more conditions and requires about 700.00 packets, and the PTW attack, which does not need more than 35.000 to 40.000 packets.

3 Wi-Fi Protected Access (WPA)

Since WEP is not secure at all and chosen the right attacking method it is only a matter of minutes to hack WEP someone had to act. IEEE 802.11i (also known as WPA2) was far from completeness and so the Wi-Fi Alliance, a global non profit association with more than 300 companies, introduced in April 2003 a subset of 802.11i which they called Wi-Fi Protected Access (WPA). WPA does address all weaknesses of WEP and implements security features that can be used on nearly all WEP capable hardware (backwards compatibility). It also adds user authentication which was largely missing in WEP.[3]

3.1 Understanding the different Key Types

In WPA and WPA2 (IEEE 802.11i) many different keys are in use. To managed the different keys a key hierarchy is build. The two main key types are pairwise keys, which are used for communication between an access point and the client and the group keys which are used for broadcast transmissions.
On top off all WPA/WPA2 keys their exists the so called Pairwise Master Key. This key is either be transferred during authentification when using WPA-Enterprise or preconfigured on all clients if using WPA-PSK.
In WPA-Enterprise the PMK is a random 256Bit number. In WPA-PSK a method called PBKDF2 (Password-Based Key Derivation Function) is used to compute the PMK from the preshared key (PSK). PBKDF2 is a key derivation function that is part of Public-Key Cryptography Standards (PKCS) #5 v2.0. The PBKDF2 is feed with the PSK, the SSID, the length of the SSID, how often the hash function should be executed and the length of the key it should produce.

$$\text{PMK} = \text{PBKDF2}(\text{PSK}, \text{SSID}, \text{SSID\_Length}, 4096, 256)\ ^{1}$$

[1] http://www.rsa.com/rsalabs/node.asp?id=2127

The PMK is only used for authentification and to generate the Pairwise Transient Key (PTK). The PTK is computed every time a new connection is established. This is done during the 4-way handshake.[4]
The first 128Bits of the PMK are used as the Key Confirmation Key (KEK). This key is used to bind the PMK to the AP and client and to proof to the AP the knowledge of the PMK. Bits 129-255 of the PTK are used as the Key Encryption Key (KEK) that is used to safely exchange group keys. The last 256 bits are used as the Transient Key (TK). This key is used to encrypt the communication between AP and Client. It depends on the encryption method used (TKIP or AES), on how this key will be used to generate the actual communication encryption key. TKIP uses the first 128Bit of the Transient Key as the actual session key and the second 128Bit as the key to encrypt the MIC (MIC-Key).[4]

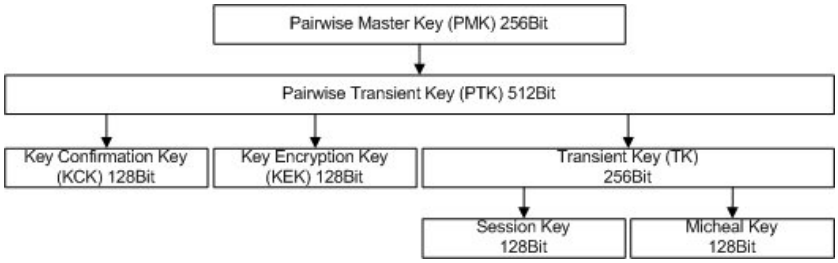

Fig. 1. Key hierarchy in WPA and 802.11i [4]

Group keys are used to communicate with more than one client. The AP generates the group transient key (GTK) and encrypts it with the Key Encryption Key (KEK) of the specific client. The client can decrypt this key and group communication is established. This group key is renewed whenever a client disconnects from the AP or after a specific time interval.[4]

3.2 Authentication mechanisms

Before communication between AP and client can happen a mechanism is needed to establish e secure connection between the participants. After successful 802.11 Authentication and Association (weak authentification) the link between the client and AP is up and ready for further authentication.
In 802.11i there are two different methods to authenticate to the access point. Using WPA/WPA2 preshared key, the AP and the client know the same psk and so during the 4-Way handshake they are able to establish the final secure connection.[5]
Using Enterprise modus (802.1X) it is a little bit more complex and another party, the authentification (RADIUS) server is involved. After successful 802.11 Authentication and Association 802.1X ports remain blocked and no packets

can be exchange. Client and authentication server now execute a mutual authentication protocol which is some secure kind from the EAP family (EAP-TLS, EAP-TTLS, EAP-PEAP) to authenticate each other. The communication between them is relayed through the AP. After successful authentication a secret is generated. This is called the Master Session Key (MSK). This secret is securely transferred (encrypted through the radius protocol) from the authentication server to the AP. Now client and AP have the same Master Session Key and can compute the PMK. Now the 4-Way handshake can be executed and the key hierarchy can be build.[6]

4 Hacking WPA

4.1 Weaknesses

So now that the basics of WPA keys and encryption method was shown, what are the real security threats and why are they possible? The biggest security hole is the MIC value of message 2-4 of the 4-Way handshake. This MIC value was created using the whole EAP message and putting it through the HMAC_MD5 hashing algorithm which is than secured by the MIC Key. As the whole 4-way handshake is not encrypted all packets can be captured and all necessary information extracted.

