Securing home Wi-Fi with WPA3 personal

1st Erik Lamers
Security & Network Engineering
University of Amsterdam
Amsterdam, The Netherlands
elamers@os3.nl

2nd Raoul Dijksman
Security & Network Engineering
University of Amsterdam
Amsterdam, The Netherlands
rdijksman@os3.nl

3rd Arjan van der Vegt
Technology - Connectivity
Liberty Global
Schiphol-Rijk, The Netherlands
avdvegt@libertyglobal.com

4th Mayur Sarode
Technology - Connectivity
Liberty Global
Schiphol-Rijk, The Netherlands
msarode@libertyglobal.com

5th Cees de Laat
Security & Network Engineering
University of Amsterdam
Amsterdam, The Netherlands
delaat@uva.nl

***Abstract*—Wi-Fi Protected Access 3 (WPA3) has become a mandatory part of the Wi-Fi certification on July 1st 2020. Therefore, the adoption rate of WPA3 is expected to grow soon. In this paper, we focus on WPA3 personal transition mode, in particular the security of this mode. We argue that transition mode is a requirement in home environments for the foreseeable future. We investigate whether it is possible to secure a WPA3 personal transition mode network in such a way that downgrade attacks are not feasible. We find that even with the security recommendations that the Wi-Fi Alliance recently issued for WPA3, common implementations running in transition mode can still be downgraded to WPA2. In our experiments, we can see that there are differences between WPA3 implementations in terms of security. The Wi-Fi Alliance has already announced upcoming additions to the WPA3 standard. These additions offer essential improvements to the security of WPA3 personal transition mode networks. We believe that the WPA3 certification should be extended to include the recently announced additions to WPA3. On top of this, we make several recommendations to ensure the safe operation of WPA3. Together these changes will resolve most of the implementation differences we observed. Furthermore, we argue that mutual authentication is an essential stepping stone towards a more secure Wi-Fi ecosystem and discuss two mechanisms.**

***Index Terms*—Wi-Fi, WPA3 personal, SAE, SAE-PK, transition mode**

I. Introduction

The Wi-Fi Alliance announced Wi-Fi Protected Access 3 (WPA3) in June of 2018 [1]. WPA3 has become mandatory for Wi-Fi certified implementations on July 1st 2020 [2, 3]. Therefore, the adoption rate of WPA3 is expected to grow reasonably soon. However, as we can see from the current statistics, WPA3 is not actively used by the public at this moment. We can see that at the 1st of May in 2020, only 24 out of roughly 645 million access points that have been recorded by WiGLE use WPA3 [4]. With more people working from home because of the COVID-19 pandemic, the security of home Wi-Fi has become more important than ever before.

In the past decade, more and more ways were found to crack WPA2 Pre-Shared Key (WPA2-PSK) [5]. This means that a new secure authentication mechanism for personal Wi-Fi is needed. WPA3 aims to fix a lot of the security issues in WPA2. WPA3 is primarily a revision of the Wi-Fi handshake mechanism. This means that many of the pros and cons of WPA2 will be carried over to WPA3, with which it shares many techniques in common [6]. In WPA3 personal the traditional IEEE 802.11i four-way handshake is preceded by Simultaneous Authentication of Equals (SAE) based on the Dragonfly handshake [7, 8, 9].

To use WPA3 exclusively without WPA2 fallback, quite a modern and high-end network is required. Legacy devices, including most modern Internet of Things (IoT) hardware, will be unable to connect to such a network. For a heterogeneous and organically grown network of devices, it will take quite a while for the WPA2-only devices to be updated or replaced. Many of the smart home and IoT devices often do not support WPA3 at all. We argue that access points (APs) with WPA3 need to be backwards compatible with WPA2, simply because older devices will have to keep functioning for the foreseeable future. This is where WPA3 transition mode (WPA3-TM) comes in, which makes Protected Management Frames (PMF) optional and allows stations to connect using either WPA3 (if supported) or WPA2. The specification states that stations should use the WPA3-SAE authentication only with PMF enabled [7]. The downside of running WPA3 in transition mode is that stations can easily be downgraded back to WPA2, reintroducing all the security flaws that WPA3 is trying to fix. The possibility of this downgrade attack has already been shown in the Dragonblood paper [10].

For this research, we will focus on personal Wi-Fi implementations of WPA3 and all the possibilities that come with these implementations. We limit ourselves to WPA3 personal mode in a home network, as this will be the main topic of our research question. The 'home' implementation often consists

of one AP/gateway that serves multiple stations. Currently, this setup will most likely run WPA2 personal [4]. The experiments and recommendations will be tailored towards such a network setup, making this research relevant to both home users and the ISPs that are currently implementing WPA3. Because we focus on the home setup, we will utilise commonly seen Wi-Fi stations in a home setup for our research clients. Furthermore, we aim to make recommendations that could improve the overall security of WPA3 and aim to make clear what a minimal secure WPA3 rollout would look like. The cryptography used within WPA3 is out of scope.

