ingenieur
wissenschaften
htw saar

Hochschule für
Technik und Wirtschaft
des Saarlandes
University of
Applied Sciences

STL SYSTEMTECHNIKLABOR

Attacking Wireless Networks

Mark Matinussen and Matthias Riegler

Technical Report – STL-TR-2016-08 – ISSN 2364-7167

stl.htwsaar.de

Technische Berichte des Systemtechniklabors (STL) der htw saar
Technical Reports of the System Technology Lab (STL) at htw saar
ISSN 2364-7167

Mark Matinussen and Matthias Riegler: Attacking Wireless Networks
Technical report id: STL-TR-2016-08

First published: November 2017
Last revision: July 2016
Internal review: Sven Bugiel

For the most recent version of this report see: https://stl.htwsaar.de/

Title image source: pprobbins, http://www.freeimages.com/photo/1168197

htw saar – Hochschule für Technik und Wirtschaft des Saarlandes (University of Applied Sciences)
Fakultät für Ingenieurwissenschaften (School of Engineering)
STL – Systemtechniklabor (System Technology Lab)
Prof. Dr.-Ing. André Miede (andre.miede@htwsaar.de)
Goebenstraße 40
66117 Saarbrücken, Germany
https://stl.htwsaar.de

Attacking wireless networks

Written for the seminar "Hacking" at Saarland University on 28th July, 2016

Mark Matinussen
HTW Saar
Saarbrücken, Germany
pib.mark.martinussen@htwsaar.de

Matthias Riegler
HTW Saar
Merzig, Germany
pib.matthias.riegler@htwsaar.de

***Abstract*—Nowadays, there are wireless networks absolutely everywhere, from schools to coffee shops to our homes. But wireless networks are radio waves, meaning everyone with a receiver can receive the traffic. To solve that problem, there exist cryptographic extensions for wireless networks. But there is no guarantee that these extensions actually provide sufficient confidentiality and integrity. Bad implementations, design flaws or broken cryptographic primitives can make these systems susceptible to a multitude of attacks. This paper presents the most important ones.**

I. Introduction

People like to communicate with each other, they like to talk, share ideas, dreams and qualms, lie, complain and rejoice. The Internet provides an exceptionally good platform for this, never before have we been able to talk to more people simultaneously than nowadays. We access it from anywhere, with our mobile phones, laptops, and watches, wireless networks make this Internet-connectivity ubiquitous.
However, are we aware how many others listen in on our chatter? Wireless devices send out radio waves, so how do we know our personal information is not sent to some conspicuous looking guy with a mask?

This paper deals with the topic of wireless security, how wireless networks are secured and summarises different methods of breaking confidentiality. It does not aim to provide in-depth explanation or proof of correctness for any of the attacks, instead it is merely thought as an overview, encouraging to read up more on the topics. We will move from topic to topic, presenting hacks and ultimately evaluating if modern wireless infrastructure even can provide security.

II. Overview of wireless networking

Nowadays, wireless networks are an integral part of our computer infrastructure, even though they are so common, attacking them is not always trivial. To get a better idea of how an attack works and at which point it is mounted, we need a basic understanding how wireless networks operate.
A wireless network, or Wi-Fi, as defined in IEEE 802.11 [1] consists of a client with a wireless link and a wireless access point. The access point (*AP*) and the wireless link of the hosts make up the essential infrastructure. It is important to note that access points are link layer devices, meaning they do not use sophisticated protocols. Just as in wired transmission, the protocol defines a frame with which data is sent. The data is encapsulated, sent and upon arriving at the *AP*, disassembled and used to create a normal Ethernet frame for further routing.

A. Open WLANs

The problem with wireless networks is that they are constantly broadcasting. The wireless networks use radio frequencies generally ranging from 2,4GHz to 5GHz which everybody in range can intercept.

This makes unprotected Wifi's exceptionally dangerous, as it is possible to mount pretty much any attack imaginable.
From simply capturing the traffic and viewing potentially sensitive information about the user (Sniffing) to hijacking sessions by modifying incoming packets and thus opening up the possibility for spoofing or placing malicious software on the network or even the access point to turn it into a bot.
There are also some more sophisticated attacks. The 802.11 frame has a field called `Duration`, It is used to rule the time a frame may reserve a specific frequency. Given enough frames populating the network on multiple frequencies, an attacker can mount a Denial of Service attack. This is done by inserting himself in an MITM-like manner into the network, capturing but also still forwarding frames. He would only modify said field with an arbitrary value, which will force the *AP* to keep the frequency reserved even though no data is being sent.

