WiFi Security – WEP and 802.11i

Levente Buttyán and László Dóra
Laboratory of Cryptography and System Security
Department of Telecommunications
Budapest University of Technology and Economics
{buttyan, doralaca}@crysys.hu

Abstract: *In this paper, we present a tutorial on WiFi security by giving an overview of the related standards such as WEP and 802.11i.*

Keywords: *WiFi, WLAN, WEP, 802.11, 802.11i, WPA, WPA2, RSN, 802.1X, EAP, RADIUS, TKIP, RC4, CCMP, AES, authentication, access control, integrity, confidentiality*

1. Introduction

In the last decade, wireless networks gained a substantial momentum. One of the most beneficial features of wireless networks is that they support user mobility in a convenient way. The downside is that wireless networks are more susceptible to attacks than their wired counterparts. This increased vulnerability mainly stems from the lack of physical connections and the broadcast nature of radio communications. It is, therefore, important to provide appropriate security measures for wireless networks, which ensure the robustness of their operation even in case of malicious attacks.

In this paper, we focus on the security issues related to a particular wireless technology, namely WiFi networks. We do not propose novel results here; our intention is rather to give a tutorial on existing solutions, and to summarize their strengths and weaknesses. In particular, we give an overview of WEP and 802.11i. Interested readers can find more detailed descriptions in several excellent books that have been published on this topic recently. For instance, we found [Edney+04] to be particularly well-written and entertaining.

The rest of this paper is organized as follows: We first present the operation of WEP, which is the security solution that was originally proposed for WiFi networks. However, it has been quickly realized that WEP has several design flaws and it does not provide adequate protection. We briefly summarize these flaws in this paper. Then, we present 802.11i, which can be viewed as the successor of WEP. In particular, we describe how authentication and access control is provided and how the cryptographic keys are derived in 802.11i. We also briefly describe the TKIP and the AES-CCMP protocols, which are used in WPA and RSN (WPA2), respectively.

2. WEP

Security has been considered an important issue in WiFi networks from the beginning. Consequently, early versions of the IEEE 802.11 wireless LAN standard [802.11] have already featured a security architecture, which is called WEP (Wired Equivalent Privacy). As its name indicates, the objective of WEP is to render wireless LANs at least as secure as wired LANs (without particular security extensions). For instance, if an attacker wants

to connect to a wired Ethernet network, (s)he needs physical access to the Ethernet hub. However, this is usually made difficult by placing the hub in a locked room. In case of an unprotected wireless LAN, the attacker has an easier job, because (s)he does not need to have physical access to any equipment in order to connect to the network. WEP is intended to transform this easy job into a difficult one. More precisely, WEP is intended to increase the level of difficulty of attacking wireless LANs such that it becomes comparable to the difficulty of attacking wired LANs (e.g., breaking into locked rooms).

Unfortunately, WEP does not make attacks as difficult as its designers hoped. This would not have been a problem if the weaknesses had been discovered in due time. But things happened differently: WEP has already been deployed when cryptographers and IT security experts discovered its flaws [Walker00, Borisov+01, Arbaugh+02]. It became evident that WEP did not provide adequate protection, and following this discovery, tools that automate the cracking of WEP keys have soon appeared on the Web.

In response to these developments, the IEEE came up with a new security architecture for WiFi networks, which is described in an extension to the 802.11 standard. This extension is called 802.11i. We will discuss 802.11i in the next section, while in this section, we are concerned with WEP. The motivation of discussing WEP is that, despite its known weaknesses, many systems still support it (for backward compatibility), and thus, probably many people and organizations still use it. Those people and organizations should be aware of the limitations of WEP, and consider switching to WPA or RSN (WPA2).

2.1 Operation of WEP

There are two basic security problems in wireless LANs: Firstly, due to the broadcast nature of radio communications, wireless transmissions can be easily eavesdropped. Secondly, and more importantly, connecting to the network does not need physical access to the network access point (AP), therefore, any device can try to illegitimately use the services provided by the network. WEP attempts to solve the first problem by encrypting messages. The second problem is addressed by requiring the authentication of the mobile stations (STAs) before allowing their connection to the network.

