TTM4137 Wireless Security Lab Assignment 2011

WLAN Security Analysis and Construction

NTNU, Department of Telematics
version 3.6, September 19, 2011

Contents

0 The Lab

0.1 Objectives

In this lab assignment you will explore many of the protocols and mechanisms proposed in IEEE 802.11 to provide security for wireless local area networks (WLANs). This will be both on the analysis side and the constructive side. You will perform security analysis and assessment of WLANs already running, and configure, secure and deploy new ones yourself. Hopefully, you will get a better understanding of the weaknesses, strengths of these security mechanisms, and the challenges of network security management.

The lab project is divided into three stages and presented in the Chapters 1, 2 and 3 in this document. In the first part you will assess the security of the Wired Equivalent Privacy (WEP) protocol. This should give you the experience of how much effort it actually takes to bypass WEP protection by eavesdropping on the radio communication. In the second part you will put your hands on the Pre-Shared Key (PSK) mode of Wi-Fi Protected Access (WPA) and try to find the password by an intelligent exhaustive search method. In the third part you will configure your own Extensible Authentication Protocol - Transport Layer Security (EAP-TLS) to be used between the stations and the network with the mechanism of Robust Security Network (RSN). This is the best that the IEEE 802.11-2007 standard [3] can offer of security.

Proper understanding of security depends on understanding the threats, that is why a significant part of the lab is devoted to breaching security mechanisms. Students should keep in mind that such attack skills can be foul play and may be illegal to apply outside the lab environment.

0.2 Organisation

0.2.1 Student Groups

This project is carried out in groups of three students. Due to limited capacity in the lab, each group will have to sign up for **one of the weeks 39 or 40** and finish their work in the lab during that week. Only eight groups are available for each week, which means that you have to hurry with the registration if either the week number or a choice of particular group fellows is important for you. Consider coordinating your choice of group with projects in other courses running in parallel.

Please register within **Monday, September 19** by signing up via the web link for *group registration* posted on It's learning [4]. Contact the course staff at [5] if there's no free places in groups left.

0.2.2 Location and Time

The lab work and supervision will take place in **Room G-122** Elektro building. You can enter the laboratory room and work there **any time from Monday 08:00 to Friday 14:00 on your assigned week**. The laboratory room is reserved for this course only.

0.2.3 Work Load

The total amount of work will be approximately 40 hours, some groups will need to use more, some will do with less. *Do not postpone the work, get started as soon as possible!* It is a good idea to start with reading the background material referred to in this document several days before your practical work starts. The group members should agree on a reasonable work schedule (see lab milestones in Section 0.3.2).

0.2.4 Supervision

The teaching assistants are Md. Abdul Based (office B-220) and Razib Hayat Khan (office B-222). They will be available to help you and check your work. They will be present in the lab during the hours

Monday	14:15–15:00
Tuesday	13:15–14:00
Wednesday	13.15–14.00
Thursday	13:15–14:00
Friday	12:15–14:00

If you are stuck with a problem, then do the following:

1. Think, and try a fresh approach. Check other sources if needed.
2. Try asking the group next to you. (Cooperation is good, copying is bad.)
3. If you are still stuck, ask one of the teaching assistants.

An email address for course-related questions is [5]. You can also open and discuss threads on the lab discussion forum at It's learning.

0.2.5 Tools

Each group will be working with two desktop PCs equipped with wireless network interface cards (NIC, D-Link DWL-G520, Hardware version: B4). A basic knowledge of the Linux environment will be a prerequisite. Ubuntu Linux version 8.04 has been set up to reduce the work load, and the programs you will be needing and the required drivers have been preinstalled. Your job will be to perform the actual auditing and configuration in order to get the mechanisms to work. When you enter the lab for the first time, choose two computers standing close to each other. The username is `ttm4137`, and the password is initially set to

```
ttm4137
```

Change the password so that other groups don't use your computers throughout the whole week.

We describe how the laboratory PCs were prepared for this lab project. The following packages were added to a blank 32-bit Ubuntu 8.04 Hardy Desktop Edition installation: `aircrack-ng`, `madwifi-tools`, `kismet`, `john`, `wireshark`, `hostapd`, `bridge-utils`, `vim`, `openssh-server`. A package `freeradius` in a compilation that supports `openssl` and EAP-TLS was installed.The following line has been edited in `/etc/kismet/kismet.conf`:

```
source=madwifi_g,wifi0,Atheros
```

The following lines were edited in `/etc/ssh/sshd_config`:

```
PermitRootLogin no
AllowUsers ttm4137
```

0.2.6 Laboratory Journal (log)

It is best practice to keep a log of the complete *work process*. This lab log will work as the basis for your laboratory report. By laboratory log we mean a text file or a document created as you go, which includes records of the objectives for the current session, necessary foundation for the work, what should be done, how it was done, and finally the result from the session. Remember to take some screenshots of relevant steps that you might want to include in the report.

One good reason for doing the log is that you have a limited time available in the lab (you will make use of the notes when composing the report outside the lab hours). It is also the proper engineering and scientific approach to your work. The laboratory log, however, will not be assessed by the course staff.

0.2.7 Your Creativity

The assignment is formulated as a set of minimum requirements, but we encourage you to let your creative engineering power take you beyond the path we have been staking! For example, try other attacks (like [10]) or modify the ones described here, experiment with your setup, try other EAP protocols, etc. Describe your results in the lab report.

0.3 Acceptance and Evaluation

0.3.1 Submission

The assignment submission consists of two parts:

1. Demonstration of achieved milestones (pass/redo).
2. Lab report (grade).

0.3.2 The Milestones

We have set a few milestones for you, where we will check your work:

Challenge 1: WEP analysis. Demonstrate a terminal window with the calculated WEP key and a working Internet connection over the wireless network.

Challenge 2: password-based PSK analysis. Demonstrate your WPA/WPA2-PSK setup, a terminal window with the calculated password and a wireless Internet connection.

Challenge 3: construction of an RSN BSS. Demonstrate your RSN-EAP setup and a working wireless Internet connection.

Ask a student assistant to acknowledge your work progress at these checkpoints. All milestones must be approved by a student assistant **before Friday 14:00** on your assigned week.

0.3.3 The Report

Your report will be assessed and count as 20% of the final grade. The report must be submitted via It's learning **within the following Friday** (one week after you are done with the practical part) for evaluation. Remember to add all group members on the report when uploading the report to It's learning.

A laboratory report is a structured document, usually written and polished after the work is completed. The report is based on notes recorded during the course of experimentation (laboratory journal) and other sources.

The recommended structure of the report (see [6] for a more thorough description):

- Title Page (or heading on the first page).
- Introduction describing the objectives of your work, the lab set-up and some theoretical background.
- Experimental Procedure that *only describes occasions when you did not follow the lab description procedure.*
- Results.
- Answers to the Questions (marked with **Q** in this lab description).
- Discussion.
- Conclusion. Were the primary goals fulfilled? What should have been done differently, and why?
- References.
- Appendices.

The evaluation criteria The course staff will apply these evaluation criteria to the submitted lab report:

- Format
  - The report should not be longer than 9 A4 pages including text, references and figures, but excluding the title page and appendices.
  - Use 11pt font size, normal line separation, one-column layout and reasonable margins.
  - The submitted file format must be pdf.
  - You should write either in English or in Norwegian, with good grammar and syntax.
  - Title page (or heading on the first page) should include names of all group members, the group number, date and title.
  - The reference list should be formatted according to the common rules for citation [7].
  - We recommend using LaTeX when writing your report.
- Content
  - Clear and logical structure.
  - Clear identification of problems and objectives.
  - Precision of facts.
  - Presentation, your own analysis and evaluation of the results.
  - Logical discussion part with reasoning that clearly shows that you have understood what you have been doing.
  - Answers to the **Q**-questions (include question numbers and text when answering).
  - Quality of references. As a starting point you could refer to the course textbooks, but other references are required as well. When you find information on the Web, it is important that you are careful about its quality. We recommend that you search for information in the university library databases [9, 8], and that you choose references to published articles and books.
  - Your level of understanding of the material.
  - Effort beyond the lab requirements. Presence of new ideas.