In this paper, we will answer the following question: **How can WPA3 personal transition mode be secured in such a way that downgrade attacks are not feasible?** To answer this question, we define the following sub-questions:

- How can WPA3 personal transition mode be manipulated in downgrading clients to WPA2?
- What techniques can be utilised to prevent these downgrade attacks?

In this paper, we make the following contributions:

- We retest the downgrade vulnerabilities that have been found in the Dragonblood paper [10].
- We make theoretical suggestions on how to improve the security and mutual trust of WPA3 capable devices.

The remainder of this paper is structured as follows: in section II, a literature review of relevant research is given. The methodologies for the experiments are elaborated in section III. Section IV contains the results of the experiments that were conducted. Then we discuss our findings in section V and draw conclusions in section VI. Finally, future work is covered in section VII.

II. Related work

In 2018 Christopher Kohlios and Thaier Hayajneh did one of the first security reviews of WPA3 and the SAE authentication method [11]. They found that WPA3 brought several improvements, such as protection against KRACK [5]. Furthermore, they highlight one of the most significant improvements that the SAE handshake brings; namely, forward secrecy, which provides protection against offline dictionary attacks. However, they also mentioned that WPA3 is not a silver bullet as it does not solve all issues with WPA2.

A year later Mathy Vanhoef and Eyal Ronen published the Dragonblood paper [10], showing that the many options that are available for the SAE handshake within WPA3 can lead to novel side-channel attacks. While usually hard to implement, these new attacks show that WPA3 contains flaws before widespread adoption. Furthermore, they show that it is possible to perform offline dictionary attacks on WPA3 enabled APs in the right circumstances. An interesting aspect of this paper is that it shows that the implementation between vendors varies widely. The Dragonblood paper revealed that while bringing multiple improvements, WPA3 needs extra work to be considered a secure replacement for WPA2. Among others, the Dragonblood paper helped the Wi-Fi Alliance to come up with security recommendations for WPA3 implementations [12]. These recommendations are crucial for a secure rollout of any WPA3 network and should be included in the specification itself.

The main problem with WPA3-TM is that without a proper implementation, it is trivial to downgrade a station to WPA2 [10, 11]. An attacker can spoof the known WPA3-TM Service Set Identifier (SSID) and only advertise WPA2-PSK. The station, while performing the four-way handshake, can detect this downgrade. However, at this point, the client has already sent too much information. The attacker can then perform an offline dictionary attack on the WPA Pre-Shared Key (WPA-PSK) [10]. There are multiple ways to solve this particular problem. As already recommended by the Dragonblood paper, stations can remember that they have previously connected to an SSID with WPA3 and refuse to connect to the same SSID with any older WPA implementations.

The Wi-Fi Alliance suggests separating WPA3 and WPA2 networks so that the WPA3 network can run in WPA3 only mode [12]. While this protects the WPA3 network from WPA2 vulnerabilities, it affects usability, as users would have to connect to a different network based on their devices.

III. Methodology

To further identify the problems with current WPA3 implementations, we conduct four experiments. These experiments are performed on two access points, referred to as vendor A and B. These access points were provided with specific support for WPA3 from both vendors, dating from June 2020. Both the APs are still in the development phase. Within the experiments, the access point's Authentication Key Management (AKM) suite is either set to WPA2, WPA3 or WPA3 with WPA2 (WPA3-TM) personal mode. WPA2 is defined as WPA2-PSK in Counter Mode Cipher Block Chaining Message Authentication Code Protocol (CCMP) mode with 802.11w PMF turned off. WPA3 refers to WPA3-SAE only mode with PMF turned on as mandated by the specification [7]. WPA3-TM accepts both WPA2-PSK and WPA3-SAE with PMF set to optional. Other configuration parameters such as channel, band and power are set equally between the two APs. All experiments are conducted on the 2.4GHz band. Android, iOS, macOS, Windows and Linux devices are used as stations with modern software versions. Appendix A contains a full list of all hardware and software versions used.

A. Stations auto-connect behaviour

Firstly, the behaviour of the station and how it auto-connects to Wi-Fi is examined. An auto-connect is defined as the station automatically authenticating with a previously saved SSID, without the need for user input. The Dragonblood paper recommends stations to remember if an SSID supports WPA3 and refuse to connect to the SSID with WPA2 to protect against downgrade attacks [10]. The stations and their implementations are analysed by setting up an AP with its AKM suite set to WPA3-TM. A station, without prior

knowledge of the SSID, is connected. After a successful connection, the AP configuration is altered to use WPA2 only. The auto-connect behaviour of the station is observed and captured. The packet capture is analysed to determine the exact authentication method used. The same experiment is repeated for every station and alternating the AKM. The alterations in AKM are WPA2 to WPA3-TM, WPA2 to WPA3, WPA3-TM to WPA2 and WPA3 to WPA2.

B. Station BSS selection

If the client does not select the Basic Set Service ID (BSSID) with the best AKM, in a scenario where it sees multiple BSSIDs with the same SSID but different AKMs, it will leave room for a downgrade attack. The following experiment is conducted to analyse the BSS selection algorithm behaviour for every station. An AP is set up with two BSSIDs with identical settings, including the same SSID. The first BSS is set up with WPA2 while the second is set up for WPA3. The station is then commanded to connect to the SSID, and the chosen BSSID is observed. This is repeated for every station.