The authentication of the STA is based on a simple challenge-response protocol consisting of the exchange of four messages. First, the STA signals that it wants to authenticate itself (authenticate request). In response to this, the AP generates a random challenge and sends it to the STA (authenticate challenge). The STA encrypts the challenge with a secret key known only to the STA and the AP, and sends the result back to the AP (authenticate response). If the AP can successfully decrypt the STA's response (i.e., the decryption results in the same random value that the AP sent to the STA), then it concludes that the response was generated by the STA (since no one else knows the key to generate a correct response), and thus, the STA is authenticated. Otherwise the authentication fails. Based on the result of the authentication, the AP decides if it grants access to the network or not, and informs the STA about its decision (authenticate success/failure).

Once authenticated, the STA communicates with the AP by encrypted messages. The key used for encryption is the same as the one used for authentication. The encryption algorithm specified by WEP is the RC4 stream cipher. Stream ciphers produce a long pseudo-random byte sequence out of a short secret seed value; this pseudo-random sequence is then XORed to the clear message (byte by byte) in order to generate the encrypted message. WEP works in the same way. The sender (the STA or the AP) of a message M initializes the RC4 algorithm with the secret key, and XORs the pseudo-random sequence K produced by RC4 to M. The receiver of the encrypted message $M \oplus K$ uses the same secret key to initialize the RC4 algorithm, which will then produce the same pseudo-random sequence K. Then K is XORed to the encrypted message to obtain the clear message: $(M \oplus K) \oplus K = M$.

In fact, the description above is not precise enough: there is one more thing that WEP does when encrypting messages. It is easy to see that if encryption worked as we described in the previous paragraph, then every message would be encrypted with the same pseudo-random sequence K, since RC4 is initialized with the same secret key before encrypting every message. This would be bad for several reasons. Let us assume, for instance, that an attacker eavesdrops two encrypted messages $M_1 \oplus K$ and $M_2 \oplus K$. By XORing these two messages together, (s)he gets $(M_1 \oplus K) \oplus (M_2 \oplus K) = M_1 \oplus M_2$. This is equivalent to one message being encrypted with the other, but clear messages are far from being pseudo-random sequences. Thus, $M_1 \oplus M_2$ is a very weak encryption, and the attacker is likely to be able to break it based on the statistical properties of the clear messages. It is also possible that the attacker (partially) knows the content of one of the messages (e.g., the value of the header fields), in which case (s)he can easily compute the (partial) content of the other message.

In order to address this problem, WEP appends an IV (Initialization Vector) to the secret key before initializing the RC4 algorithm, where the IV changes for every message. This ensures that the RC4 algorithm produces a different pseudo-random sequence for every message. The receiver should also know the IV in order to be able to decrypt the messages received. For this reason, the IV is sent in clear together with the encrypted message. In principle, this is not a problem, since the knowledge of the IV is not enough to decrypt the message: the secret key is also needed for the proper initialization of the RC4 algorithm. As for the sizes, we note that the IV is 24 bits long and the secret key is (usually) 104 bits long[1]. Figure 1 illustrates the WEP encryption and decryption procedure.

[1] In various marketing materials, this is interpreted as "128-bit security". This is of course misleading (as marketing materials in general), since out of 128 bits 24 bits are transferred in clear, hence known by the attacker.

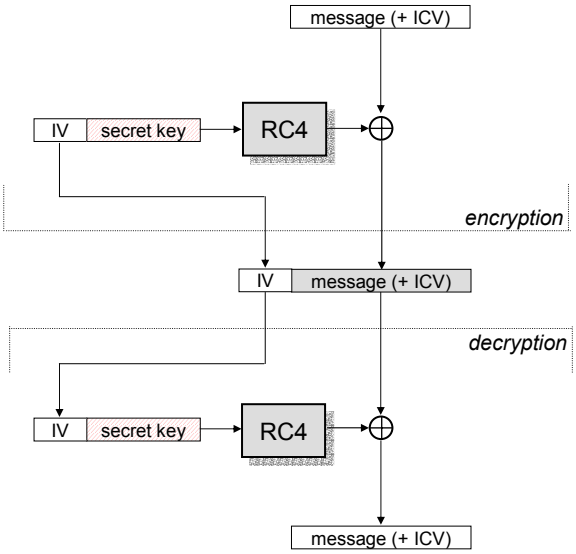

Figure 1: Encryption and decryption in WEP

Figure 1 also shows that before encryption, the sender attaches an integrity check value (ICV) to the clear message. The purpose of this value is to enable the receiver to detect any malicious modifications of the message by an attacker. In case of WEP, the ICV is a CRC value computed for the clear message. Since a CRC value alone cannot enable the detection of malicious modifications (as the attacker can compute the new CRC value for the modified message), the CRC value is also encrypted in WEP. The rationale is that in order to modify the message in an unnoticeable way, now the attacker must encrypt the new CRC value, but (s)he cannot do that without the knowledge of the secret key. This reasoning is not quite true, as we will see below.

