Wireless Encryption and WPA2 Weaknesses

Kyle Moissinac, David Ramos, Giovanna Rendon, Abdelrahman Elleithy

Department of Computer Science
William Paterson University Wayne
New Jersey, United States
moissinack@student.wpunj.edu , amosd9@student.wpunj.edu , rendonrodriguezg@student.wpunj.edu , elleithya@wpunj.edu

2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC) | 978-1-6654-1490-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/CCWC51732.2021.9376023

***Abstract*—Encryption is prevalent in many forms of modern communications. Still, wireless communications security holds higher priority since any device can easily intercept the data in its range. This paper will focus specifically on IEEE 802.11 wireless communication and the evolution of its encryption protocols from WEP to the current WPA3 to identify current weaknesses and methods for improvement. There are mechanisms to offer better protection for wireless network communications, such as RADIUS servers, certificates, and new exchange approaches from WPA3. However, some of these solutions not only entail resource overhead, but vulnerabilities have also been discovered in the WPA2 and even in the newest WPA3 protocols. Therefore, this paper includes the cryptography algorithms selected and the overhead required to implement them, such as computational resources, certificates, and external server requirements.**

Keywords—WiFi, wireless communication security, WEP, WAP, WPA2, WPA2 flaws, WPA3, encryption, network security, wireless vulnerabilities

I. Introduction

Since 1997, consumers worldwide have been enjoying WiFi through a multitude of devices, ranging from laptops to cell phones, video games consoles, and even refrigerators. As a result of the widespread dissemination of these communications, the need to protect these connections has become critical. To achieve this protection, the security of communications through WiFi has transitioned from open WiFi networks to Wired Equivalent Privacy (WEP), to WiFi Protected Access (WPA), to WiFi Protected Access 2 (WPA2), and finally to WiFi Protected Access 3 (WPA3) [1]. Even though much of this data is sent unencrypted in public networks such as schools, libraries, and hotels, WiFi networks rely on encryption protocols to ensure data and communication security. These protocols have evolved and differ in encryption methods, security offered, overhead, and weaknesses.

First, the security protocols in WiFi networks are based on cryptographic algorithms. Cryptography is based on complex mathematical problems such as the integer factorization problem and the elliptic curve cryptography problem. The purpose of cryptography is to provide confidentiality, integrity, authentication, non-repudiation, and obfuscation to data in-use, in-transit, and at-rest [2]. However, cryptography faces challenges in regards to resources such as low-power devices and fast-response-time applications. Furthermore, there are three categories of cryptographic algorithms: hash, symmetric, and asymmetric, which can be stream or block ciphers depending on the quantity of data that they handle at a time [2]. Symmetric algorithms make use of a shared key, while asymmetric algorithms use a private and public key [2]. The first protocol used in WiFi used asymmetric algorithm, and some standard algorithms used in wireless encryption protocols are Rivest Cipher 4 (RC4) and Advanced Encryption Standard (AES) [1].

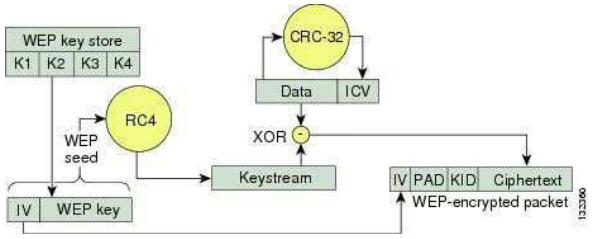

Fig. 1. WEP encryption process

The first protocol used in WiFi networks, WEP, was released along with the original 802.11 standards in 1997. WEP is based on the symmetric 64-bit to 128-bit stream cipher RC4 for encryption and the 32-bit cyclic redundancy check (CRC-32) checksum for integrity [3] as seen in [4, Figure 1]. Even though an initialization vector (IV) is added as an additional security to this encryption, WEP is no longer considered a secure protocol. The IV is only 24-bits; thus, is usually repeated, which results in detectable patterns that can be identified in collisions, and packets can then predicted using statistical analysis [3]. WEP protocol makes wireless networks vulnerable to threat actors through rogue access points, wireless replay attacks, and data interception [2]. Therefore, new protocols were developed to address these security flaws.