1. The PMK is generated using the PBKDF2(PSK, SSID, SSID_Length, 4096, 256) function since all necassary information is known
2. The PTK (Pairwise Transient Key) is generated
   Information needed:
   - PMK and length of PMK
   - MAC of client and AP, can be extracted from packet 3
   - AP nonce, can be extracted from packet 3
   - Client nonce, can be extracted from packet 2

   PTK = sha1_prf(PMK, PMK_LEN, Pairwise key expansion, data, sizeof(data))
   Where data is composed of:
   LowerMac, HigherMac, LowerNonce, HigherNonce
3. The MIC is generated:
   TKIP: MIC = hmac_md5(key, 16, data);
   AES: MIC = hmac_sha1(key, 16, data);

The computed MIC can now be compared to the captured MIC value from packet 4. If they are the same the PSK must be the same.[2]

4.2 Brute Force

As shown, having a 4-way handshake, it is possible to attack the preshared key. But since generating the MIC value is an exhaustive computing process, doing a

[2] http://www.ciscopress.com/articles/article.asp?p=370636&seqNum=6

brute force attack is really time consuming. Still there are programs[3] out there which are doing exactly the above mentioned method to compare the capture MIC value against generated ones which are computed from a list of likely used passwords (dictionary attack). But even with a list of just 1 million words and with a common processing power of around 100 words per seconds it will take up to 2 hours to crack the key. To improve this time consuming process several attempts that have been developed will described in the following sections.

Rainbow Tables Rainbow tables have been around for quite a while and where first described in 1980 by Martin Hellman. The whole goal of rainbow tables is to reduce the time off crypto analytic processes by using precalculated data which is stored in memory. Once a rainbow table is computed a lookup of a hash can be done in a very performant way and it is often easier, cheaper and more efficient to do the actual calculation once as calculate each time you have to recover a key. But for WPA generating a rainbow table is not that easy since the preshared key is salted with the ssid and its length thus making the hash different on different ssids. However with a quick look at http://www.wigle.net/gps/gps//Stat a preference to some ssids can be shown and so generating rainbow tables for the well used ssids have been made. The performance increase using rainbow tables instead of pure processing power is around factor 100. Now a normal computer can test up to 20.000 keys per second which means our 1 million dictionary could be tested in less than 1 minute[4].

Distributed Cracking In October 2008 the Russian security company Elcomsoft[5] shocked the world with its new password cracking suite called Distributed Password Recovery. This new version of their software is capable of using the computer's graphics processor (GPU) to accelerate cracking of Wi-FI passwords. They claim that this new method will improve password recovery time by 100. In response to this statement, a security expert from GSS[6] claimed that now WPA/WPA2 becomes obsolete and other techniques must be used to secure Wireless Networks. Using previous methods (cowpatty, aircrack) 60-200 keys/s could be tested, using GPUs this can now be raised to 10000 Keys/s per video card.
A similar program, called Pyrit[7], was posted some days before Elcomsofts announcement in the Nvidia Cuda forum[8]. The performance is very similar, which means a modern Nvidia Geforce 280GTX can compute around 11500 keys/s. Pyrit has only about 2000 lines of code and is available as open source.

[3] http://www.willhackforsushi.com/Cowpatty.html
[4] http://www.renderlab.net/projects/WPA-tables/
[5] http://www.elcomsoft.com
[6] http://www.gss-security.com
[7] http://code.google.com/p/pyrit/
[8] http://forums.nvidia.com/index.php?showtopic=76778

At moment this type of software only runs on graphic cards made by Nvidia. Using the Nvidia CUDA SDK, programmers can take use of the many-core parallel processing power of modern GPUs.
So with all this new and great processing power, what does this mean to the security of WPA? To explain this, a closer look at the possible key space of WPA should be taken. As already mentioned, a WPA Key can be between 8 and 63 characters long and may consist of all printable ASCII characters. The following table shows the amount of possible keys when using a certain amount of characters from a certain character set.

keys	Numeric (10)	Alpha (26)	Alpha Numeric single (36)	Alpha case sense (52)	Alpha Numeric case (62)	Printable ASCII (96)
8	1E08	2.1E11	2.8E12	5.4E13	2.2E14	7.2E15
9	1E09	5.4E12	1E14	2.8E15	1.4E16	6.9E17
10	1E10	1.4E14	3.7E15	1.4E17	8.4E17	6.6E19
11	1E11	3.7E15	1.3E17	7.5E18	5.2E19	6.4E21