1 WEP Penetration and Cryptanalysis

In this first assignment your task is to acquire Internet connectivity via an access point (AP) deploying WEP. You are considered to be a legitimate user which, for educational purposes, is left without the secret key. So it all comes down to finding this key.

1.1 WEP Attacks Historical Background

WEP analysis tools implement different theoretical attacks on WEP, the simplest being a brute force attack (trying every possible key until the correct key is found). The 40-bit key size is small enough to make this a feasible attack on WEP. The brute-force approach was further simplified in some implementations where algorithms for converting human readable pass-phrases into hexadecimal WEP keys were being used, causing the key space entropy to decrease.

To mitigate the brute force attacks vendors increased the key size, but cryptanalysis of WEP showed that the algorithm's security is in fact independent of its key size because of the problem with keystream re-use. The actual key used to initiate the RC4 stream cipher is a concatenation of a 24-bit Initialisation Vector (IV) and a 40-bit secret key. The reason for using IVs is to avoid reusing of keystreams, but the number of 2^{24} different IVs is far too small when each packet sent over the 802.11 channel needs a different IV. With a traffic of 11 Mbps a collision is likely to occur in a matter of seconds.

If you know the plaintext corresponding to the ciphertext (known plaintext attack), you are able to compute the key stream (how?). Also, if the sender reuses a keystream of bits to encrypt a new message, you are able to compute $m_1 \oplus m_2$ (how?). In WEP deploying *Shared Key Authentication* clients are authenticated by the AP with a handshake where the AP sends a plaintext challenge and the client responds with an encrypted version of this challenge. An attacker can simply XOR the challenge with the response to recover the keystream used. Since WEP has no mechanisms for preventing old IV values to be used, this opens up for active attacks where messages can be injected by an attacker without access to the secret key.

In 2001, Fluhrer, Mantin, and Shamir published a ciphertext-only attack against RC4 [11]. The attack, known as the FMS attack, recovers the secret key of RC4 with a high probability if around 4 million encrypted packets are available, and is based on three main principles:

- Some "weak" IV values set up the RC4 cipher in a way such that it can leak key information in its output bytes
- Invariance weakness allows use of the output bytes to determine the most probable key bytes
- The first output bytes are always predictable as they contain the SNAP header (we can partially deploy a known plaintext attack)

This enables the determination of key bytes from observations of the keystream. A single weak IV can determine the correct key byte with a probability of 5%, by collecting a large number of weak IVs the most probable key can be found and tested.

Vendors responded to this attack by filtering out the weak IVs, but they could not catch up as the original attack was refined and further classes of weak IVs were discovered. In 2004 a hacker using the nickname KoreK further improved the practical attacks against WEP with the release of a set of statistical attacks that reduced the amount of packets needed for key-recovery to around 500,000. The history of WEP attacks up to this point is very well described in [12]. The paper also demonstrates the fragmentation attack (related to the chopchop attack described by KoreK), where a single packet can be decrypted without the knowledge of the secret key in order to allow packet injection.

In 2007 Tews, Weinmann and Pyshkin published an attack (the PTW attack) that reduced the number of packets needed to as little as 40-85,000 without the weak IV requirement [2]. The PTW attack is an improvement of an attack proposed by Andreas Klein in [1]. The Klein's and PTW attacks have been thoroughly treated on our facultative lecture.

The FMS, KoreK and PTW attacks are implemented in the `aircrack-ng` tool suite for cracking WEP that will be used in this laboratory project.

1.2 Objectives and Work Flow

This is a brief outline of the minimum requirements for your laboratory activities in part one of this assignment. Details of the tools used will be described in more detail next. Be careful to keep a log of your activities as you proceed in order to prepare material for the laboratory report, remember to take some screenshots of relevant steps that you might want to include in the report. Have a look through the questions for your investigation listed in the end of this section before you start working, to see what theoretical questions you are expected to be able to answer after performing the practical parts of the assignment.

Your *target of analysis* is a Cisco AP that is set up in the lab with 104/128-bit WEP encryption and hidden Service Set ID (SSID).

- Check the wireless interfaces on your computers.
- Run `Kismet` to gather information on the parameters of the wireless LAN under analysis. This will include the AP's BSSID and on what channel the AP is communicating.
- Switch the wireless NICs of your computers to the monitor mode.
- Use the `airodump-ng` tool to obtain MAC addresses of legitimate network clients.
- Obtain network's SSID (a synonym for ESSID).
- Use the gathered info in an attempt to determine the secret key using the `aircrack-ng` tool suite. You should try to find the key using an active PTW attack. Use one PC to send and another PC to dump the traffic, this should speed up your attack.

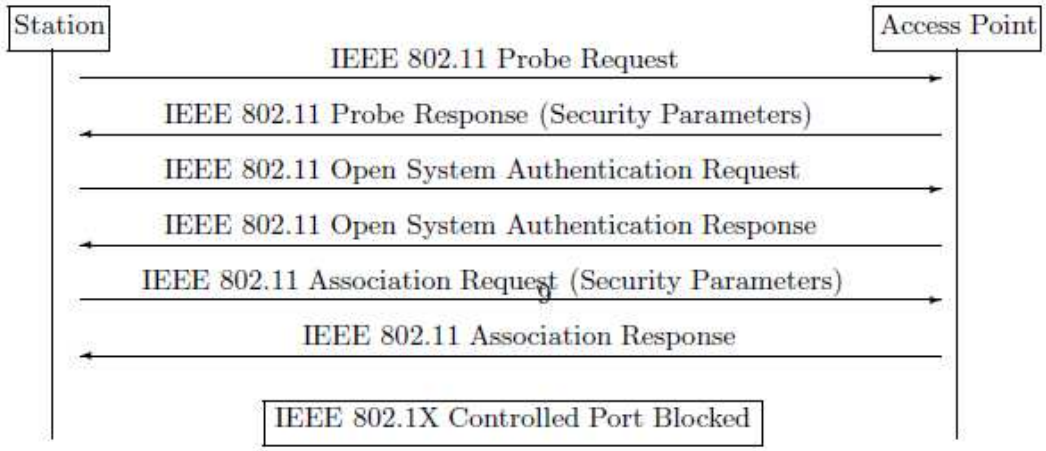

Figure 1: IEEE 802.11 Open System Authentication and Association

- Finally, demonstrate the first milestone to one of the lab assistants (see the check-list in Section 0.3.2).

1.3 Obtaining a Hidden SSID

The SSID is normally carried in every beacon frame sent by the APs. A hidden SSID is an option enabled by many network administrators hoping to hide their networks. What it means is that the beacons contain the Broadcast SSID (an SSID of length zero) instead of the configured SSID, and it is presumed that the clients participating in the WLAN know this name already. This would seem to "hide" the wireless network, as in this configuration, the only evidence of a wireless network is the MAC address of the AP.

However not only beacons, but also Probe request and response, Association request and Reassociation request frames contain the SSID [13]. Figure 1 shows the 802.11 association procedure that is carried out every time a station connects to an AP. An observation of this communication exchange allows to see a hidden SSID. In order to force a legitimate client to (re)associate, one can send a deauthentication frame using the `aireplay-ng` tool.

1.4 Tools and Programs

1.4.1 Root Privileges

Several tools require superuser access to operate as expected. It is good practice to employ root privileges only where they are needed. You can use `sudo <command>` to execute one command as root, and `sudo su` to permanently switch from your regular user to root.