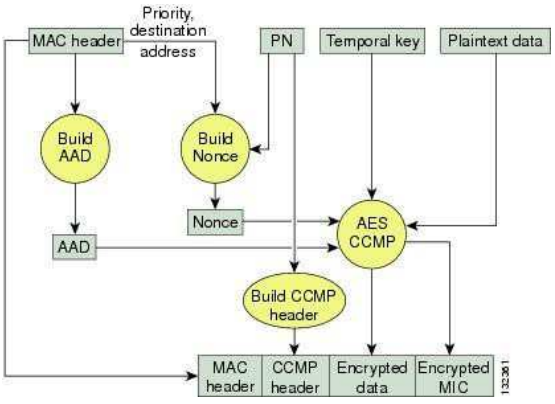

Fig. 2. WPA2 encryption process [4]

After WEP, WPA and, subsequently, WPA2 were introduced with the IEEE 802.11i standard. WPA is a security protocol introduced in 2003, which can utilize either RC4 or

978-0-7381-4394-1/21/$31.00 ©2021 IEEE

AES, and it is also based on the Temporal Key Integrity Protocol (TKIP) [2]. This protocol addresses encryption and authentication, and it has two modes, which are personal and enterprise. The main difference between WEP and WPA was the addition of a session key. However, it has also been proven to have major flaws and be vulnerable to brute force attacks and rainbow tables containing the most common SSIDs.

WPA was superseded by WPA2 in 2004. As illustrated in [4, Figure 2], WPA2 is based on Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP) that provides authentication but increases the overhead [5]. WPA2 uses the symmetric block cipher AES with a key size of 128-bits and an IV of 48-bits. WPA2 is also available in two modes, similar to WPA, which are the personal mode that uses a Pre-Shared Key (PSK) as seen in [6, Figure 3] and the enterprise variant [2]. In the personal mode, the PSK uses an 8 to 63-character shared key, and it is common in home routers/access points [2]. On the other hand, the enterprise variant uses N 802.1x/RADIUS server for authentication instead of PSK as illustrated in [4, Figure 4], and it is common in large enterprise networks [7]. Nonetheless, in 2016, a flaw in WPA2 was discovered, which allows threat actors to use key reinstallation attacks (KRACKs). These attacks expose confidential information and allow the injection and manipulation of data [8-10].

Finally, WPA3 was released in 2018, and it uses Simultaneous Authentication of Equals (SAE) to replace WPA2's PSK exchange protocol. WPA3 also adds Opportunistic Wireless Encryption (OWE) defined in the Internet Engineering Task Force (IETF) RFC8110 to open networks [10]. WPA3 also has personal and enterprise modes. Some of the enhancements on WPA3 include protection for personal mode and preventing legacy protocols [10]. Another protection offered by WPA3 is resilient to offline dictionary attacks through its Dragonfly handshake [10]. However, the following year after the release of WPA3, a group of vulnerabilities were discovered. These vulnerabilities are called Dragonblood, and they exploit the Dragonfly handshake, which is used in both personal and enterprise modes. With these vulnerabilities, threat actors can steal passwords or impersonate users [11, 12].

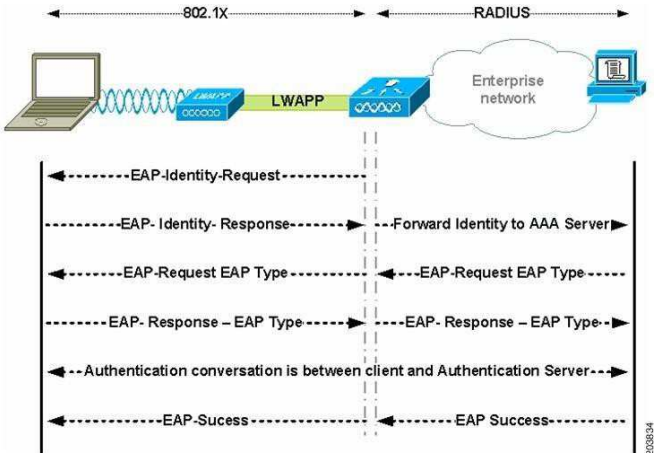

Fig. 3. WPA2 enterprise - EAP protocol flow [6]