With a semi-modern Nvidia Geforce 8800GTS, which is available for about 100€, running one of the above mentioned programms, around 7500 Keys/s can be tested. This is not very much and as shown in the next table using 8 characters containing at least alpha numeric symbols seems very safe because the average time to crack the password is around 6 years (2176 days).

days	Numeric (10)	Alpha (26)	Alpha Numeric single (36)	Alpha case sense (52)	Alpha Numeric case (62)	Printable ASCII (96)
8	0.077	161	2176	41249	168472	5566277
9	0.77	4189	78364	2144989	-	-

But in time off large available bot-nets, super/ cloud computing and available software like Elcomsofts one, that is able to use up to 10.000 Computer with up to 3 GPUs, one can safely assume that a combined processing power of 10.000 GPUs should not be too hard to get. What this means is shown in the next table.

days	Numeric (10)	Alpha (26)	Alpha Numeric single (36)	Alpha case sense (52)	Alpha Numeric case (62)	Printable ASCII (96)
8	0	0.02	0.2	4	17	556
9	0	0.42	7.8	214	1044	53436
10	0	11	282	11153	64760	5129881
11	0.007	283	10156	580005	4015166	-

As the table show, numeric keys are not really safe and even at a key length of 16 symbols the average time to crack the key is around 2 years (771 days). Since technology is improving at a high rate a safe minimum key length should be at least 12-14 characters from either Alpha Numeric case sensitive key space or better using all printable ASCII characters.

4.3 Key exporting

By now using a secure preshared key hacking into a WPA protected network can be really time consuming and so there must be other ways to get the key.
One of these methods is key exporting. In a PSK environment every client must have knowledge of the used key. So each client is a possible target to extract the stored key. Using linux as operating system it is likely that the PSK is stored in some wpa supplicant configuration file. Using windows as operating system it depends on the version used where the key is stored.

- Vista C:\ProgramData\Microsoft\Wlansvc\Profiles\Interfaces
- XP HKLM\SOFTWARE\Microsoft\WZCSVC\Parameters\Interfaces
- 2000 HKLM\SYSTEM\CurrentControlSet\Control\Class\Adapter_ID_Number

So hacking a pc connected to that wireless network and extracting the stored key can be far less time consuming than attacking the network itself.

5 WPA Enterprise

So as seen so far using preshared keys is not secure at all. There are so many different ways to get compromised and it is also not very feasible in a corporate environment because off security reasons the preshared key should be changed periodically.
As stated in the WPA Authentication mechanisms section, using WPA Enterprise AAA is done through some kind of Extensible Authentication Protocol (EAP). During this process the Master Session Key (MSK) is transferred to the client which is comparable to the stored preshared key in WPA-PSK. But since the is different each time a client authenticates all client stations are using different encryption keys. This means that in enterprise mode, knowing one encryption key does not lead to the ability to decrypt the whole traffic that is transferred in this network.

5.1 Roaming in WPA Enterprise

Roaming in multi Access Point scenarios is happening all the time, even if the client does not change its position but due changing environmental conditions. Every time a client change its AP the whole 802.11X authentication mechanism has to be executed which can take up to a few seconds. This is far too long to provide interrupt free network connection. To improve the speed of re-authentication, 802.11i does provide some fast-roaming features.

- Key Caching: An already established encryption key (PMK) is transferred to the new AP and so only the 4-way handshake has to be done between the client and the new AP.

– Pre-Authentication: A client can authenticate simultaneously with multiple new APs while it is still associated to its currently used. Clients that use Pre-Authentication can use their established connection to the network to send 802.1X to other APs.

These two new features allows the roaming to be done in less than 1/10th of second which is fast enough time sensitive applications.[9]

6 Performance issues using Secure Methods

At last we want to investigate if these security methods have some influence to performance of data transmission. To explore performance issues a small test setup has been built with one Access Point (Linksys WRT54GS) running OpenWRT, a client pc (Lenovo T61) with different wireless lan cards and another pc connected via 100MBit/s Ethernet providing the second endpoint. The distance between the AP and client was around 5meters including one brick wall. To measure the actual performance iperf[10] was executed for 5 minutes which measured the average transfer rate in KBit/s.

	Intel 4965AGN	Nec WL54AG	Netgear WG511
Open	27206	21004	21213
WEP 128Bit	25093	20617	20178
WPA TKIP	22975	18209	19532
WPA AES	26886	20442	21293
WPA2 AES	26623	na	na

As expected WEP and WPA TKIP influences the performance pretty bad, costing in the worst case 4MBit/s (15%) when using WPA TKIP with the Intel wlan card. But switching to WPA AES does change the picture. As the numbers show only a slight, non relevant decrease in performance was measured. This is due the fact that AES encryption is performed in hardware (not downwards compatible to older wlan cards) and TKIP encryption is performed in software. As this table shows using WPA/WPA2 AES with reasonably new wlan cards does not influence data transmission performance at all.