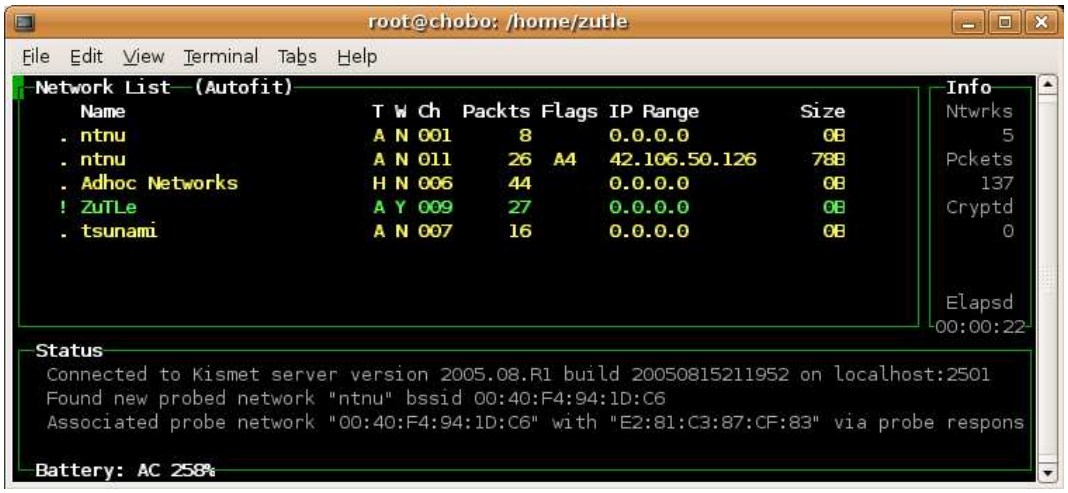

Figure 2: Kismet client interface.

1.4.2 Kismet

`Kismet` is a wireless network detector program. It "sniffs up" all nearby wireless networks and presents detailed information about the APs, such as the BSSID (MAC address) and the SSID (name). `Kismet` can record and dump traffic data to a cap file, which can then be read by the program `Wireshark`. `Kismet` determines a hidden (cloaked) SSID when frames containing the SSID have been observed. Additionally, Kismet issues alerts (WLAN-specific) to suspicious activity. These alerts can be input to an intrusion detection system, or input to network statistics collection, and much more.

Start the program by typing

```
sudo kismet_server
```

in one terminal window and

```
kismet_client
```

in another window. When the client screen pops up, hit space to close the help screen. Then press 's' followed by 'B' to sort the networks by BSSID. After this you can start using the program. Choose networks using the arrow keys and press enter to display info on them. Hit the 'q' key to go back. If you are interested in a specific network, put the cursor over it and press 'L' in order to lock the channel hopping to this network's channel.

Some of the most used commands are:

c Clients. Displays the clients associated with the selected network.

w Alerts. Brings up a window to monitor what alerts have been issued.

q Cancel. To get back to the previous meny.

x Close. Closes a pop-up window (more or less equivalent to q)

Q Exits the program.

h Brings up the help pop-up screen giving you an overview of the different options

Stop `kismet_server` by pressing `Ctrl+c` after the necessary information is obtained. `Kismet` documentation can be found at [14]. See also a quick tutorial at [15].

1.4.3 Configuring Wireless Network Interface Cards

Before we can start launching our attack, let's make sure our hardware and software are set up correctly. We are using wireless NICs based on the Atheros chipset, which requires a driver called MadWifi. This will cause the interface name to be set to `ath0` or `ath1` by default.

To see information about the network interfaces on your PC, type

```
ifconfig
iwconfig
```

The second command outputs more details about the wireless interfaces.

`Wlanconfig` is an utility special to the `madwifi` driver. It is used for manipulating wireless interfaces. With it, you can create multiple virtual interfaces in various modes from one base device `wifi0`. To create an interface issue the following command:

```
wlanconfig ath create wlandev wifi0 wlanmode <mode>
```

In this assignment we will be using three different modes:

Managed Normal mode. `<...> wlanmode sta`

Master Virtual Access Point. `<...> wlanmode ap`

Monitor Raw monitoring mode. Used to passively sniff traffic (without responding to RTS/CTS and the like) `<...> wlanmode monitor`

By default, `wlanconfig` creates an interface named `ath1` in managed mode at the startup. Your wireless NIC needs to be in monitor mode before you can start listening to traffic from surrounding networks using the `aircrack-ng` tool suite.

A convenient way of putting your NIC into monitor mode is by using the script `airmon-ng` which is included in the `aircrack-ng` tool suite. It is not recommended to have more than one `ath` interface working at the same time. So first you have to destroy all the existing interfaces:

```
airmon-ng
wlanconfig ath0 destroy
wlanconfig ath1 destroy
...
```

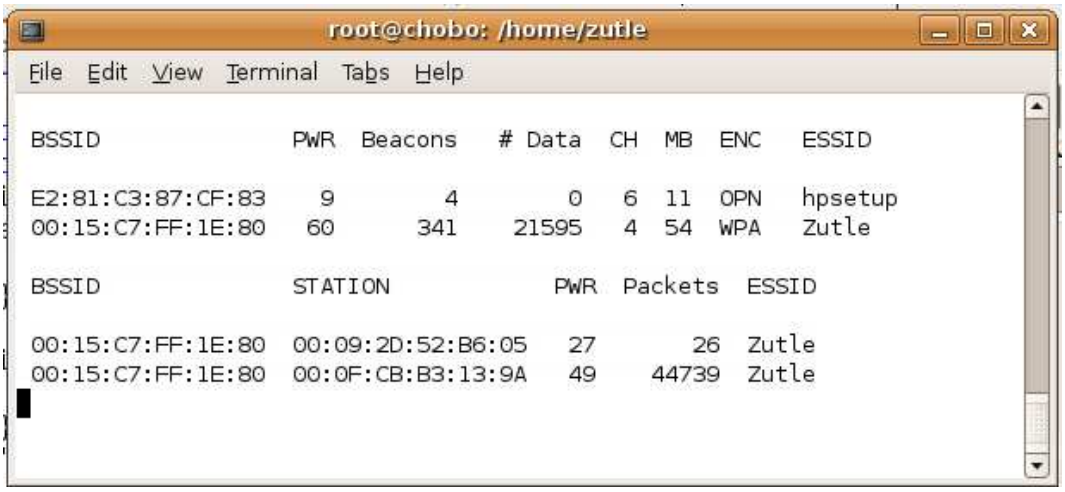

Figure 3: Airodump.

Then create a new interface in monitor mode and switch it to the necessary channel (the one that is used by the target AP):

```
airmon-ng start wifi0
iwconfig ath1 channel 7
```

1.4.4 airodump-ng, aircrack-ng and aireplay-ng

`Aircrack-ng` is a suite of programs for analysing WEP and WPA/WPA2 protected networks. See the `aircrack-ng` documentation at [16].

`Airodump-ng` is a packet capture program that sniffs traffic. You can run `airodump-ng` like this:

```
airodump-ng <options> <interface>
```

The following example runs `airodump-ng` on the interface `ath1`, dumping the traffic of the WLAN with BSSID `00:11:22:33:44:55` operating on channel 6 to the file `filename-01.cap`:

```
airodump-ng -w filename -c 6 -d 00:11:22:33:44:55 ath1
```

Note that you have to complete the instructions of Section 1.4.3 and set the correct channel with `iwconfig` before running `airodump-ng`. It is also convenient to see the list of network clients on the `airodump-ng` screen (Figure 3). A traffic dump file output by `airodump-ng` can be fed to `aircrack-ng` or to `Wireshark`.