Finally, we say some words about WEP keys. The standard supports that each STA has its own key, which is known only to that STA and the AP. However, this makes key management at the AP's side complicated, since the AP must store a key for every STA. For this reason, most implementations do not actually support this option. The standard also specifies a default key, which is known to every STA and the AP. Originally, this key was intended to be used for the encryption of broadcast messages originated by the AP. But most WEP implementations support only this default key. Hence, in practice, in most wireless LANs there is a single common key. This key is installed in every mobile device and in the AP manually. Clearly, this solution can only be used to protect the communications from an outside attacker, but the devices that belong to the network can (in principle) decrypt each other's messages (and impersonate each other).

2.2 WEP design flaws

As it will be clear from the brief overview below, WEP does not actually achieve any of its original design goals. The discovered flaws are instructive; they demonstrate many pitfalls of security protocol design.

Authentication: Authentication in WEP has several problems. First of all, authentication is not mutual, meaning that the AP does not authenticate itself to the STA. Second, the authentication and the encryption mechanism use the same secret key. This is not desirable, since an attacker can exploit the weaknesses of both the authentication and the encryption method to break the secret key. Having different keys for different functions is a better security engineering practice.

The third problem is that the STA is authenticated only at the time when it tries to connect to the network. Once the STA is associated to the AP, anyone can send messages in the name of that STA by spoofing its MAC address. Seemingly, this is not a real problem, since the attacker does not know the secret key which is needed to construct well-formed encrypted messages. Hence, the attacker's messages will be dropped by the AP anyway. But as we mentioned before, often every STA uses the same secret key. This means that the attacker can fabricate messages in the name of one STA by using encrypted messages of another STA recorded earlier. This will not be detected by the AP.

The fourth problem stems from the fact that WEP uses RC4 in the authentication protocol for encrypting the random challenge. Thus, an attacker can easily obtain the challenge C and the encrypted challenge $R = C \oplus K$, from which (s)he can compute the pseudo-random sequence K. However, knowledge of K allows the attacker to impersonate the STA later on, as (s)he can now compute the response $R' = C' \oplus K$ for any other challenge C'. The IV mechanism of WEP does not mitigate this problem, since the IV is selected by the sender of the encrypted message; in our case, the sender is the attacker, who will always select the IV that was appended to R. Moreover, since in practice, every STA uses the same key, the attacker can connect to the network in the name of any STA. Obviously, successful association to the AP is just part of the game; in order to send and receive messages in the name of a legitimate STA, the attacker needs to know the secret key. Other flaws in WEP described later in this paper will allow the attacker to get that one too.

Integrity protection: The integrity protection of WEP messages is based on attaching an ICV to the message, where the ICV is a CRC value computed for the message, and encrypting it with the secret key. Formally, the encrypted message can be written as $(M \,||\, \mathrm{CRC}(M)) \oplus K$, where M is the clear message, K is the pseudo-random sequence produced by the RC4 algorithm from the IV and the secret key, CRC(.) denotes the CRC function, and || denotes concatenation. It is well-known that the CRC function is linear with respect to the XOR operation, which means that $\mathrm{CRC}(X \oplus Y) = \mathrm{CRC}(X) \oplus \mathrm{CRC}(Y)$. Based on this observation, an attacker can manipulate protected WEP messages by flipping any of their bits unnoticeably, although (s)he does not get access to the contents of the messages. Let us denote the changes that the attacker wants to make in the message by ΔM. Then the attacker wants to obtain $((M \oplus \Delta M) \,||\, \mathrm{CRC}(M \oplus \Delta M)) \oplus K$ from the original protected message $(M \,||\, \mathrm{CRC}(M)) \oplus K$ that (s)he eavesdropped. For this purpose, it is