C. Downgrade attack

As the Dragonblood paper revealed, a downgrade from WPA3 to WPA2 can be achieved by creating a WPA2 BSS with the same (B)SSID [10]. To confirm that this attack is still possible, we recreate a similar scenario where a station is already connected to an AP, and an attacker tries to actively disconnect the station and try to force it to connect to an evil twin AP advertising WPA2 only.

To establish a baseline, we initially connect the station under test to the legitimate AP with WPA2 and try to deauth the station using Aircrack-ng. If the deauth succeeds, we reset the network settings on the station, and the AP is upgraded to WPA3-TM. The station is then reconnected to the AP.

Next, the attacker creates an evil twin AP advertising WPA2-PSK with the same (B)SSID as the legitimate AP. The attacker then uses a different Wi-Fi interface to flood the station with deauth frames. The attack is successful if the station disconnects from the legitimate AP and connects to the evil twin AP using WPA2. The station does not have to fully connect with the evil twin. For this attack scenario, it is enough for the station only to send the first authenticated Extensible Authentication Protocol over LAN (EAPoL) frame of the 802.11i handshake [9, 13, 14]. The attacker can passively sniff the evil twin WPA2 handshake and crack the PSK using hash cracking tools like Hashcat. This attack is visualised in Figure 1. The same experiment is conducted with the legitimate AP running in WPA3 only mode. This is repeated for every station.

D. Access point denial of service

If an AP is susceptible to Denial of Service (DoS) attacks, an attacker could make the AP drop the existing connection or prevent new stations from connecting. This could increase the chance of stations attempting to connect to an evil twin. As mentioned in the Dragonblood paper [10] there are ways to overload the AP using the SAE handshake. One of the most prevalent is flooding the AP using forged SAE handshakes. Because the SAE handshakes are computationally expensive, effective forging of the handshake can cause an AP to overload. The tools that were published from the Dragonblood paper can be utilised to achieve this [15]. This tool will be used on both APs to see the effects on the network. During this DoS attack, we will measure the average throughput of a connected station and check if other stations are still able to communicate with the AP. We then compare the results to a baseline that was recorded beforehand.

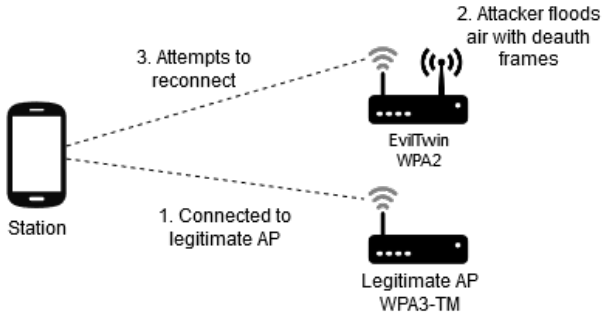

Fig. 1. Downgrade a WPA3 transition mode station using brute force deauthentication attack.

The following method is used to create the measurements; we start by connecting the Android 10 station to the AP under test. A TCP connection using iPerf2 is run for 60 seconds. The average throughput is recorded every second. This is repeated five times for a total of 300 measurements.

IV. Results

The results are categorised into sections for each experiment described in the methodology. In section IV-E, we will discuss the theoretical solutions for the downgrade attacks.

A. Station auto-connect behaviour

The results for this experiment are comparable for both APs and thus focuses on the stations. One observation that we made is that all stations refused to auto-connect to an AP that has downgraded its authentication method from WPA3 only.

When it comes to WPA3-TM, there are differences in implementation between different vendors. Android 10 stores the authentication method used on the first connection and refuses to connect using any older authentication method. However, if the authentication method on the AP is upgraded, Android will auto-connect to the upgraded WPA3-TM SSID. This upgraded authentication method is **not** stored on the phone. Which means that the phone will auto-connect to a WPA2 network with the same SSID, even if it has been previously upgraded to WPA3-TM.

Similar behaviour can be observed with iOS. Where iOS differs from Android is in the auto-connection when downgrading back to WPA2. After initially using WPA3-TM, iOS will auto-connect to a WPA2 only network. If WPA3 only mode was used, iOS would not auto-connect. Although, the

WPA2 connection can still be forced by user selection of the SSID. Furthermore, the user cannot see that a lower AKM is being used. This is because iOS does not show which authentication mechanism is used or what AKM is stored on the phone.

macOS is similar in behaviour to iOS, though, on macOS, we can see the AKM that is stored on the station. When using WPA2, macOS will already store this as *WPA2/WPA3 personal*. This can also be seen in our experiments. macOS will auto-connect to a known WPA2 or WPA3-TM SSID. It will not auto-connect to a WPA3 only SSID. A user has to manually select the SSID again in order to upgrade to WPA3 only. If the user does this, the stored AKM is then updated to *WPA3 personal*. If the network is downgraded to WPA2 again, macOS will not auto-connect. Furthermore, when the user manually clicks on the WPA2 SSID, macOS will display a clear warning message that this network was previously joined using WPA3, and the user has to confirm the connection. If the user accepts this risk and confirms that warning message, the stored AKM is downgraded to *WPA2/WPA3 personal*.