When `airodump-ng` has gathered enough packets, you can start the PTW attack by running `aircrack-ng`. Note that the PTW attack requires a cap file as input. Keep the `airodump-ng` program running as well to get even more material for the cracking process. The more IVs, the less time it takes to recover the key (in general). `Aircrack-ng` has the following synopsis:

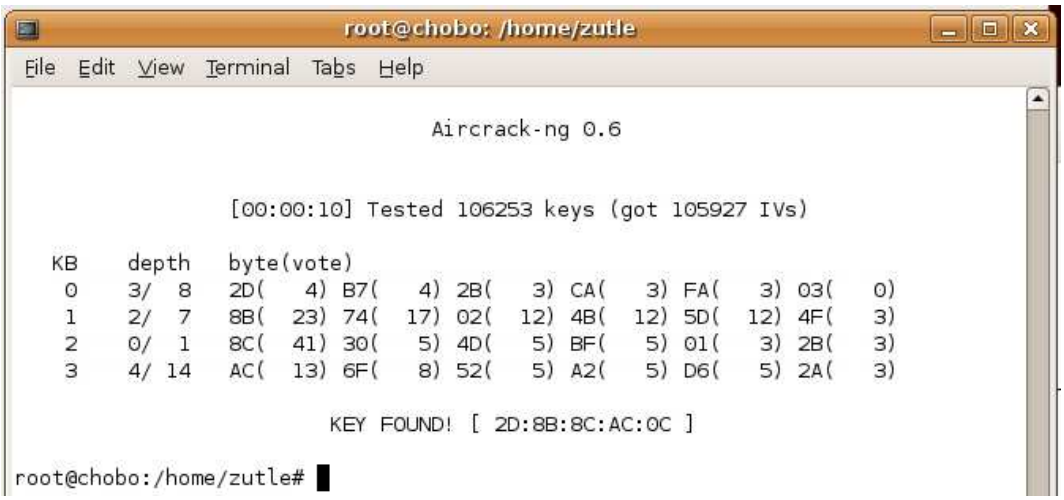

Figure 4: Aircrack-ng.

```
aircrack-ng <options> <capture file(s)>
```

The following example runs `aircrack-ng` PTW against an AP with a given BSSID using our cap dumpfile:

```
aircrack-ng -b 00:11:22:33:44:55 filename-01.cap
```

Figure 4 demonstrates the insecurity of WEP[1].

The time it takes to find the key and break WEP is proportional to the amount of traffic. But you can speed up the process by using traffic injection tools. There are several available attacks in `aireplay-ng`, and you will have to specify one when you run the program (this is the first parameter passed from the command line). A brief description of the attacks supported by `aireplay-ng` is provided below. Visit [16] for a more detailed information and command syntax.

Deauthentication Use this when there is a lot of clients associated with the target AP. It will deauthenticate the users, forcing them to reauthenticate and hence generate traffic.

Fake authentication Use this to associate with the AP. This is required for launching the attacks below[2].

Interactive packet replay Use this to inject a packet of your own choice. You can use `wireshark` to have a look at some of the captured traffic, and then try to find a request and a response.

[1]This key is found in ten seconds, after injecting arprequests for about three minutes.

[2]Alternatively you can set the source MAC address in the transmitted packets to be the same as for an already authenticated client, i.e. to masquerade. Use the `aireplay-ng -h` option for this.

ARP-request reinjection Use this attack to automatically sniff up ARP-requests and inject them into the network. You will need the MAC address of an associated client (either use fake authentication (above), or set your MAC address to resemble that of an already associated client).

KoreK chopchop This attack decrypts a single packet. If you can manage to decrypt a packet, you can actually forge your own arprequests using `packetforge-ng`, this can be useful if there are no clients associated with the AP.

Fragmentation attack Decrypts longer packets than chopchop and is much faster.

In order to perform packet injection, you will have to be associated with the AP first. To associate with the AP use the fake authentication attack like this (first an overview, then an example):

```
aireplay-ng --fakeauth <delay> -a <bssid> -e <essid> <interface>
aireplay-ng --fakeauth 1000 -a 00:11:22:33:44:55 -e networkname ath1
```

To inject traffic use the ARP-replay attack:

```
aireplay-ng --arpreplay -b <bssid> <interface>
```

1.4.5 Wireshark

Wireshark is a mighty network protocol analyser with a convenient graphical interface. When you have recovered an encryption key, you can decrypt the IEEE 802.11 packets and read the plaintext data. Open a .cap file obtained by `airodump-ng`, go to Edit → Preferences → Protocols → IEEE 802.11 and enter the decryption key. Among the packets, find an assembled HTML page, right-click on "Line-based text data" and choose "Export Selected Packet Bytes". View the saved html file in your browser. You can get more info about Wireshark at [17].

1.5 Questions

After performing this part of the assignment you should be able to answer these questions, please include the answers in your laboratory report.

Q1. Describe the parameters of the AP under analysis, such as the SSID, BSSID, channel number (and optionally the frequency), encryption mechanism, associated clients, the WEP key. How many packets did the PTW attack require?

Q2. Is it possible to run a completely passive PTW attack on 104-bit WEP? Why and under which circumstances?

Q3. Why is it possible to send an arbitrary amount of ARP-requests to the AP without knowing the WEP key?

Q4. What can you do to strengthen your attack when there are some clients, but hardly

any traffic at all?

Q5. How can you obtain packets for injection if no clients are associated with the AP?

Q6. Which weaknesses of WEP are we taking advantage of in our attack, and why?

Q7. Under what circumstances would you consider WEP to be sufficiently secure?

Q8. Would these attacks be effective against a network deploying WPA with TKIP? If the RC4 cipher used in WEP is considered to be insecure, why is it reused in WPA?

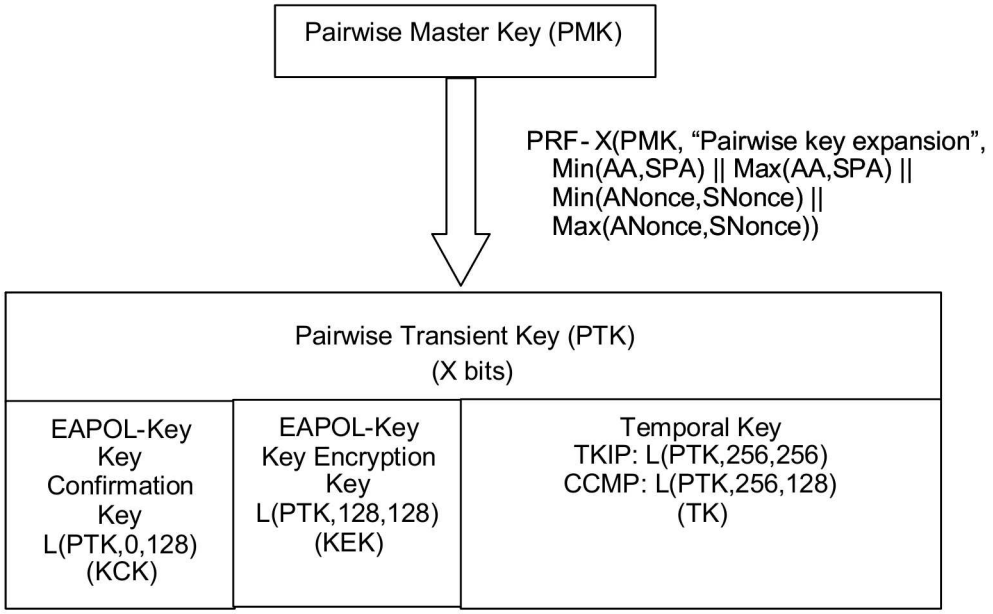

Figure 5: RSN Pairwise Kay Hierarchy

2 Password Dictionary Attack

In this part you will set up a WPA-PSK or WPA2-PSK protected WLAN and mount a password dictionary attack against it.

2.1 Background on a Pre-Shared Key Vulnerability

Wi-Fi Protected Access (WPA) is a certification program created by the Wi-Fi Alliance while waiting for the IEEE 802.11i standard to be finished. WPA devices implement a subset of the IEEE 802.11i standard, including the Temporal Key Integrity Protocol (TKIP).

Since the ratification of the 802.11i standard in 2004, the Wi-Fi Alliance refers to their approved implementation of the full 802.11i as WPA2. It differs from its ancestors, among the other things, in the encryption algorithm, which is now AES (Advanced Encryption Standard) instead of RC4. The new AES-based authenticated encryption algorithm is called CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol).

Both WPA and WPA2 use a pairwise key hierarchy defined in the IEEE 802.11i standard and pictured on Figure 5. Here PRF-X is a pseudo-random function producing X bits of output; AA — authenticator address; SPA — supplicant address; $L(\text{Str}, a, b)$ — extract b bits from Str starting from bit a; EAPOL — Extensible Authentication Protocol over LANs; EAPOL-Key — a frame type defined in 802.1X.

WPA and WPA2 security mechanism comes in two flavours:

PSK (Pre-Shared Key) used in Small office/Home office (SoHo) networks. A PSK is shared among all of the users.