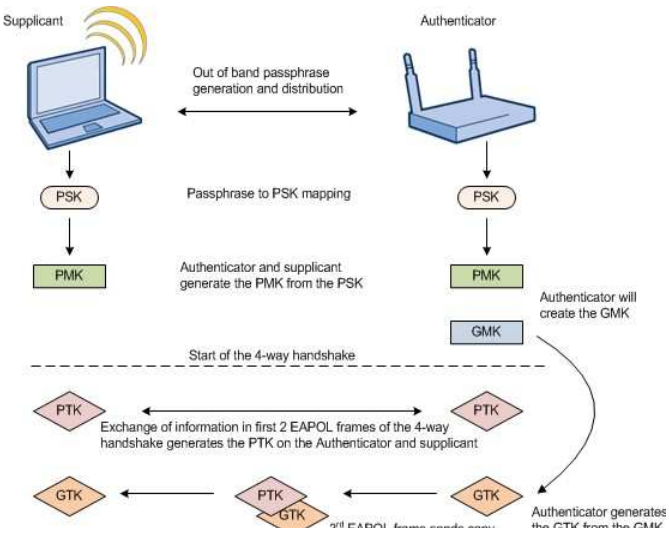

Fig. 4. WPA2 encryption key generation [4]

II. Problem Identification

All of the aforementioned vulnerabilities for each of the security protocols introduced since 1997 prove that the complex task of securing wireless communications is an issue that still remains even after the release of the latest security protocol, WPA3. For example, WPA2-PSK has an inherent weakness due to its shared key, and it has no method of preventing a Wireless Replay Attack [9]. While WPA2-Enterprise addresses both of those issues, it also requires extra RADIUS servers and certificates to accomplish this [7]. Furthermore, both variants of WPA2 and WPA3 have had published vulnerabilities related to the symmetric key exchanges required by the protocols, and they were compromised by KRACK and Dragonblood, respectively [8, 9, 12]. Moreover, WPA2 and WPA3 and their corresponding increase in compute resource overhead introduce additional challenges for low-power and low-end IoT devices [5, 13]. Finally, the continually increasing availability of raw computational power that can be used to crack any encryption attempts has also forced increased overhead for the encryption protocols [5]. Therefore, wireless traffic is easily captured. Like any cryptographic scheme, attacks can be overcome by attacks very quickly if the scheme is compromised or given enough time and/or resources.

While WPA2-Enterprise is more robust, WPA2-PSK compromises convenience and needlessly exposes more weaknesses than WPA2-Enterprise. The increased key sizes and shorter re-key intervals afforded by WPA2-Enterprise, makes brute force decrypt attempts more expensive from a computational perspective and less attractive from a value perspective since a shorter interval means it is necessary to redo the brute force key to obtain a larger block of data [33]. However, it entails the negative effect of requiring more compute power on the client and AP side, which could be a problem on low-end consumer and small business equipment. On the other hand, the most significant vulnerability within WPA2-PSK is exposed or easily guessed pre-shared key (PSK). Since a WPA2-PSK master session key (MSK) is derived from the PSK and therefore common to all sessions if

the key is exposed, any device can easily copy and decrypt wireless traffic. If the PSK is weak, meaning easy to guess, an offline brute force can iterate through the possibilities to find the key in significantly fewer attempts than the protocol accommodates [33]. A PSK has to be a minimum of 8 characters but can be as many as 63 [34]. For example, some organizations require it to be 20 or more characters. Choosing a short PSK consisting of a word found in the dictionary such as 'password' or a simple string of numbers such as '12345678' makes it easy to remember. Still, it dramatically increases the ease with which it is guessed while using a large and complex PSK will likely exponentially increase the time required to guess it.

Furthermore, the problem with WPA2-PSK is that the temporal keys used for data encryption are derived from the easily-discovered values that appear in easily-captured over-the-air transmissions along with that original shared key. This issue is exacerbated since all wireless clients use the same pre-shared key, so identifying the key allows for the decryption of all the other user connections on that SSID [33]. Compared to WPA2- PSK, WPA2-Enterprise has a significant advantage in this area since every connection has its own key by leveraging an external RADIUS authentication server. Despite having valid credentials to connect to the network, it is not possible to use these credentials to decrypt another client's traffic. This is because instead of using a PSK for validation and access to the network and then using that same key to encrypt the traffic, the RADIUS server provides a public key certificate which is used to create a secure channel with the client. The RADIUS server securely negotiates authentication; afterward, elements of the authentication are used along with random numbers from both the client and RADIUS server as part of a unique per-session key for the actual data encryption [10].

In WPA2-Enterprise, the actual data is transferred using the same asymmetric key encryption length as PSK, but the chance of the temporal key exposure is significantly lower. In addition to the cryptographic benefits of reduced key exposure, the public key component of WPA2-Enterprise also allows the client to validate that they are connecting to the intended network. This additional security benefit can prevent man-in-the-middle (MITM) 'decoy access points' from intercepting credentials and/or decrypting traffic, similar to how web browsers use the same style of public-key certificates to validate web servers [10]. Therefore, in this paper, we will look at existing attempts and potential solutions to address the above problems related to exposure of the temporal keys used for the actual data encryption, especially for non-enterprise modes.