sufficient to compute CRC(ΔM), and then to XOR ΔM || CRC(ΔM) to the original protected message. The following derivation shows why this works:

$$((M \,||\, \mathrm{CRC}(M)) \oplus K) \oplus (\Delta M \,||\, \mathrm{CRC}(\Delta M)) =$$
$$((M \oplus \Delta M) \,||\, (\mathrm{CRC}(M) \oplus \mathrm{CRC}(\Delta M))) \oplus K =$$
$$((M \oplus \Delta M) \,||\, \mathrm{CRC}(M \oplus \Delta M)) \oplus K$$

where in the last step we used the linearity of the CRC function. Since CRC(ΔM) can be computed without the secret key, the attacker can succeed despite the encryption and the ICV mechanism.

Another related integrity requirement is the detection of replayed messages. It is easy to analyze WEP's replay protection mechanism, since it does not exist. The designers of WEP simply forgot about building a replay detection mechanism into the protocol. Hence an attacker can replay any previously recorded message, which will be accepted by the AP.

Confidentiality: As we said before, when using a stream cipher, it is essential that each message is encrypted with a different pseudo-random sequence. In WEP, this is ensured by the IV mechanism; however, this has some problems too. The origin of the problem is that the IV is only 24-bit long, which means that there are only approximately 17 million possible IV values. A WiFi device can transmit approximately 500 full length frame in a second, thus, the whole IV space will be used up in about 7 hours. This means that IVs will be repeating in every 7 hours, and repeating IVs mean repeating pseudo-random sequences used for encryption. The problem is aggravated by the fact that in many networks, there is a single secret key used by every device with potentially different IVs. Hence the IV space will be used up even faster: if there are n devices, then the first IV repetitions are expected after $7/n$ hours. Another practical problem is that in many WEP implementations, the IV is initialized with 0 at startup, and then incremented by one after each message sent. This means that if there are several devices switched on nearly at the same time, then they all use the same sequence of IVs; if, in addition, they use the same secret key too, then the pseudo-random sequences used for encryption will be the same. In this case, the attacker does not even need to wait, but it gets messages encrypted with the same pseudo-random sequence immediately.

The total collapse of WEP is caused by the inappropriate use of the RC4 cipher. It is known that there exist so called weak RC4 keys [Fluhrer+01]. A weak key is a seed value from which the RC4 algorithm produces an output that does not look random. More precisely, when a weak key is used to seed RC4, one can infer the bits of the seed from the first few bytes produced by the algorithm. For this reason, security experts suggest to always throw away the first 256 bytes of the RC4 output. This simple solution would solve the problem of weak keys, but WEP does not adopt it. In addition, due to the ever changing IV value (which is part of the seed), a weak key will be encountered sooner or later, and the attacker will know that a weak key is used, because the IV is transmitted in clear. Based on these observations, some cryptographers constructed a method which breaks the full 104-bit secret key by eavesdropping only a few million messages [Fluhrer+01]. Compared to the previously described flaws, this one is the far most serious, because it allows the attacker to crack the secret key itself, and once (s)he has the secret key, (s)he can do everything. Moreover, the attack is not only powerful, but easy to

automate, and thanks to some “helpful” people, automated attacking tools are readily available on the Web for public use.

3. 802.11i

When the flaws in WEP became apparent, the IEEE started to develop a new security architecture for WiFi networks, which is described in the 802.11i specification [802.11i]. The new concept is called RSN (Robust Security Network) in order to distinguish it from WEP. RSN was designed more carefully than WEP. It includes a new method for authentication and access control, which is based on the model defined in the 802.1X standard. The mechanisms for integrity protection and confidentiality are also changed, and they use the AES (Advanced Encryption Standard) cipher instead of RC4.

However, it is not possible to switch from WEP to RSN overnight. The reason is that for efficiency reasons, many WiFi devices (mainly WLAN adapter cards) support the encryption algorithm in hardware. Thus, old devices support RC4 and not AES. This problem cannot be solved by a simple firmware update; the hardware needs to be changed, which slows done the deployment of RSN.

This has been realized by the IEEE too, and they included an optional protocol in the 802.11i specification, which still uses the RC4 cipher, but fixes of the flaws in WEP. This protocol is called TKIP (Temporal Key Integrity Protocol).