7 Rogue Access Points

The cost of wireless hardware and therefore of wireless networks as a whole are getting cheaper these days. Everybody can buy and operate his own access points (AP). These can be used not only for setting up wireless networks but also to get illegal access or sniff information. For example an AP could replace another and take over his identity. Other participants of the WLAN won't be able to detect the difference on their own.[7]

[9] http://blogs.zdnet.com/Ou/?p=67
[10] http://sourceforge.net/projects/iperf

7.1 Categories of Rogue Access Points

There are diverse categories of rogue APs. A normal AP providing useful functions to the network might not be configured correctly. These so called **open APs** could expose access to the network to listening extern devices. Within this network, end-users could be attacked or data could be exported. More dangerous are **backdoor APs**. Normally they are installed by an employee who does not configure the AP correctly, but they can also be inserted into the network on purpose by an intruder. Backdoors can be very harmful because they are attached to the wired network of an organization and can therefore avoid WLAN-security mechanisms. The most hazardous rogue APs are **fake APs**. An intruder can plant an AP in the network to reveal identification details of personnel or set up a man-in-the-middle attack. The access points can be located inside or outside the organizations building. They normally have the same security configuration and SSID like official APs to increase the difficulty of detecting them.[7]

7.2 Detecting Rogue Access Points

The most simple approach of detecting rogue APs is to send stuff through an organisation with an antenna and programs like NetStumbler [11] to detect unofficial AP. The problem is, that this detection method takes a lot of time and is therefore very cost intensive. Furthermore access points can be unplugged fastly when the detection is happening. The hardware used must be able to listen on different frequencies because devices can be using divergent standards like 802.11a or b.[8] A more cost efficient way is to put sensors over the organizational area. Sensors also provide the possibility to scan for rogue APs permanently. The sensed data can be administered from a central point. These sensors also have to listen on multiple frequencies. These two approaches can be enhanced through adding a wired listener at the layers 2 and 3. Routers and Switches can determine whether other devices with a certain MAC-address are connected to them or not.[8]

Another approach is solely wired. Here, only switches and routers at network layers 2 and 3 are queried to get MAC-addresses of attached devices. The problem is that MAC-adresses can be spoofed. This detection variant therefore can be undermined quite easily.[8] Similar is the usage of a Mobile Manager which keeps a record of all devices belonging to the network. As long as the identities of these devices are not compromised, rogue APs can be detected.[7]

The differences of temporal traffic characteristics between wired and wireless links can also be used to detect rogue APs. This approach assumes that a wired network has lower spreading of packets than a wireless network. This is true due to the worse link and channel variability of wireless networks. The shape of resulting graphs show whether wireless access points are in the network or not.[8]

[11] www.netstumbler.com

8 Conclusion

As we show WEP cannot be considered as secure. There are many different methods to crack WEP secured networks and the time it takes does get smaller and smaller. With additional, higher level securing methods like using vpn, WEP could be used but since WPA TKIP is downwards compatible there is simply no point in using WEP. Using WPA with preshared keys security depends only on the complexity of the used key. With all the new hardware and cracking methods weak keys are a real security issue. In corporate usage 802.11i does provide a secure mechanism that assures mutual authentification, integrity and confidentiality and so it should be the way to go.

References

1. Nick L. Petroni Jr., W.A.A.: The dangers of mitigating security design flaws: A wireless case study. IEEE Security and Privacy **1, no. 1** (2003) 28–36
2. Scott Fluhrer, Itsik Mantin, A.S.: Weaknesses in the key scheduling algorithm of rc4. 4th Annual Workshop on Selected Areas of Cryptography (2001)
3. Alliance, W.F.: Wi-fi protected access: Strong, standards-based, interoperable security for todays wi-fi networks (April 2003)
4. Eckert, C.: IT-Sicherheit: Konzepte - Verfahren - Protokolle. Oldenbourg (2007)
5. Changhua He, J.C.M.: Analysis of the 802.11i 4-way handshake. Third ACM International Workshop on Wireless Security **WiSe'04** (2004) 43–50
6. Changhua He, J.C.M.: Security analysis and improvements for ieee 802.11i. The 12th Annual Network and Distributed System Security Symposium **NDSS'05** (2005) 90–110
7. Zhiqi Tao, A.R.: Detecting rogue access points that endanger the maginot line of wireless authentication. (2005)
8. Beyah, R.; Kangude, S.Y.G.S.B.C.J.: Rogue access point detection using temporal traffic characteristics. Global Telecommunications Conference, GLOBECOM apos;04. IEEE **4** (2004) 2271 – 2275