Windows is more conservative, as it stores the AKM used and will not auto-connect to the network using a different AKM. Windows will refuse to connect using any other authentication mechanism than the one stored. A disadvantage of this approach is that if the station initially connected using WPA2 and the AP is upgraded to support WPA3-TM, Windows will still auto-connect using WPA2, even when it supports SAE. A user has to manually forget the network and connect again in order to upgrade to WPA3. The advantage of this approach is that the station cannot be downgraded to WPA2 if WPA3 is used. In essence, we can say that Windows makes no distinction between WPA3-TM and WPA3 only.

The Linux machine using NetworkManager behaves similarly to Windows as it remembers the first authentication method used to connect to the network. It refuses to auto-connect to that network with another AKM. Like Windows, NetworkManager will use WPA2 to connect to a known network that has been upgraded to WPA3-TM. Downgrading was not possible if the initial connection was made using WPA3. The summarised results of all stations can be seen in Table I.

TABLE I
STATION AUTO-CONNECT BEHAVIOUR ON AKM CHANGE BETWEEN WPA2, WPA3-TM AND WPA3.

Device	**2>3-tm**	**2>3-only**	**3-tm>2**	**3-only>2**
Android 10 (S10)	yes	no	partial	no
iOS 13.5.1	yes	no	yes	no
macOS 10.15.5	yes	no	yes	no
Windows 10 v2004	wpa2	no	no	no
NetworkManager 1.22.14	wpa2	no	no	no

B. Station BSS selection

The results for the BSS selection algorithm are as follows; Windows and macOS selected the BSS with the highest AKM. While iOS and NetworkManager randomly selected one of the BSSIDs. After making a selection on NetworkManager, a second entry appeared in the list of available Wi-Fi networks for the other BSSID. Android was not affected as it displayed each BSSID as a separate selection. Only macOS showed the AKM of the selected SSID. For the other stations, we were unable to distinguish between the two BSS, based on the information provided by the user interface.

C. Downgrade attacks

The downgrade attacks described in section III are conducted on all stations. We will discuss the results for each station individually.

It was possible to downgrade iOS from WPA3-TM to WPA2 using the brute force attack from Figure 1. The connection to the legitimate AP was dropped due to timeout issues. Subsequently, iOS started connecting to our evil twin AP and dropped the connection after the WPA2 handshake failed. This was enough for the attacker to brute force the Wi-Fi password [13]. When the iPhone connected to a WPA3 only SSID, the deauth brute force would succeed in disconnecting the iPhone from the network, but, the iPhone would refuse to connect to the evil twin AP with WPA2.

Android 10 was significantly harder to downgrade than the iPhone. First off, we were only able to forcibly disconnect the Samsung S10 in combination with AP vendor B. Even with AP vendor B, the S10 did not disconnect after initiating the deauth frame DoS. The DoS would need to take place for about 10 minutes before the S10 started losing the connection to the AP. When the connection was eventually lost, a few things could happen. If a handshake was attempted with the evil twin, the attack is successful. However, a race condition between the legitimate AP and the evil twin could also occur because both devices would send out comparable beacon and probe frames, and the association requests with both APs could collide. If one of these collisions took place before Android initiated the four-way handshake with the evil twin, it would receive an error message from either of the APs indicating that information was sent incorrectly or was not understood. This causes Android to temporarily disable the network and not attempt any further connections for a set time. This was one hour in our results. Furthermore, Android would put the network into a permanently disabled state when the incorrect password error message was received from the evil twin. In this state, Android would not attempt to connect until the user manually selects the SSID again, and entered the password. At this point, the evil twin does have enough information to brute force the WPA2 PSK. One important advantage Android had was already shown in the auto-connect experiment. If the initial connection to the network was made using WPA3, Android would refuse to downgrade to WPA2. So, this downgrade was only possible when the network had been upgraded to WPA3-TM from WPA2.

The observed behaviour of macOS was considerably different from the phones. The baseline attack was possible. However, during the WPA3-TM downgrade, we were unable

to make macOS forcibly disconnect, even with two WLAN interfaces flooding deauth frames into the air. Nevertheless, we were able to downgrade the MacBook from WPA3-TM in an alternative manner. Every time the laptop went into power-saving mode (screen off), the Wi-Fi would be disconnected. When the user interacted with the laptop again, the Wi-Fi reassociated. In this brief window, the evil twin could downgrade macOS to WPA2.

The baseline test was successful on Windows. However, we were not able to perform a downgrade. This is due to the fact the Windows will not auto-connect to different AKMs as discussed in Section IV-A. It was possible to disconnect the Windows machine from the WPA3 network using the deauth DoS.

The Linux machine with NetworkManager had similar results to Windows when it comes to the downgrade attack. One advantage that the Linux machine had is that it stayed connected, even when under DoS from two WLAN interfaces. The summarised results of all stations can be seen in Table II.