Enterprise doesn't use PSK but one of several types of EAP (Extensible Authentication Protocol) for authentication. This mode of operation is the medium-sized and enterprise's choice, where the use of a single PSK is discouraged (see Section 3).

We refer to [19, §8 and §9], [22] and [23] for more details.

The station authentication and association procedure in the PSK mode is depicted on Figure 8, except for the Stage 3 (802.1X authentication) which is not being performed. A string of 256 bits is used as a common secret PSK shared between the stations and the AP. It can be input either directly as 64 hexadecimal digits, or using a more user-friendly password method. We will be interested in the latter approach as it is less secure. A password may be from 8 to 63 printable ASCII characters. It is then used to generate PSK using a known algorithm:

$$\text{PSK} \;=\; \text{PBKDF2}\,(\,\text{password}\,,\;\text{SSID}\,,\;\text{SSIDlength}\,,\;4096\,,\;256\,)\;\;,$$

where PBKDF2 (Password-Based Key Derivation Function) is a key derivation function defined in PKCS#5 v2.0 standard. 4096 is the number of hashes and 256 is the length of the output.

The Pairwise Master Key (PMK) is then computed as

$$\text{PMK} = \text{PSK}\;\;.$$

The Pairwise Transient Key (PTK) is derived from the PMK using the 4-Way Handshake, and all information used to calculate its value is transmitted in plain text (see Figure 6).

Here ANonce is an authenticator's nonce and SNonce is a supplicant's nonce.

The unpredictability of the PSK and the PMK is determined by the quality of the password. The password could be subjected to both dictionary and brute force offline attacks. An attack against PSK password was originally proposed by R. Moskowitz in [24]. An attacker captures the 4-Way Handshake messages, either by passively monitoring the wireless network traffic, or actively generating deauthentication frames to speed up the process. In fact, the first two messages are required to start guessing at PSK values. Note from Figure 5 that

$$\text{PTK} \;=\; f\,(\,\text{PMK}\,,\;\text{MAC}_{AP}\,,\;\text{MAC}_{STA}\,,\;\text{ANonce}\,,\;\text{SNonce}\,)\;\;,$$

where PMK equals PSK in our case. The attacker reads the MAC addresses and ANonce from the first message and SNonce from the second message. Now the attacker can start *guessing* the PSK value to calculate the PTK and the derived KCK (Key Confirmation Key, an integrity key protecting handshake messages). If the PSK is guessed correctly, the Message Integrity Code (MIC) of the second message could be obtained with the corresponding KCK, otherwise a new guess has to be made.

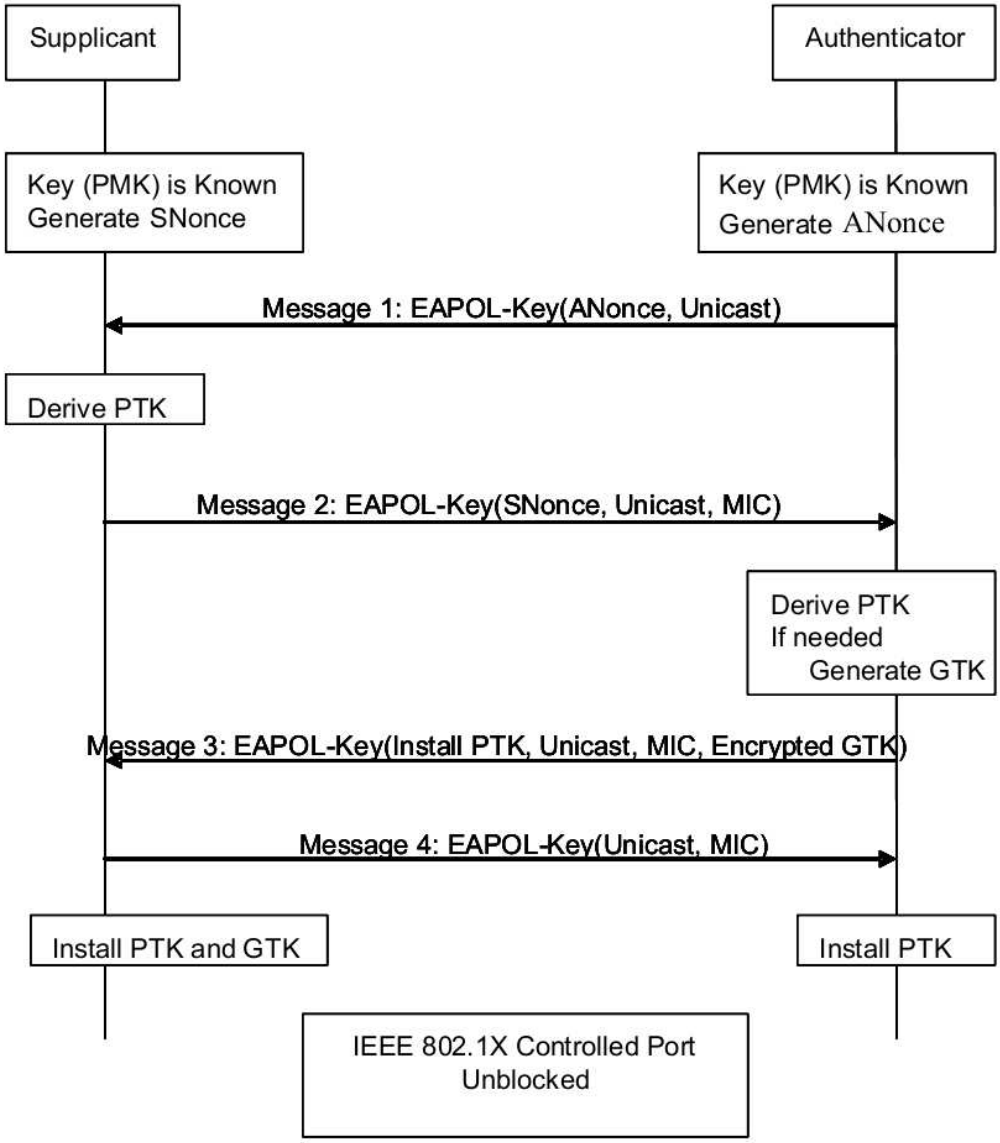

Figure 6: The 4-Way Handshake [3].

The program `cowpatty` was created to exploit this flaw, and its source code is included in the `aircrack-ng` to allow PSK dictionary and brute force attacks on WPA and WPA2. The protocol design (4096 hashes for each password attempt) means that a brute force attack is very slow (just a few hundred passwords per second with the latest single processor). Note that it is not possible to pre-compute a table of PSK from a dictionary of potential passwords because the SSID is also an argument to the PBKDF2().

2.2 Objectives and Work Flow

A sketch of your work plan is the following.

- Set up a WPA-PSK or WPA2-PSK wireless AP on one of your PCs.
- Use your personal laptop or other WLAN-enabled device to connect to your AP.
- Use `aircrack-ng` and `john` on your second PC to perform a password dictionary attack against your WLAN.
- Demonstrate your results in Milestone 2 (see the check-list in Section 0.3.2) to one of the teaching assistants.

2.3 Access Point Set Up with Pre-Shared Key

2.3.1 Virtual Access Point (VAP)

Any wireless NIC that can be put into `master mode` can function as an AP. To create a VAP under the MadWifi drivers, use the `wlanconfig` tool like this:

```
wlanconfig ath create wlandev wifi0 wlanmode ap
```

2.3.2 brctl

To provide Internet access for your clients, you will have to set up a bridge between the Ethernet and wireless interfaces on the AP machine. To do so, we will be using a tool called `bridge-utils`. To create a bridge do this:

```
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 ath0
ifconfig br0 up
```

To check the status of the bridge, type

```
brctl show
```

You need to broadcast a request for an IP address, subnet mask and default gateway to be assigned from a DHCP server to the bridge:

```
dhclient br0
```

To delete the bridge do this:

```
ifconfig br0 down
ifconfig eth0 0.0.0.0 down
ifconfig ath0 0.0.0.0 down
brctl delif br0 eth0
brctl delif br0 ath0
brctl delbr br0
```

2.3.3 Hostapd

We will use the server program ("daemon") `hostapd` in conjunction with a WLAN NIC in master mode to implement an 802.11i standard AP . Start by making a backup copy of the file `/etc/hostapd/hostapd.conf`. Then begin editing the original file. Here you will have to specify different security settings for your virtual AP. Start by setting up `hostapd` as simple as possible, minimizing the editing of the conf file.