III. Related Work

WPA3 replaces WPA2-PSK and implements an IEEE 802.11-2016 standard called Simultaneous Authentication of Equals (SAE), reducing or eliminating offline brute force attacks on weak passwords due to its interactive nature [10, 35]. This helps mitigate the issue of users opting for weak and easily guessed passwords. WPA3 is still very new and only supported on a minimal number of devices, including only the latest releases of operating systems from late 2019. Supported client operating systems include Apple iOS 13, MacOS 10.15, Windows10 May 2019 Update [36-37]. Supported access points include Cisco AireOS 8.10, Aruba Instant 8.4.0.0 [38-40]. Additionally, while SAE adds some protection by eliminating the PSK, we found no mention of any improvements that allow WPA3-Personal to address AP authentication, which still leaves non-enterprise environments susceptible to man-in-the-middle (MITM) attacks.

Another solution focused on key exchange and aimed at reducing vulnerability to MITM attacks has been proposed in "Secure In-Ban Wireless Pairing" [41]. This proposal eliminates the pre-shared key, which has proven to be a vulnerable factor in wireless cryptography. Furthermore, the authors' solution is based on the tamper-evident pairing (TEP) protocol, in which any interference in the exchange can be detected, thus offering protection against MITM attacks [41]. TEP is compatible with legacy 802.11 and low-power devices, and it offers a simple solution for home users. Even though users are not required to have the technical knowledge to utilize this method, users must "push a button in each device" [41] as seen in [41, Figure 5]. This scheme also requires the acknowledgment and calibration of the noise floor because its nature is based on the medium's energy level. The authors evaluated their proposal and concluded that their solution is viable in the real world while likely offering protection against MITM attacks [41].

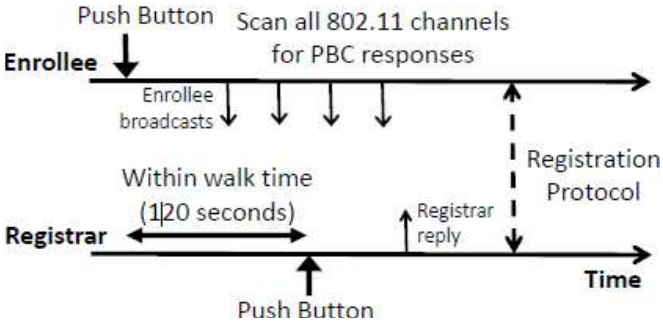

Fig. 5. Push-button configuration (PBC) between the enrollee and registrar [41]

As an additional solution, there have been multiple proposed machine learning applications in cryptography due to its capability to handle large amounts of data, solve problems, and support mutual learning. Many of these proposals take advantage of mutual understanding to securely exchange a public key in which "neural cryptography creates a channel for secure key exchange by mutual learning for synchronization of special kind of neural networks called Tree Parity Machines (TPM)" [42]. In this method, the machines can generate a secret key without communicating by using the same input that each machine uses to generate outputs until they have been synchronized [42- 43]. Optimization algorithms for TPM machines have also been proposed, such as the cuckoo search (CS) algorithm [42] as seen in [42, Figure 6]. This type of key exchange scheme by mutual learning is not vulnerable to brute force attacks due to the required computing power. However, Klimov, Mytyagin, and Shamir found, in their simulated attacks of chaotic synchronization in neural networks, that this proposal is vulnerable to geometric and probabilistic attacks [44].

IV. Proposed Solution

An incredible amount of resources have contributed toward making the current standards as secure as possible, and the few major vulnerabilities have already been or are currently being addressed. However, the advancements from WEP to WPA to WPA2 to WPA3 have shown the need to address weaknesses and the continually increasing overhead to keep communications secure compared to the ever-increasing computing power available to attack these algorithms [10]. To further aggravate the matter, many home users and small businesses rely on a relatively insecure variant of the modern protocols called WPA2-PSK simply because it is the easiest to utilize. Thereby, the purpose of this paper is to document and explore the possibilities of providing the benefits of the more advanced WPA2-Enterprise cryptography most efficiently to non-technically savvy home and small business users where WPA2-PSK is being used.