TABLE II
DOWNGRADE USING EVIL TWIN BRUTE FORCE DEAUTH ATTACK, USING WPA2 ONLY DEAUTH AS A BASELINE.

Software	2 deauth	3-tm>2	3-only>2
Android 10 (S10)	yes	partial	no
iOS 13.5.1	yes	yes	no
macOS 10.15.5	yes	partial	no
Windows 10 v2004	yes	no	no
NetworkManager 1.22.14	yes	no	no

D. Access point denial of service

The Dragondrain tool was used to measure if the AP could handle a high number of forged SAE handshakes per second [15]. Up to 200 forged SAE commit frames per second would be sent. Neither of the APs ever returned an equal number of SAE handshakes to the number of commit frames we sent. This is mostly due to anti-clogging measures taken by the AP. In the 802.11-2016 standard, this is also known as the *dot11RSNASAEAntiCloggingThreshold* [9]. The Dragondrain tool has some features to evade these anti-clogging measures, like using a single client MAC address.

AP vendor A had difficulties dealing with the forged SAE handshakes. Communication from the AP to the station was impaired as the session with the station regularly dropped. This could lead to application timeouts. Of the $5 * 60s$ measurements with a single station connected, we saw a 99% average decrease in downlink throughput over the 300 total measurements. Meanwhile, on AP vendor B, no session drops were detected, and the downlink throughput with the same measurement setup only decreased by 35%. These results are presented in Figure 2.

E. Wi-Fi trust model

At the moment of writing, Personal Wi-Fi trust is built up from the station towards the AP using a password or by a third party authentication service. This poses the problem that

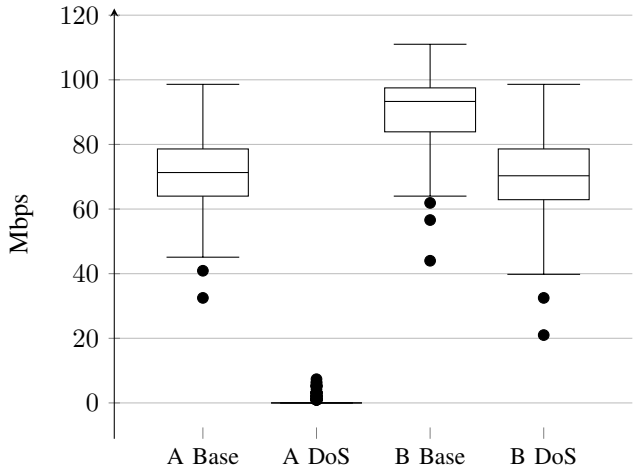

Fig. 2. The downlink throughput in Mbps from AP Vendor A and B to a Android 10 device while under DoS compared to the baseline.

it is non-trivial for a station to determine if the AP is genuine or not. The Wi-Fi alliance also realised this and is trying to develop a new addition to the WPA3 specification in the form of SAE Public Key (SAE-PK) [16].

SAE-PK can protect WPA3 SAE-PK enabled clients against evil twin attacks. It does this by implementing a way for the station to authenticate the AP by generating a static public and private key pair. The Wi-Fi password becomes a cryptographically generated hash encoded with base32 of the SSID, public key and a modifier value [17]. This password serves as a fingerprint of the APs public key that can authenticate the AP to the station. The password can be shared in various manners. The Wi-Fi Alliance suggests the use of the Wi-Fi Uniform Resource Identifier (URI) scheme [16]. This way, all information, including SAE-PK support, can be encoded into a QR code, which can easily be distributed to stations. Such a Wi-Fi URI can also contain the full AP public key, decreasing the total reliance on the password for AP authentication.

Once the password and possibly the public key are distributed to the station, it can connect using SAE-PK. Support is indicated in the 802.11 beacon and probe frames by setting the SAE-PK flag in the Robust Security Network Extension Element (RSNXE) information. For both the station and the AP, the SAE_PK status code (127) should be included in the SAE commit messages to indicate an SAE-PK authentication. A station can automatically enable SAE-PK for a network based on the password format. If both parties indicate support, the AP will send an SAE-PK element in the SAE confirm message. This element contains a SIG, the public key and the encrypted modifier. The modifier is wrapped using the encryption key derived from the SAE handshake. The SIG consists of a signature of the following information; the encoded SAE commit messages, the modifier, public key, BSSID and the station MAC. Once the station receives this element, it unwraps the modifier, verifies the public key using the fingerprint password and verifies the signature using the

public key [16]. If every step is successful, the SAE handshake is complete, and the AP authenticated. Subsequently, the four-way 802.11i handshake can take place. The SAE-PK process is depicted in Figure 3.

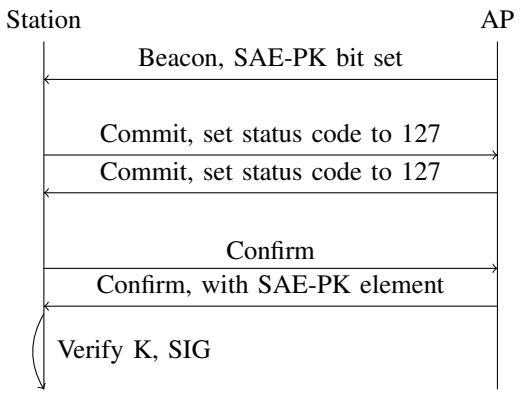

Fig. 3. SAE-PK sequence diagram using passive scanning. Where K is the AP public key and SIG the signature generated by the AP.