Manufacturers have immediately adopted TKIP, as it provided a solution to the problems of WEP, and it could be deployed immediately without changing the hardware. They did not wait until the 802.11i architecture is finalized by the lengthy standardization procedure, but they issued their own specification, called WPA (WiFi Protected Access), based on TKIP. In other words, WPA is a specification supported by WiFi manufacturers [WPA], and it contains a subset of RSN, which can run on old devices too that support only the RC4 cipher. Authentication and access control, as well as key management are the same in WPA and in RSN, the difference between the two concepts lies in the mechanisms used for integrity protection and confidentiality.

Below, we give an overview of the 802.11i authentication and access control, and the key management procedures; these are the same in WPA and in RSN. Then, we briefly summarize the operation of TKIP (used in WPA) and AES-CCMP (used in RSN).

3.1 Authentication and access control

The model of authentication and access control in 802.11i has been borrowed from the 802.1X standard [802.1X]. 802.1X was originally intended for wired LANs, but it turned out that the same concepts can be used in wireless LANs too (with a few extensions).

The 802.1X model distinguishes three entities in the authentication procedure: the supplicant, the authenticator, and the authentication server. The supplicant would like to access the network, and for this reason it would like to authenticate itself. The authenticator controls access to the network. In the model, this is represented by controlling the state of a port. The default state of the port is „closed”, which means that

data traffic is disabled. The authenticator can „open" the port if this is authorized by the authentication server. Actually, the supplicant authenticates itself to the authentication server, and if this authentication is successful, then the authentication server grants access to the network by instructing the authenticator to open the port.

In case of WiFi networks, the supplicant is the mobile device and the authenticator is the AP. The authentication server is a process, which can run on the AP in case of smaller networks, or on a dedicated server machine in case of larger networks. In WiFi, the port is not a physical connector, but a logical control implemented in software running on the AP.

In a wired LAN, a device authenticates itself once, when it is physically connected to the network. There is no need for further authentication (at least for network access control purposes), because the port used by the device cannot be used by someone else; that would require to first disconnect the device that currently uses the port, which would be detected by the hardware of the authenticator, and the port would be disabled. The situation is different in WiFi networks, because there is no physical connection between the STA and the AP. Hence, once the STA authenticated itself and associated to the AP, someone else may try to steal its session by spoofing its MAC address. For this reason, 802.11i extends 802.1X with the requirement of setting up a session key between the STA and the AP when the STA first requests access to the network; this session key can then be used to authenticate any further communications between the STA and the AP.

The authentication procedure in 802.11i uses EAP (Extensible Authentication Protocol) to carry the messages that need to be exchanged between the STA and the authentication server [EAP]. Note that EAP is only a carrier protocol: it does not provide authentication services itself, but it can carry the messages of any higher layer authentication protocol. That is why it is called „extensible". The way how the higher layer protocol messages are embedded into EAP messages must be specified for each and every higher layer protocol. Such specifications already exist for many widely used protocols such as the TLS Handshake and the GSM authentication protocols.

There are four message types in EAP: request, response, success, and failure. EAP request and response messages carry the messages of the embedded authentication protocol from the STA to the server, and from the server to the STA, respectively. The EAP success and failure messages are used to signal the result of the authentication to the supplicant.

As we indicated before, in 802.1X, the supplicant authenticates itself to the authentication server. This means, that in WiFi networks, the EAP protocol and the embedded higher layer authentication protocol are executed by the mobile device requesting access and the authentication server. The AP only relays messages without interpreting them. The AP understands only the EAP success and failure messages. When it sees an EAP success message passing, it enables the port and lets the mobile device connect to the network.

EAP messages between the mobile device and the AP are carried by the EAPOL (EAP over LAN) protocol defined in 802.1X. EAP messages between the AP and the authentication server can be carried by various protocols. WPA mandates the use of RADIUS [RADIUS] for this purpose, while RSN specifies RADIUS only as an option. In any case, RADIUS is already quite widely deployed, therefore, it is expected that it

will be often used in RSN too. The protocol architecture that we have just described is illustrated in Figure 2.