B. KARMA attacks

KARMA attacks base on automatic wireless network selection, short AWNS. AWNS is a comfort feature of all modern operating systems. When connecting to a wireless network, the host saves the SSID of this network locally, over time creating a list of preferred networks. All access points in a network regularly send out beacon frames containing basic information about that *AP* such as SSID and supported data rates. [1] A newly joining host can capture these to determine whether a preferred network is in range. Moreover, the host sends out *probe requests*, actively searching for the listed networks. If he gets a matching reponse he will automatically connect with the saved credentials.
An attacker can monitor these probe requests and copy the list of a target's preferred networks. Using this list, he can create a fake network, to which the target will try to connect. [2] Now, he can perform any kind of malicious actions due to being in control of the *AP*.
An adversary could also kick all the users from an existing network and upon them reconnecting, pretend to be the network they were connected to before.

III. Wired Equivalency Privacy

To tackle the problems of unencrypted networks, the initial release of IEEE 802.11 also had build-in encryption called

WEP. WEP is a symmetric encryption scheme which also supports very basic client-*AP* authentication. The authentication process uses encrypted 128-byte nonces, which are sent by the access point, and decrypted by the client. Key exchange is not supported by WEP.

The key is such presumed to be shared via some other utility. It is either 40 or 104 bits long, while for each frame, an additional 24 bits of random Initialisation Vector is prepended. This key is used to create a keystream material with the RC4 stream cipher. The produced keystream are XORed bytewise with the plaintext and the IV appended to the frame.

A. The FMS attack

Only two years after the release of WEP, S. Fluhrer, I. Martin and A. Shamir released an attack on RC4, this attack was effective enough to lead to the connotation that WEP is broken. The FMS attack abuses a weakness in the Pseudo-random generator (PRG) of RC4 which allows an attacker to recover the key given he can capture a few million frames.

RC4 consists of two algorithms, one algorithm which permutes the input key into an array of the numbers 0-255 and initializes an internal state (KSA), and the second one to create a single pseudorandom byte of key material from that state (PRGA).
The attack abuses the fact that on one hand the first 24 bits of the complete key (IV and key) are sent in plaintext and on the other that guessing the first byte of the encrypted message is feasible. The most apperant design flaw of WEP is, that there are only 2^{24} possible IVs, which implys that an IV has to be reused each few million frames. Due to the predictability of first few keybyts of the KSA, an attacker can retrace the steps of the KSA and from that point on and can predict the output of the next step of the key-scheduling algorithm. In fact, an adversary has a $\sim 5\%$ chance that the internal state of the algorithm does not change until the end of the first algorithm, making the prediction easy. [3], [4]

B. KoreK attacks

In 2004, a user of the internet forum *netstumbler*, called KoreK posted an implementation of a new WEP cracker, which was able to retrieve WEP passphrases much quicker than the FMS attack. Instead of focusing on a single correlation between two generated bytes of the KSA, this implementation uses 17.
Similarly to the FMS attack, the gist of the attack is, that knowing the first n bytes of the key scheduling and the first two bytes of the output is enough information to retrieve the $n+1$th byte of key scheduling. [5]
Having access to more than a single correlation increases the effectiveness manifold. In the set of attacks, there are three groups: The first in which also the *FMS attack* is included, which as described above, uses the first n bytes of the KSA and the first byte of the output. The second group additionally uses the second byte of the output and the third group, which are called *inverse attacks*, play a major role in reducing the computational complexity by excluding certain values.