The lines

```
interface=ath0
bridge=br0
driver=madwifi
```

in `hostapd.conf` specify the name of your wireless interface, the bridge and the name of your wireless NIC driver. Also the value `wme_enabled` should be set to 0. Make other necessary configurations as well. Your aim is to set up a WPA or WPA2 AP that uses a password-based PSK.

Give some thoughts to the selection of the password that will establish the PSK. You want to adhere to the following or similar restrictions in order to make this experiment both feasible and challenging. The rules of the password selection "game" should satisfy the following conditions:

- the password stems from one and only one properly spelled English word;
- the password includes one or two digits/punctuation marks;
- not more than one capital letter.

Let one member of your group secretly construct the password and carry out the configuration, then the rest of you will try to find it and attack your network.

Start the daemon by executing

```
hostapd -d <pathToConfigFile>
```

The parameter "d" sets the program in debug output mode. See comments in the `hostapd.conf` and [25] for detailed information on configuring the `hostapd`.

Use your personal laptop, mobile phone, player or other WLAN-enabled device to connect to your AP (if you don't have a single WLAN device, ask a student assistant).

2.4 The Password Dictionary Attack

2.4.1 airodump-ng, aireplay-ng and aircrack-ng

A tutorial [26] will be your main guide.

Use `airodump-ng` to record the traffic on your WLAN in order to capture the 4-way authentication handshake. Don't be tempted to manipulate your STA or AP. To speed up the process of catching a 4-way handshake you may try the active deauthentication attack of `aireplay-ng`. Note that `airodump-ng` may in some cases not tell you when you get the handshake. You can try to find the 4-way handshake in your dump using the `Wireshark`, or just proceed to the next step if you believe that you've already got a handshake.

Locate and download a good dictionary from the Internet. For the sake of feasibility don't bet for more than 30 000 words. We suggest the tiny wordlist from [27]. Finally, use `aircrack-ng` to run the attack.

2.4.2 john

`John the Ripper` is one of the most popular password testing programs. Since your password isn't just a plain word, we'll use `john` to apply different variations to our dictionary words. Start with the default set of rules to produce your extended dictionary:

```
john -wordfile=words.lst -stdout -rules > extendedwords.lst
```

The argument `-stdout` tells `john` to output words to the standard output.

Try to run `aircrack-ng` on your extended wordlist. While it runs, look at the rules you're using in the section `[List.Rules:Wordlist]` of `/etc/john/john.conf`. If it takes really long to find your password with the default rules, consider cooperation with your fellow that knows the password in order to write your own rules to speed up the process.

There's a way to go without a temporary file by redirecting the output from `john` to `aircrack-ng`:

```
john -wordfile=words.lst -stdout -rules | aircrack-ng -a 2 \
-e my_ssid -w - dump.cap
```

2.5 Questions

Please include the answers to the following questions in your laboratory report.

Q9. Why is TKIP a more secure encryption mechanism than the one-key WEP alternative?

Q10. Which information is used to compute the PMK and the PTK in the password-based PSK scenario? Why is the offline password brute-force attack possible?

Q11. Does the compromise of password imply the disclosure of traffic from previous sessions?

Q12. Does the use of AES in WPA2-PSK give an advantage against a password dictionary attack, as compared to RC4 in WPA-PSK?

Q13. What would be advantages and disadvantages of using a precomputed database of PMKs in a PSK password dictionary attack?

Q14. Suppose an attacker employs 30 days of processing time on the IBM System p5 supercomputer at NTNU (find out its processing power at [28] and recall the password enumeration speed in the lab). Which minimum requirements would you put on a WPA password to ensure that the probability of successful attack does not exceed 2^{-20}?

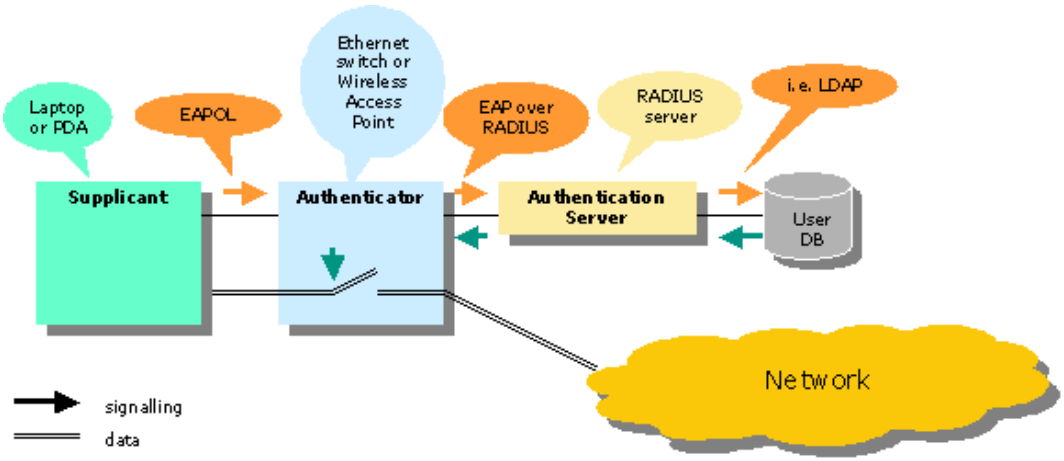

Figure 7: The controlled port concept of 802.1x

3 Setting up RSN-EAP Wireless Access Point

3.1 Objectives

In this part you will set up a wireless AP and configure its security features. You will be implementing the RSN-EAP configuration of the IEEE 802.11-2007 based on two computers equipped with wireless NICs. One will run the authenticator, and also the Radius authentication server (AS), the other will be the supplicant. You should employ:

1. CCMP for enforcing link confidentiality.
2. Mutual authentication of supplicant and AS employing digital certificates and the EAP-TLS authentication protocol.

3.2 Elements of Construction

The system and its components are shown in Figure 7.

Supplicant. A WLAN station requesting wireless Internet access will act as a supplicant in the security protocols.

Authenticator. The role of the AP. Use the `wlanconfig` utility to set up the interface.

Authentication Server. Normally, this role is run by one centralized machine, but in this lab project you will use the same machine to run both an authenticator and an AS. The role of AS is implemented by the software package `FreeRadius`.

When a supplicant attempts to associate with the authenticator, it will rely on the client software `wpa_supplicant` to initiate the authentication request. The authenticator

will then forward the supplicant's request and subsequent EAP messages to the AS, and send AS's EAP messages back to the supplicant. Only after the supplicant is authenticated will it be able to associate with the authenticator (see Figure 8).

There are many EAP protocols that include the use of public-key certificates. Some employ certificates only on the server side, like EAP-PEAP, while others, like EAP-TLS, can use client certificates as well. We will use public-key certificates on both client and server side to ensure mutual authentication in the lab set up.

You will set up an authentication mechanism based on EAP-TLS (considered to be the most robust protocol). You should modify the configuration files (they are split up across several text files according to their functionality), establish your own certificate authority (CA) and create certificates for the server and at least one user. Use `OpenSSL` tool for this. After having created the private keys and certified the public keys indicate the correct paths for the certificate files in the configuration files of both the supplicant and the Radius server.

We recommend reading [19, §8 and §9] or [3, §5.4.3 and §8] to get the background and understanding of the framework you are going to set up in this part.

3.3 Tools and Programs

3.3.1 OpenSSL

Now you separate the responsibilities for each group member, so that one of you is the CA, another one the server owner, and the third one is representing the client. Do the following:

- Create a private, a public key and a self-signed certificate to be used by your CA.
- Create a private and a public key for the server.
- Create a private and a public key for the client.
- Sign the server's and the client's certificates with the CA's private key.