SAE-PK is the first standardised Wi-Fi personal authentication mechanism that supports AP to station authentication. The base32 encoding scheme of the password avoids confusion between upper- and lowercase characters [17]. The security of SAE-PK hangs on the security and implementation of the fingerprint generation algorithm. In the WPA3 specification draft, the Wi-Fi Alliance calculates that the minimal required password length of 12 characters would take an attacker around 12 years to find the modifier value. The minimal recommended length for the password is also 12, but with a more cryptographically intensive brute force search for the modifier value. The minimal recommended option will take an attacker around 3000 years to find the modifier value. The Wi-Fi Alliance has already stated that the minimum required password and search length can be increased in later revisions [16].

Another addition to the WPA3 specification draft is the introduction of the Transition Disable indication (TDi) [16]. The TDi will be made a Key Data Element (KDE) of EAPoL key messages in the 802.11i four-way handshake [9]. Its purpose is to indicate to a station that certain AKMs should not be used when connecting to this network. The bitmap inside the TDi KDE signals to the station which AKMs it should use and which it should disable. If an authenticated TDi is received from the AP, a station will disable the AKMs as per the bitmap if the station supports any of the corresponding AKMs that should not be disabled.

Furthermore, if a TDi is received by the station, it should also disable the use of WEP and TKIP for that network. This allows APs to signal to the stations that they should only use secure mechanisms to connect the APs corresponding network and that stations should not connect using any older authentication suites than advertised in the TDi. Note that the station can only store this TDi after the first connection to that network was made. Thus a combination with SAE-PK should be made in order to prevent trust on first use.

An alternative way of achieving mutual authentication between the station and AP is to use verifiable SSIDs. At the moment, anyone can use any arbitrary SSID anywhere. There are no restrictions on the use of SSID names, apart from the conventions defined in the standard [9]. We suggest a new way of achieving mutual trust between the station and AP by combining two proven technologies. Namely, the current authentication mechanisms for clients with the use of domain names for SSIDs. If an SSID consisted of a Fully Qualified Domain Name (FQDN), this SSID would become unique enough to be used as an identity. This would give several options for mutual authentication. One way of achieving AP to station authentication would be to include trusted certificates on the AP corresponding to the SSID FQDN. These certificates should be signed by a trusted Certificate Authority (CA). A station can verify the validity of an AP certificate offline by using the CA certificate store. This could be implemented as an extension of the current WPA3 specification. The certificate would be sent and verified before the four-way 802.11i handshake [9]. This way, the station could verify that the party owns the AP and corresponding domain. For the certificate sending process, small parts of the Extensible Authentication Protocol with Transport Layer Security (EAP-TLS) process can be utilised to ensure minimal changes are needed to the station and AP logic [18].

The authentication with verifiable SSIDs gives several advantages. First off, Wi-Fi passwords can remain unchanged. Furthermore, if trusted certificates are used, these certificates must adhere to the expiration deadline that is enforced by the CA. Ensuring that the certificates are updated at least every 825 days [19]. This gives an incentive to AP manufacturers to make sure that the process of deploying a certificate for an SSID is administrator friendly, as the certificate would need to be replaced multiple times over the lifespan of a network. New certificate authorities like Let's Encrypt can be used to automate the process of certificate signing. A disadvantage of using this method is that the SSID of existing networks would need to be changed in order to support this feature. This means the user would need to connect to a new SSID once this SSID has been upgraded to a verifiable SSID.

V. Discussion

Within the results, we see many options for improving and extending WPA3. However, the question that we should ask ourselves is, which of these options will bring the most significant improvement in terms of security as well as being user friendly. As we have seen with other Wi-Fi standards and additions, adoption takes a long time [4]. This presents the requirement that additions need to be future proof or risk being outdated before they are widely adopted.

We argue that SAE-PK will be a promising addition to the 802.11 standard and will improve the security of WPA3-TM. Mainly because it is the first personal Wi-Fi addition that enables authentication of the AP, the fact that SAE-PK requires a fixed password scheme is a downside. While the base32

encoding of the password has the benefit that users cannot confuse upper- and lowercase or any special characters, this format is still quite different from the simple Wi-Fi passwords we see today. There is a chance that this could hamper the adoption as users would have to change their current Wi-Fi setup. Nonetheless, this addition will be beneficial for medium scale Wi-Fi deployments, like restaurants and malls. They can efficiently utilise the Wi-Fi URI scheme to distribute the SAE-PK password and public key to their customers, in the form of a QR code. Our suggestion for verifiable SSIDs has a similar flaw. Namely, the need to change existing SSID into an FQDN format when updating to the new scheme.