KoreK further developed an experimental attack on WEP he named *chopchop*. It is quite extraordinary as it is not a key recovery attack, but instead, uses the CRC Checksum and the access point which acts as an oracle to decrypt the packet. The attack can be described as follows:
Given an arbitrary packet of length l with a checksum and an oracle O_{AP} that returns whether the checksum of the packet was correct or not. Checksumming in WEP protected networks is done via a simple CRC32 checksum that is appended at the end of the frame. An attacker can now capture a single encrypted packet, removing the lth byte and then taking a guess to what is was, reappends it and corrects the checksum. He then sends it to the oracle, if O_{AP} returns true, he guessed correctly and can now repeat the with l-1. In average, he needs to query O_{AP} $128 \cdot l$ times to fully decrypt the packet. [4], [6]

C. PTW attack

In 2007, the attacks on WEP were improved even further with the development of the PTW attack by Pyshkin, Tews and Weinmann. [5] While all previous attacks relied on at least a few values not changing and thus a larger amount of frames had to be captured, the PTW attack implemented different, much more efficient conditions.
The PTW attack uses the *Klein Correlation* [7], which has no requirements to the internal state of RC4. Besides that, it gets rid of some if the conditional decision-making of FMS in favor of a more general system.

IV. Wi-Fi Protected Access

In an effort to substitute WEP with something more secure, the IEEE released a new security standard for IEEE 802.11 networks - IEEE 802.11i. In the meantime, a new encryption method surfaced called WPA. WPA was thought to be the direct replacement for WEP, thus it was designed to run on any hardware, even the old which was built for WEP. Thus, WPA still uses a derivative of the RC4 stream cipher, TKIP (Temporal Key Integrity Protocol). Moreover, a more sophisticated method for integrity checking was introduced with MICHAEL (MIC). This was done to prevent some attacks, including the base *chopchop* attack, as the normal CRC32 mechanism is very weak. Additionally, a protection against simple replay-attacks was implemented with a sequence counter, which when not increased as expected will drop the packet. Still, WPA was only intended to be a temporary replacement and as such, the legacy of RC4 is still included which makes it very much vulnerable.

Breaking TKIP

In 2008, Erik Tews and Martin Beck published an MIC key recovery attack on TKIP that allows an attacker to decrypt and send a small number of arbitrary packets [4]. This attack is based on the previously discussed *chopchop* attack but had to be changed slightly to account for the improvements made in TKIP over WEP.
While the attack works best when the network supports IEEE 802.11e QoS [1], which enables simultaneous communication over up to 8 independent channels, it is not obligatory. The QoS features are beneficial for an attacker, as they allow to bypass the sequence counter by sending the packet on a different channel then it was received on. The different channels have their own sequence counter. Is IEEE 802.11e not supported by the targeted network, the attacker has to disconnect the client from the *AP* to prevent him from producing more traffic

which would increase the sequence counter. Either way, an attacker would listen for ARP communication with which he would begin the attack. The benefit of ARP is, that it has an easily recognisable structure and is standardised thus a lot of the encrypted data is already known. As mentioned above, the legacy of WEP lives on in TKIP. This means that even though MIC is now used for integrity checking, the CRC32 checksum is still present.
The *chopchop* attack has to be modified in so far that while a packet with a wrong CRC32 checksum is dropped silently by the oracle, the MIC may only be wrong twice each 60 seconds, otherwise the connection is suspended. This means that if the attacker makes an incorrect guess, the packet is dropped without warning, and on a correct guess, which will be indicated by a *MIC failure report* frame, he has to wait one minute before guessing the next time. Given enough time, he can recover the entire ARP request, reverse the MIC algorithm and recover the key. He can then send further packets to the network to recover more information and ultimately, decrypt arbitrary packets. [8]

V. Wi-Fi Protected Access 2

Succeeding WPA was WPA2, which was ratified in 2004 and is since the recommended standard for wireless network encryption. WPA2 got rid of RC4-based encryption and instead implements AES-based encryption. Additionally, message integrity and authentication are done by means of a CBC-MAC. Both WPA and WPA2 have two operating modes - Personal and Enterprise, most importantly they differentiate in how authentication is done. In Personal mode, a user enters his credentials as either an 8 to 63 character ASCII passphrase or an 64 Hex-character string. [1]
Either way, the algorithm converts this string by some operations into a Pairwise Master Key (PMK). In Enterprise Mode, however, the client makes use of the 802.1X/EAP framework to authenticate, which at the end results in the creation of a Master Session Key (MSK). This key is only known by the authentication server and the client and is used to create the respective PMK which is sent to the *AP*. The goal is to expose as little as possible of the original key. Please consider Figure 1 for further details on the key hierarchy.