Although you are already familiar with the `OpenSSL` [20] tool (recall the the TTM4135 course), we give a quick summary of the required commands. We use a neat tutorial [21] as a basis.

```
mkdir CA
cd CA
mkdir newcerts private
echo '01' >serial
touch index.txt
```

Note that when copy-pasting from pdf, some symbols, as, for example, quotation marks, may be treated wrong (make sure that the file `index.txt` contains `01`).

Now download the configuration file `openssl.cnf` provided at the end of the [21] page into the `CA` folder and inspect the file. Update security-related parameters with respect to

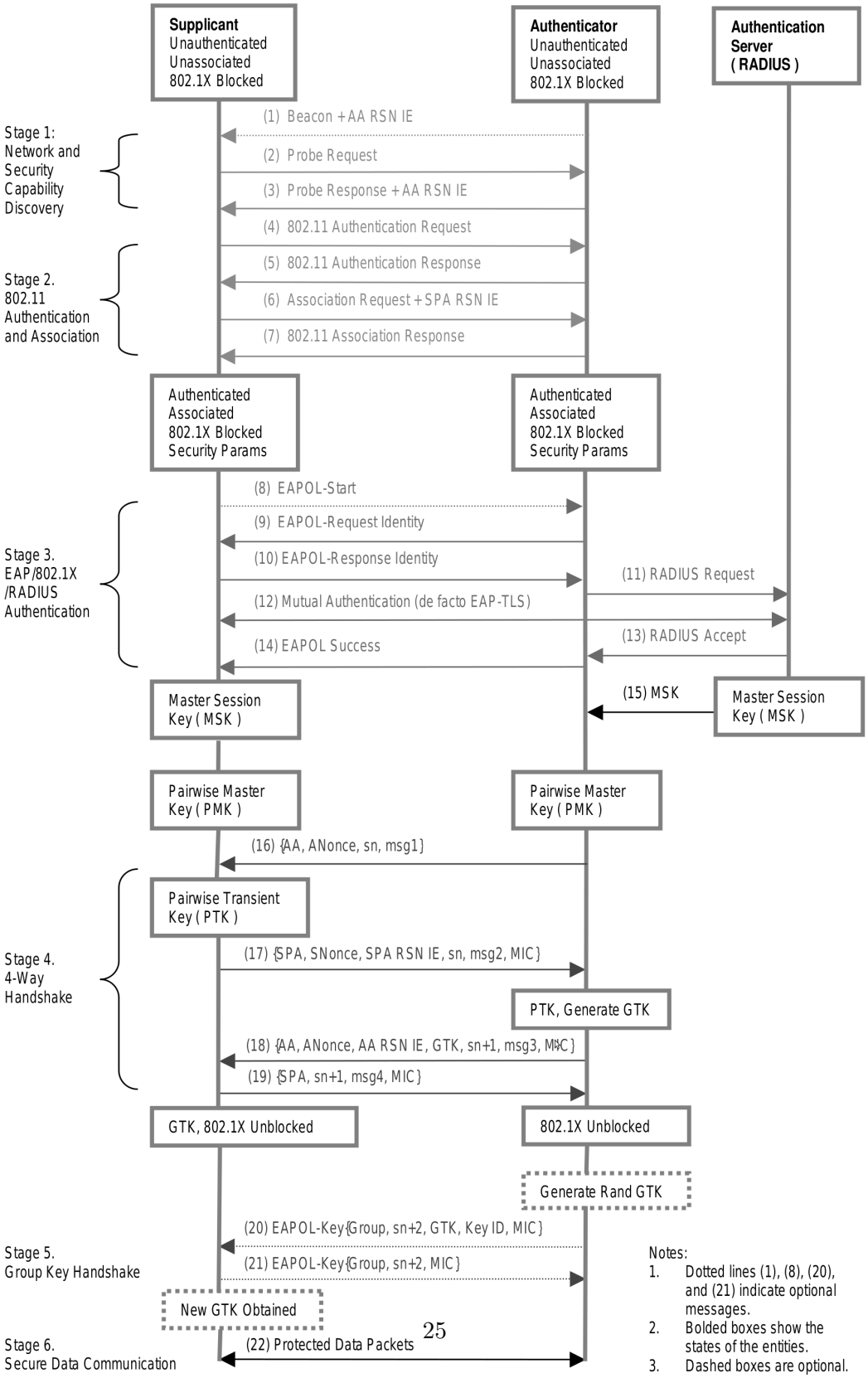

Figure 8: Robust Security Network (RSN) Association Establishment Procedures [18].

the strength of the cryptographic primitives. Also set the following values in order to avoid entering them several times:

```
# Default values for the above, for consistency and less typing.
# Variable name                    Value
#------------------------------   ------------------------------
0.organizationName_default      = NTNU
organizationalUnitName_default  = ITEM
localityName_default            = Trondheim
stateOrProvinceName_default     = Sor-Trondelag
countryName_default             = NO
```

Let's create the CA's self signed certificate first:

```
openssl req -new -x509 -extensions v3_ca -keyout private/cakey.pem \
-out cacert.pem -config ./openssl.cnf
```

During this command execution you will be prompted to create a password for storing the CA's private key.

Then we create certificate requests for the supplicant and the AS. Each new private key will be stored with a new password which you will create:

```
openssl req -new -out supplicantreq.pem \
-keyout private/supplicantkey.pem -config ./openssl.cnf
openssl req -new -out asreq.pem -keyout private/askey.pem \
-config ./openssl.cnf
```

To sign the supplicant's and the AS's public keys, type:

```
openssl ca -out newcerts/supplicantcert.pem -config ./openssl.cnf \
-infiles supplicantreq.pem
openssl ca -out newcerts/ascert.pem -config ./openssl.cnf \
-infiles asreq.pem
```

Use the following command to display the information about a certificate:

```
openssl x509 -in <CertificateFileName> -noout -text
```

You can transfer certificates and keys to clients by using `scp` (secure copy) in one of the following ways:

```
scp <username>@<ipaddress>:<RemoteSourceFile> <LocalDestinationFolder>
scp <LocalSourceFile> <username>@<ipaddress>:<RemoteDestinationFolder>
```

3.3.2 Authentication Server

Use your AP machine from the part two. The back-end AS will be implemented on the same machine as the one running `hostapd`.

`FreeRadius` has several configuration files, but the only ones you will have to modify are `eap.conf` and `clients.conf` in the folder `/etc/freeradius/`. In the former file, set the default EAP method to TLS, and locate the part of the config file specifying the TLS module. After you have created the keys and certificates with `OpenSSL`, you will have to give the correct paths for the key and certificate files. You will also need a file with Diffie-Hellman parameters and another one with random data. Create them by typing

```
openssl dhparam -out dh 1024
openssl dhparam -out random 256
```

Before you start `freeradius`, you may have to stop an already-running instance by typing

```
/etc/init.d/freeradius stop
```

Run the server by typing

```
freeradius -X
```

This will start the daemon in the foreground, letting you see what is going on.

3.3.3 Hostapd

Configure `hostapd` as an AP deploying EAP-TLS, CCMP and an external Radius authentication server (that will run on the same PC). When you are specifying the IP address of the Radius server, simply use `127.0.0.1` (loopback), or localhost. Make sure you comment out the PSK specific configurations used before. Start the `hostapd` process.