Running multiple SSIDs with different authentication mechanisms, as suggested by the Wi-Fi Alliance, can improve security. However, the security depends on the way these multiple SSIDs are used. The passphrases for WPA-PSK and SAE should be different to increase the security for WPA3 clients. AP vendor B has a quite novel solution to this. They provide the option to set a separate passphrase only for SAE authentications. This means it is no longer necessary to broadcast multiple SSIDs for AKM setups. WPA2 only clients would still use the WPA passphrase while SAE capable clients use the SAE passphrase to authenticate. One big disadvantage of two passphrases on one SSID is that it is user-unfriendly. A user would have to know the capabilities of the station he is using in order to fill in the right password. This can lead to confusion.

Furthermore, even then it is still possible for an attacker to crack the WPA-PSK and join the network using WPA2 authentication. Separation of the two networks (WPA2 and WPA3) is one of the few ways we can be sure that a WPA3 network can operate securely at this moment. This is probably the main reason that the Wi-Fi alliance recommended network separation in their security considerations for WPA3 [12].

User experience is also a factor when splitting SSIDs. A user either needs to be aware of his station security capabilities or the network must be set up in such a way that the users only see the correct SSID for their station. For WPA2 this is possible, as stations can ignore the SSID with authentication mechanisms the station does not support. However, for WPA3, this is more difficult, as the station would need to be made aware that it should connect the WPA3 SSID. At the moment, this can only be done by instructing the user to select the correct SSID. Furthermore, if the network is logically or physically separated, users might experience usability problems. For example, if a user in the WPA3 network wishes to connect to a station in the WPA2 network, this might not be possible in the standard setup. Some routing would need to be implemented on the AP to let the two networks talk to each other. If the networks would have a shared LAN, this could again impact security.

VI. Conclusion

The results show that multiple stations are still able to be downgraded on WPA3-TM networks, even the high-end Android 10 and iOS devices with WPA3 support. The downgrades in our experiments were all achieved using ten year old consumer hardware with no special or proprietary software and low inherent difficulty to perform.

Both the Apple iOS and macOS stations would auto-connect to a WPA2 only SSID, even when they initially connected using SAE on a WPA3-TM enabled AP, and thereby exposing the necessary information to crack the PSK. Because Windows and NetworkManager only establish a connection using the stored AKM, they are not affected if the previous connection was made using WPA3. We advise iOS and macOS to avoid downgrading to WPA2-PSK if SAE was previously available on that SSID.

Furthermore, we recommend displaying the AKM of a given SSID. This can be done in the same fashion as macOS, i.e. while prompting for the Wi-Fi password. We believe it is crucial that users have the ability to verify the network they are connecting to. Moreover, we advise NetworkManager and Windows to supply users the possibility to upgrade to SAE when made available on a known network. In the current situation, the connection would remain WPA2-PSK even if the network has been upgraded to WPA3-TM. Such an upgrade option can be presented by asking the user if they wish to upgrade when the AP advertises SAE.

Since the station initialises the connection to the AP, most of our recommendations are geared towards the stations. At the time of writing, the AP does not have a standardised method of influencing the station's decisions. Combined with the fact that currently there is no mutual authentication in personal mode. This leaves the station susceptible to downgrade attacks. Section IV-E illustrates that SAE-PK allows stations to authenticate the AP. We conclude that this greatly increases the mutual trust between the station and the AP. The fact that stations can determine if a network is using SAE-PK based on the password format is an additional benefit, as this removes trust on first use. SAE-PK can be made more effective in combination with TDi. As the results expose, different vendors have different approaches when it comes to remembering the authentication mechanism used by a specific network. TDi provides the AP operator with more control to inform the station which authentication mechanisms should be used. This standardisation can unify most of the Wi-Fi authentication implementation differences we have observed. SAE-PK, together with TDi, gives the AP operator the means to secure a WPA3 transition roll out. We recommend that SAE-PK is made mandatory for WPA3 personal transition mode certifications.

VII. Future Work

The WPA3 specification draft suggests replacing the Wi-Fi network password for a generated password [16]. The effect of such a change could prove beneficial for the security of the network. Investigating these effects is of interest. Furthermore, the generation of the password is an interesting topic for future work as a base32 encoding of the public key fingerprint is used as the password. While this provides an out of bound delivery

mechanism to bootstrap the trust, we wonder how random this new password mechanism is and if the entropy is high enough to combat attacks, some of which are already discussed by the Wi-Fi Alliance [16].

Furthermore, making the AP identifiable for authentication purposes is another way to uniquely identify the network. We wonder what the privacy implications are. We would value a discussion about the feasibility of verifiable SSIDs within the security community. A proof of concept for this method should be made in order to prove its usefulness.

References

[1] Wi-Fi Alliance. *Wi-Fi Alliance® introduces Wi-Fi CERTIFIED WPA3™ security*. URL: https://www.wi-fi.org/news-events/newsroom/wi-fi-alliance-introduces-wi-fi-certified-wpa3-security (visited on 01/05/2020).

[2] Wi-Fi Alliance. *Security*. URL: https://www.wi-fi.org/discover-wi-fi/security (visited on 01/05/2020).