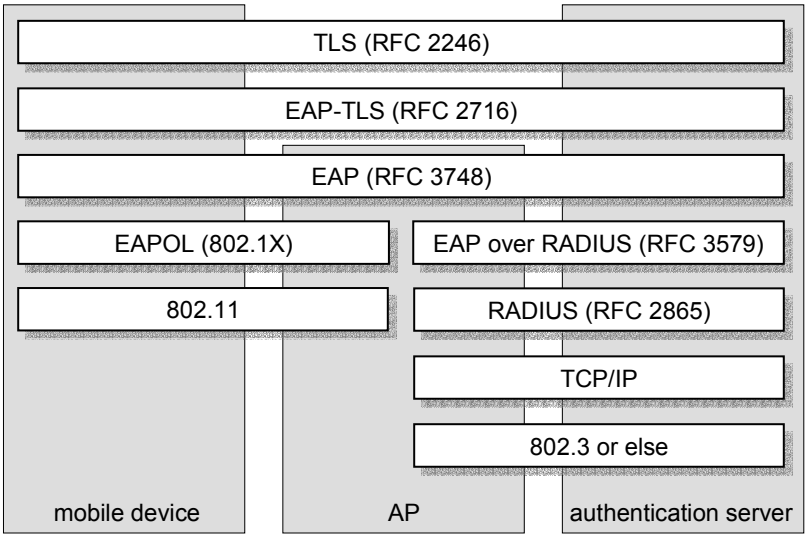

Figure 2: Authentication protocol architecture in 802.11i when TLS authentication is used

As we mentioned before, the result of the authentication process in WiFi is not only the authorization for the mobile device to access the network, but also a session key to protect further communications between the mobile device and the AP. However, as authentication takes place between the mobile device and the authentication server, the session key is established between them too, and it must be securely transferred to the AP. The RADIUS protocol makes this possible by means of the MS-MPPE-Recv-Key RADIUS attribute, which has been specified for key transfer purposes. The session key is transferred in encrypted form, where the encryption uses a long-term key shared by the AP and the authentication server. This latter key is usually installed manually in the AP and in the RADIUS server by the system administrator.

3.2 Key management

The session key established between the mobile device and the AP as the result of the authentication procedure is called pairwise master key (PMK). It is a pairwise key, because it is known only to that mobile device and the AP (and the authentication server, but it is considered to be a trusted entity), and it is a master key, because it is not used directly for encryption or integrity protection of messages, but it is used to derive encryption and integrity keys. More precisely, both the mobile device and the AP derives four keys from the PMK: a data-encryption key, a data-integrity key, a key-encryption key, and a key-integrity key. These four keys together are called pairwise transient key (PTK). We must note that AES-CCMP uses the same key for encryption and for integrity

protection of data, therefore, in case of AES-CCMP, the PTK consists of three keys only. Besides the PMK, the derivation of the PTK also uses as input the MAC addresses of the parties (the mobile device and the AP) and two random numbers generated by the parties. This is illustrated in Figure 3.

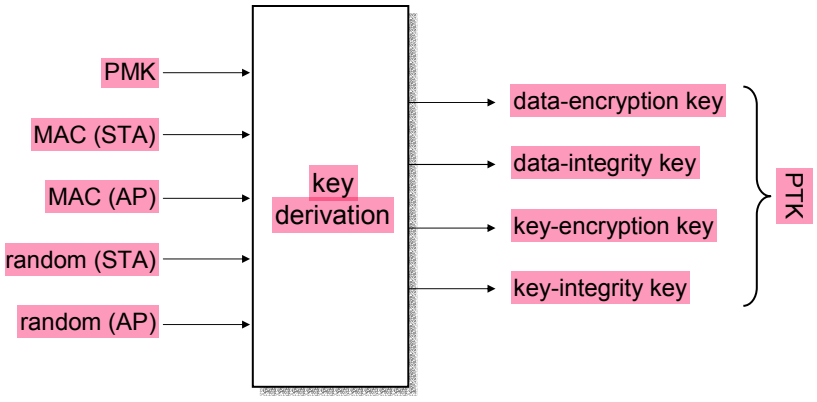

Figure 3: Derivation of the PTK from the PMK, the MAC addresses of the parties, and the random numbers

The mobile device and the AP exchange their random numbers using the so called four-way handshake protocol. This protocol also provides evidence to each party that the other party possesses the PMK. Messages of the four-way handshake protocol are carried by the EAPOL protocol in EAPOL messages of type Key. The contents of the messages and the operation of the four-way handshake protocol are described as follows:

1. First, the AP sends its random number to the mobile device. When the random number is received by the mobile device, it has everything needed to the derivation of the PTK. Hence, the mobile device computes the PTK.
2. The mobile device sends its random number to the AP. This message also carries a Message Integrity Code (MIC), which is computed by the mobile device using the key-integrity key just derived from the PMK. Upon reception of this message, the AP has everything needed to the derivation of the PTK. Hence, the AP computes the PTK, and then uses the key-integrity key to verify the MIC. If the verification is successful, then the AP believes that the mobile device possesses the PMK.
3. The AP sends a message that contains a MIC to the mobile device. The MIC is computed using the key-integrity key of the PTK. If the mobile device can

successfully verify the MIC, then it believes that the AP possesses the PMK too. This message also contains the starting value of a sequence number that will be used to number further data packets, and hence, to detect replay attacks. This message also signals to the mobile device that the AP has installed the keys and it is ready for encrypting all subsequent data packets.

4. Finally, the mobile device acknowledges the reception of the third message. This acknowledgement also means that the mobile device is ready for encrypting all subsequent data packets.

Once the PTK is derived and the keys are installed, subsequent data packets between the mobile device and the AP are protected by the data-encryption and data-integrity keys. However, these keys cannot be used to protect broadcast messages sent by the AP. Those broadcast messages should be protected with keys that are known to *all* mobile devices and the AP. Therefore, the AP generates additional key material, which is called group transient key (GTK). The GTK contains a group encryption key and a group integrity key, and it is sent to each mobile device separately encrypted with the key-encryption key of the given mobile device.

3.3 TKIP and AES-CCMP

Both TKIP (Temporal Key Integrity Protocol) and AES-CCMP (AES CTR Mode and CBC MAC Protocol) are based on the key hierarchy described in the previous subsection. In particular, they use the data-encryption and data-integrity keys (of the PTK) to protect the confidentiality and the integrity of the data packets sent between the mobile device and the AP. However, they use different cryptographic algorithms. TKIP, just like WEP, uses RC4, but unlike WEP, it provides real security. The advantage of TKIP is that it runs on old WEP hardware after some firmware upgrade. AES-CCMP needs new hardware that supports the AES algorithm, but it provides a clearer and more elegant solution, than TKIP does.

TKIP fixes the flaws in WEP as follows:

Integrity: TKIP introduces a new integrity protection mechanism called Michael. Michael operates at SDU level (i.e., it operates on data received by the MAC layer from higher layers before those data are fragmented). This makes it possible to implement Michael in the device driver, which in turn allows the introduction of Michael as a software upgrade.

In order to detect replay attacks, TKIP uses the IV as a sequence number. Thus, the IV is initialized with some initial value and then incremented after the transmission of every message. The receiver keeps track of the IVs of the recently received messages. If the IV of a freshly received message is smaller than the smallest stored IV value, then the receiver drops the message, while if the IV is larger than the largest stored IV value, then it keeps the message and updates its stored IVs. If the IV of an incoming message falls between the smallest and the largest stored IV value, then the receiver checks if that IV is already stored; if so, then it drops the message, otherwise it keeps the message and stores the new IV.

Confidentiality: Recall that the main problem with WEP encryption was that he IV size was too small and that the existence of RC4 weak keys was not taken into consideration. In order to overcome the first problem, in TKIP, the IV size is increased from 24 bits to 48 bits. This seems like an easy solution, but the difficulty is that the WEP hardware still expects a 128-bit long RC4 seed value. Thus, the 48-bit IV and the 104-bit key must somehow be compressed into 128 bits. As for the problem of weak keys, in TKIP, each message is encrypted with a different key. Thus, the attacker cannot observe sufficiently many messages that are encrypted with the same (potentially weak) key. The message keys are generated from the data-encryption key of the PTK. TKIP's new IV mechanism and the generation of the message keys are illustrated in Figure 4.