3.3.4 Supplicant Side

Start by creating a configuration file for `wpa_supplicant` holding the parameters needed for EAP-TLS. You can use the following template:

```
########################
#  WPA Supplicant config file suggestion
#  WPA2-EAP/CCMP using EAP-TLS
########################

ctrl_interface=/var/run/wpa_supplicant

network={
ssid="your net's ssid"
```

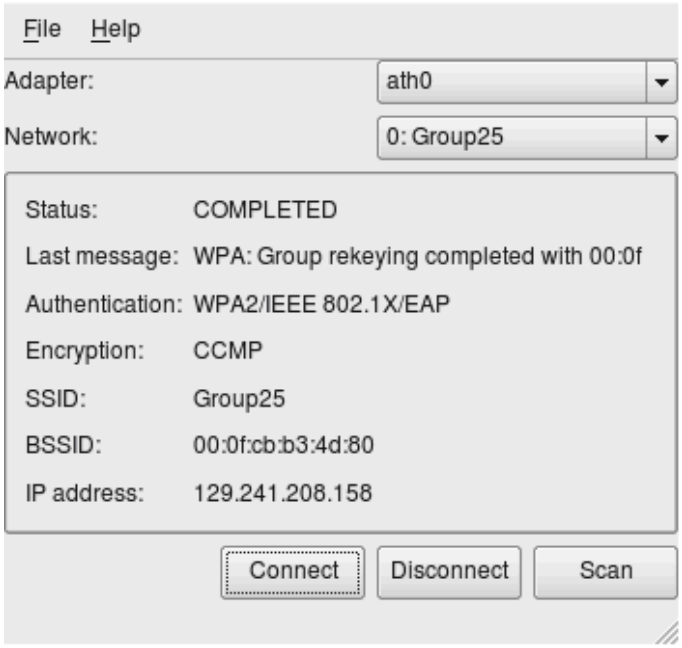

Figure 9: The `wpa_supplicant` with its graphical frontend wpa gui

```
key_mgmt=WPA-EAP
proto=WPA2
pairwise=CCMP
group=CCMP
eap=TLS
ca_cert ="/mylocation/cacert.pem"
client_cert="/mylocation/supplicantcert.pem"
private_key="/mylocation/supplicantkey.pem"
private_key_passwd="passhrase"
identity="any name goes"
}
```

The configuration parser is very picky, for instance, don't include additional spaces in the configuration file. This will give an error saying that the file can't be parsed.

Start the `wpa_supplicant` by the following command:

```
wpa_supplicant -D madwifi -i Ath0 -c <pathToConfigFile>
```

There is a graphical user interface program called `wpa_gui` available. You can try it out if you like (Figure 9).

3.4 Questions

Please include the answers in your laboratory report.

Q15. When would it be appropriate to use WPA2-PSK instead of WPA2-EAP (as key management scheme)?

Q16. EAP-TLS deploys mutual authentication of two communicating parties. What kind of attack is possible against authentication protocols lacking such authentication?

Q17. Give an overview of the EAP authentication protocols which can be used in WPA2-Enterprise WLANs.

Q18. List the security-related protocols used in your RSN-EAP-TLS setup and explain their purpose.

Q19. How can this lab project be improved? (Answering is optional and does not influence your grade.)

Acknowledgement

Many people have contributed to the development and improvement of this lab assignment and its text. Professor Stig F. Mjølsnes started out in the spring of 2006 and set up the framework and the basic content, structure and text. KOMTEK5 student Lars Haukli joined in during the summer of 2006 and made tremendously good progress by testing out and identifying the best NICs and drivers for this purpose. He collected, tested, and selected a working environment of software tools for the Linux platform, and contributed enthusiastically (and mostly after lunch) to the technical content of this lab description. The first students of TTM4137 carried out the assignment with success. All went smoothly, much thanks to dedicated supervision by teaching assistant PhD-student Marie Moe and lab assistant KOMTEK5 student Jan Tore Sørensen. Marie continued as teaching assistant in 2007 and made sure that the experience gained was put to good use in supervision and by editing a new version of the text. The challenge of password dictionary attack was worked out by the teaching assistants PhD-student Martin Eian and PhD-student Anton Stolbunov in 2008.

References

[1] Andreas Klein. *Attacks on the RC4 stream cipher*. Designs, Codes and Cryptography, 48(3):269-286, (2008)

[2] Erik Tews and Ralf-Philipp Weinmann and Andrei Pyshkin. *Breaking 104 Bit WEP in Less Than 60 Seconds*. In Sehun Kim and Moti Yung and Hyung-Woo Lee, editors, WISA, volume 4867 of Lecture Notes in Computer Science, pages 188-202, Springer, (2007)

[3] IEEE Std 802.11-2007, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2007.

http://standards.ieee.org/getieee802/802.11.html, visited 29/08/2011. Also available in the course site at it's learning in folder "Readings".

[4] TTM4137 Lab Group Registration. http://www.item.ntnu.no/fag/ttm4137/ttm4137/, visited 29/08/2011

[5] TTM4137 staff email address. ttm4137@item.ntnu.no

[6] Handbook: Laboratory Reports. http://www.xmarks.com/site/www.ecf.utoronto.ca/~writing/handbook-lab.html, visited 29/08/2011

[7] Citation. http://en.wikipedia.org/wiki/Citation, visited 29/08/2011

[8] Universitetsbiblioteket i Trondheim (UBiT) elektronisk søkeportal. http://www.ntnu.no/ub/eubit/portal.php, visited 15/09/2010

[9] BIBSYS Ask. http://ask.bibsys.no, visited 29/08/2011

[10] Toshihiro Ohigashi1 and Masakatu Morii. *A Practical Message Falsification Attack on WPA*. http://jwis2009.nsysu.edu.tw/location/paper/APracticalMessageFalsificationAttack onWPA.pdf, visited 29/08/2011

[11] Fluhrer and Mantin and Shamir. *Weaknesses in the Key Scheduling Algorithm of RC4*. In SAC: Annual International Workshop on Selected Areas in Cryptography. LNCS, 2001

[12] Andrea Bittau and Mark Handley and Joshua Lackey. *The Final Nail in WEP's Coffin*. In IEEE Symposium on Security and Privacy, pages 386-400, IEEE Computer Society, 2006

[13] Robert Moskowitz. *Debunking the Myth of SSID Hiding*. Technical report. ICSA Labs, a division of TruSecure Corporation, 2003. http://www.library.cornell.edu/dlit/ds/links/cit/redrover/ssid/wp_ssid_hiding.pdf, visited 29/08/2011

[14] Kismet. http://www.kismetwireless.net/, visited 29/08/2011

[15] Aaron Weiss. *Introduction to Kismet*, 2006. http://www.wifiplanet.com/tutorials/article.php/3595531, visited 29/08/2011

[16] Aircrack-ng. http://www.aircrack-ng.org/documentation.html, visited 29/08/2011

[17] Wireshark Foundation. http://www.wireshark.org/, visited 29/08/2011

[18] Changhua He and John C. Mitchell. *Security Analysis and Improvements for IEEE 802.11i*. In NDSS. The Internet Society, 2005.

[19] John Edney and William A. Arbaugh. *Real 802.11 Security*. Addison Wesley, Reading, Massachusetts, July 2003.

[20] The OpenSSL Project. http://www.openssl.org, visited 29/08/2011

[21] Marcus Redivo. *Creating and Using SSL Certificates.* http://www.eclectica.ca/howto/ssl-cert-howto.php, visited 29/08/2011

[22] Guillaume Lehembre. *Wi-Fi security — WEP, WPA and WPA2.* http://www.hsc.fr/ressources/articles/hakin9_wifi/hakin9_wifi_EN.pdf, visited 29/08/2011

[23] Wi-Fi Alliance. The State of Wi-Fi Security. http://www.wi-fi.org/knowledge_center_overview.php?docid=4582, visited 29/08/2011

[24] Robert Moskowitz. *Weakness in Passphrase Choice in WPA Interface*, 2003. http://wifinetnews.com/archives/2003/11/weakness_in_passphrase_choice_in_wpa_interface.html, visited 29/08/2011

[25] Jouni Malinen. *Hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator.* http://hostap.epitest.fi/hostapd/, visited 29/08/2011

[26] DarkAudax. Tutorial: How to Crack WPA/WPA2. http://www.aircrack-ng.org/doku.php?id=cracking_wpa, visited 29/08/2011

[27] English Wordlists. ftp://ftp.openwall.com/pub/wordlists/languages/English/, visited 29/08/2011

[28] Roar Aspli and Pia Bjernemose. NTNU buys Scandinavia's most powerful supercomputer. http://www.ntnu.no/pm/05.06/engelsk_pressemelding.pdf, visited 29/08/2011