[3] Wi-Fi Alliance. *Wi-Fi CERTIFIED™ Certification Overview*. URL: https://www.wi-fi.org/file/certification-overview (visited on 05/06/2020).

[4] WiGLE LLC. *Statistics*. URL: https://wigle.net/stats (visited on 01/05/2020).

[5] Mathy Vanhoef and Frank Piessens. "Key reinstallation attacks: Forcing nonce reuse in WPA2". In: *Proceedings of the 24th ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Dallas, TX, USA: ACM, Oct. 2017, pp. 1313–1328. ISBN: 978-1-4503-4946-8.

[6] Stephen Orr. "Advancements in Wireless Security". In: *Cisco Live* BRKEWN 2006 (Jan. 2020). Cisco Library.

[7] Wi-Fi Alliance. *WPA3™ Specification*. URL: https://www.wi-fi.org/file/wpa3-specification (visited on 04/06/2020).

[8] D. Harkins. *Dragonfly Key Exchange*. RFC 7664. IETF, Nov. 2015. URL: http://tools.ietf.org/rfc/rfc7664.txt.

[9] IEEE. "IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications". In: *IEEE Std 802.11-2016* (Dec. 2016). DOI: 10.1109/IEEESTD.2016.7786995.

[10] Mathy Vanhoef and Eyal Ronen. "Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd". In: *Proceedings of the 41th IEEE Symposium on Security & Privacy (S&P)*. San Francisco, CA, USA: IEEE, May 2020, pp. 808–824. ISBN: 978-1-7281-3497-0.

[11] Christopher P Kohlios and Thaier Hayajneh. "A comprehensive attack flow model and security analysis for Wi-Fi and WPA3". In: *Electronics* 7.11 (2018). DOI: 10.3390/electronics7110284.

[12] Wi-Fi Alliance. *WPA3™ Security Considerations*. URL: https://www.wi-fi.org/file/wpa3-security-considerations (visited on 02/06/2020).

[13] Robert Moskowitz. *Weakness in Passphrase Choice in WPA Interface*. 2003. URL: https://wifinetnews.com/archives/2003/11/weakness_in_passphrase_choice_in_wpa_interface.html (visited on 19/06/2020).

[14] IEEE. "IEEE Standard for Local and Metropolitan Area Networks–Port-Based Network Access Control". In: *IEEE Std 802.1X-2020* (Feb. 2020). DOI: 10.1109/IEEESTD.2020.9018454.

[15] Mathy Vanhoef et al. *dragondrain-and-time*. URL: https://github.com/vanhoefm/dragondrain-and-time (visited on 16/06/2020).

[16] Wi-Fi Alliance. *WPA3™ Specification Addendum for WPA3 R3*. URL: https://www.wi-fi.org/file/wpa3-specification-addendum-draft (visited on 02/06/2020).

[17] S. Josefsson. *The Base16, Base32, and Base64 Data Encodings*. RFC 4648. IETF, Oct. 2006. URL: http://tools.ietf.org/rfc/rfc4648.txt.

[18] D. Simon, B. Aboba and R. Hurst. *The EAP-TLS Authentication Protocol*. RFC 5216. IETF, Mar. 2008. URL: http://tools.ietf.org/rfc/rfc5216.txt.

[19] CA/Browser Forum. "Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates". In: 1.7.0 (May 2020), pp. 39–43. URL: https://cabforum.org/baseline-requirements-documents/.

Appendix A
Hardware and software versions

The hardware used for the stations and the corresponding operating system versions (OS) are listed in Table III. The APs hardware and their software versions used are listed in Table IV. The driver versions, where applicable, can be found in Table V. Lastly, Table VI contains the additional software we used.

TABLE III
HARDWARE AND OS VERSIONS FOR STATIONS

Hardware	Version
Apple iPhone X	iOS 13.5.1
Samsung Galaxy S10	Android 10 May 2020
Apple MacBook Air (2017)	macOS 10.15.5
Apple MacBook Pro (2018)	macOS 10.15.5
Sony Vaio STV131A11M	Fedora 32 & NetworkManager 1.22.14
Dell Optiplex 7050	Windows 10 v2004

TABLE IV
HARDWARE AND SOFTWARE VERSIONS FOR APS

Hardware	Version
AP vendor A	Hostapd v2.9
AP vendor B	Hostapd v2.10-dev
HP Pavilion dv6 (evil twin)	Kali Linux 2020.2

TABLE V
HARDWARE AND DRIVER VERSIONS

Hardware	Chipset	Driver version
Sony Vaio STV131A11M	AR9485	ath9k 5.6.19-300.fc32
Dell Optiplex 7050	AX200	Intel 21.90.3.2
HP Pavilion dv6 (wlan0)	5100 AGN	iwlwifi 5.6.0-kali2
HP Pavilion dv6 (wlan1)	AWVS036NH	rt2800usb 2.3.0

TABLE VI
ADDITIONAL SOFTWARE AND VERSIONS

Software	Version
Aircrack-ng	1.6
Dragondrain	82616a7 [15]
Hostapd (evil twin)	2.9
iPerf	2.0.5