Attacking WPA2-secured networks

AES encryption is secure which is why no key recovery attacks exist. There are only very few cases where an adversary is able to retrieve any information. The most obvious attack is on the pre-shared passphrase. It is possible to brute force the PSK if the user has picked a common or weak phrase.
Furthermore, the Group Temporal Key (GTK), receives a lot of exposure because it is used for broadcast messages and therefore often used and vulnerable (Hole196 [9]). Though retrieving a GTK only gives an attacker minimal access to the network.

The only practical attack at this moment is centered around capturing the PTK during the four-way handshake between client and *AP*. Imagine a scenario where the authentication process is complete (the PMK is exchanged). The next step is to share the PTK, which contains keys for unicast integrity checking and encryption. The handshake starts off by the *AP* sending a random value, an anonce to the client. The client

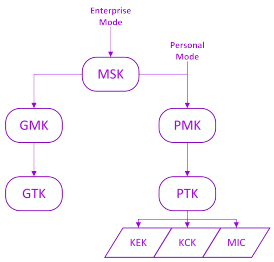

Fig. 1. Key Schedueling in 802.11i [1]

picks a random value, retrieves the MAC addresses of both partners and computes the PTK. Then it sends its own nonce back so that the *AP* can perform the same calculation.
If an attacker listens in on this exchange he can capture both nonces and within a few steps retrieve the MAC addresses of the partners (for example by inspecting ARP traffic). This is enough information to brute-force the PTK. Being in possesion of the PTK the adversary can decrypt any unicast messages sent between the client and the access point. Luckly the PTK is temporary and will be renewed regularly. Due to the key hierarchy shown in Figure 1 he is not able to derive any other keys from the PTK such that he would gain access to the MSK.

VI. Conclusion

Conclusively, there are definitely options for securing wireless networks, even in the very early times confidentiality was taken into account. As time went on, these options evolved, more sophisticated cryptographic principles were implemented and variety rose. Nevertheless, viable attacks exist, even on WPA2. Thus one can not just sit back and rely on being completely secure.

References

[1] IEEE Computer Society, "802.11-2012 - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Standard, IEEE Std 802.11™- 2012, March 2012. [Online]. Available: http://standards.ieee.org/findstds/standard/802.11-2012.html

[2] D. A. D. Zovi and S. A. Macaulay, "Attacking Automatic Wireless Network Selection," Square New York, Paper, 2005. [Online]. Available: http://theta44.org/karma/aawns.pdf

[3] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4," Cisco Systems, Inc. and Computer science department of the Weizmann Institute, Tech. Rep., 2001. [Online]. Available: http://www.crypto.com/papers/others/rc4_ksaproc.pdf

[4] M. Beck and E. Tews, "Practical attack against WEP and WPA," TU Darmstadt and TU Dresden, Paper, 2008. [Online]. Available: http://dl.aircrack-ng.org/breakingwepandwpa.pdf

[5] E. Tews, "Attacks on the WEP protocol," diploma thesis, TU Darmstadt, 2007, cryptology ePrint Archive, Report 2007/471, http://eprint.iacr.org/2007/471.

[6] Aircrack-ng, "Chopchop theory," November 2010. [Online]. Available: http://www.aircrack-ng.org/doku.php?id=chopchoptheory

[7] A. Klein, "Attacks on the rc4 stream cipher," *Designs, Codes and Cryptography*, vol. 48, no. 3, pp. 269–286, 2008. [Online]. Available: http://dx.doi.org/10.1007/s10623-008-9206-6

[8] M. Vanhoef and F. Piessens, "Practical Verification of WPA - TKIP Vulnerabilities," KU Leuven, Paper, 2013. [Online]. Available: https://lirias.kuleuven.be/bitstream/123456789/401042/1/wpatkip.pdf

[9] M. Agarwal, S. Biswas, and S. Nandi, "Advanced stealth man-in-the-middle attack in wpa2 encrypted wi-fi networks," *IEEE Communications Letters*, vol. 19, no. 4, pp. 581–584, April 2015.