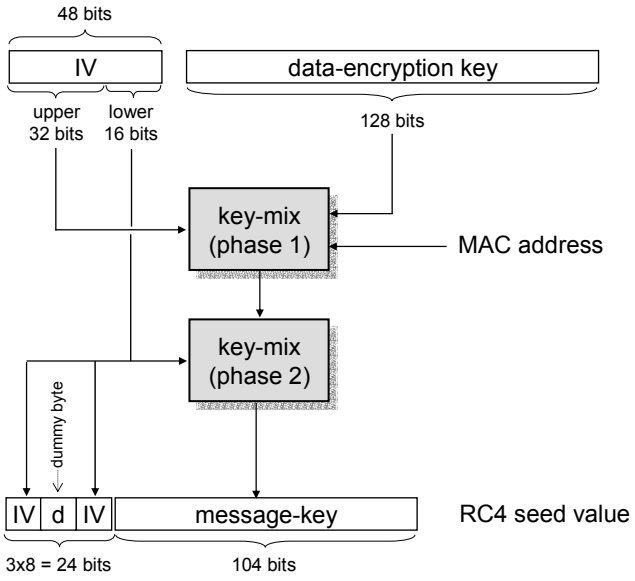

Figure 4: Generation of the RC4 seed value in TKIP

The designers of AES-CCMP had an easier job than the designers of TKIP, beacuse they were not constrained by the peculiarities of the old WEP hardware. Thus, they simply got rid of RC4 and based their design on the AES block cipher. They defined a new mode for AES, called CCM, which is the combination of two previously known mechanisms: CTR (Counter) mode encryption and CBC MAC (Cipher Block Chaining – Message Authentication Code). In CCM mode, the sender of a message computes the CBC MAC value of the message, attaches it to the message, and then encrypt the whole lot in CTR mode. The CBC MAC computation covers the header of the message too, while the encryption is applied only to the message body. The CCM mode ensures both confidentiality and integrity of the message. Replay detection is ensured by sequence numbering the messages. The sequence number is integrated into the CBC MAC value of the message by placing it in the initialization block of the CBC MAC computation.

4. Summary

In this paper, we gave a tutorial on WiFi security by reviewing the related standards: WEP and 802.11i. We presented the operation of WEP, and described its weaknesses. We also described the authentication, access control, and key management mechanisms of 802.11i, as well as the TKIP and the AES-CCMP protocols. As we saw, the advantage of TKIP is that it makes possible the usage of a subset of the 802.11i security architecture on old devices that supports only the RC4 cipher. The requirement of compatibility with old WEP hardware was a serious design constraint, and therefore, TKIP is not very elegant. On the other hand, AES-CCMP is secure and elegant, but it needs new hardware that supports the AES cipher.

5. Acknowledgement

This work has been partially supported by the Mobile Innovation Center (www.mik.bme.hu) and NKFP (under contract number 2 027 04).

6. Bibliography

[Arbaugh+02] W. Arbaugh, N. Shankar, J. Wan, K. Zhang. Your 802.11 network has no clothes. *IEEE Wireless Communications Magazine,* 9(6):44-51, 2002.

[Borisov+01] N. Borisov, I. Goldberg, D. Wagner. Intercepting mobile communications: the insecurity of 802.11. *Proceedings of the 7th ACM Conference on Mobile Computing and Networking*, 2001.

[EAP] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz. Extensible Authentication Protocol (EAP). RFC 3748. 2004.

[Edney+04] J. Edney, W. Arbaugh. *Real 802.11 Security: WiFi Protected Access and 802.11i*. Addison-Wesley, 2004.

[Fluhrer+01] S. Fluhrer, I. Mantin, A. Shamir. Weaknesses in the key scheduling algorithm of RC4. Proceedings of the 8th Workshop on Selected Areas in Cryptography. 2001.

[RADIUS] B. Aboba, P. Calhoun. RADIUS (Remote Authentication Dial In User Service) Support for Extensible Authentication Protocol (EAP), RFC 3579, 2003.

[Walker00] J. Walker. Unsafe at any key size: An analysis of the WEP encapsulation. *IEEE 802.11-00/362*, 2000.

[WPA] Wi-Fi Alliance. Wi-Fi Protected Access. http://www.wi-fi.org/white_papers/whitepaper-042903-wpa/ (accessed on April 19, 2006)

[802.1X] IEEE Std 802.1X-2001. IEEE Standard: Port-based Network Access Control, 2001.

[802.11] IEEE Std 802.11. IEEE Standard: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.

[802.11i] IEEE Std 802.11i. IEEE Standard Amendment 6: Medium Access Control (MAC) Security Enhancements, 2004.