A unique approach to solving the inherent weaknesses of WPA2-PSK for most devices is to eliminate WPA2-PSK. We believe this can be accomplished in several ways, but one of them has the potential to be as convenient as WPA2-PSK while continuing to be viable as a low-cost zero-touch consumer product. In this paper, we will consider 'most efficient' as something that does not require new standards or additional hardware and where the functionality can be provided on an inexpensive platform that a non-technically savvy consumer, such as a home or small business user, can operate. While RADIUS servers supprot many configuration options and enormous flexibility for enterprises, almost none of those features are required for the two main WPA2-Enterprise cryptographic /security benefits [45]. Without having the opportunity to research all of the functionality of Enterprise environments, we are assuming that some RADIUS server features such as Active Directory or LDAP integration, as well as the ability to support pass user-specific attributes, and the ability to support client certificates, have no place in the home or small business. Furthermore, it can be safely omitted to reduce complexity and overhead without affecting the protocol's cryptography aspects.

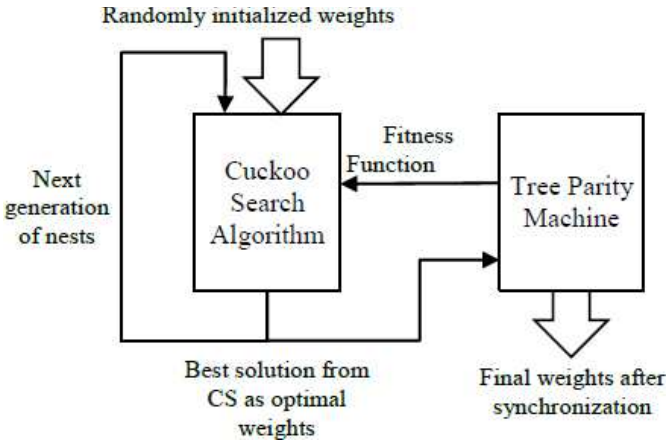

Fig. 6. Cuckoo search algorithm used in a TPM [42]

Years of innovation and refinements have contributed to relatively secure wireless communication options, but non-enterprise uses have been neglected and exposed from a security perspective unless they invest in enterprise infrastructure. Therefore, our solution is based on the hypothesis that the existing proven components of the more secure WPA2-Enterprise can be incorporated into low-end commodity devices to bring these benefits to non-Enterprise environments such as home and small businesses in the most efficient manners as illustrated in [7, Figure 7].

Our proposal acknowledges that numerous mathematicians, standards bodies, and hackers have advanced the currently accepted wireless security standards to the natural balance of security versus compute resources. It is more valuable and practical to leverage existing standards to bring the available benefits to a broader audience [10]. Consequently, we will compare readily available RADIUS server solutions, including commercial, cloud, and open-source [46-48], versus what we think is a better alternative of emulating the base RADIUS server functionality inside a consumer-grade access point. We will evaluate this solution's benefit by further researching and showing the cryptographic benefit of WPA2-PSK versus WPA2-Enterprise.

It is apparent that home and small business users already accept an easy to use solution without the advanced complexity of an Enterprise RADIUS server's full features but would certainly benefit from improved security if the solution is as simple to use. The concept of multiple user accounts and supporting client certificates for authentication are two such features that allow for flexibility but are not related to encryption. We believe that the use of per-session keys to prevent eavesdropping is the most crucial benefit. The second most important benefit is AP authentication to ensure that a connection is being established to a known network instead of an imposter.

To make a fair comparison of the benefits, we will identify the core capabilities of various RADIUS servers' applications and their requirements and cost. For the open-source or free libraries, we will attempt to dissect the key components to determine whether or not they seem to be candidates that could be integrated or emulated inside consumer-grade routers and APs, given that there are Linux-based OpenWRT/DD-WRT access points in the marketplace as well as numerous open-source Linux and WiFi HotSpot options [49-50]. Additionally, the proposal regards the RADIUS public key component as a core feature since it is used to create the secure channel for key transfer and allows for the verification of the wireless network to which the client is connecting. Since the solution is leveraging existing code and protocols, this requires no further development than merely reusing that code base to bring MITM prevention to non-enterprise environments.

V. Methods

Based on the aforementioned hypothesis, our solution is based on the assumption that existing, proven components of the more secure WPA2-Enterprise can be incorporated into low-end commodity devices. Therefore, we used existing applications to simulate the core components of WPA2-Enterprise present in

our solution. These components are WPA2-PSK key generation, a basic authentication request to a real RADIUS server, and an HTTPS handshake, which represents the additional step added in a WPA2-Enterprise handshake. The encryption of data using the symmetric key was not included since it is common to WPA2-PSK and WPA2-Enterprise.

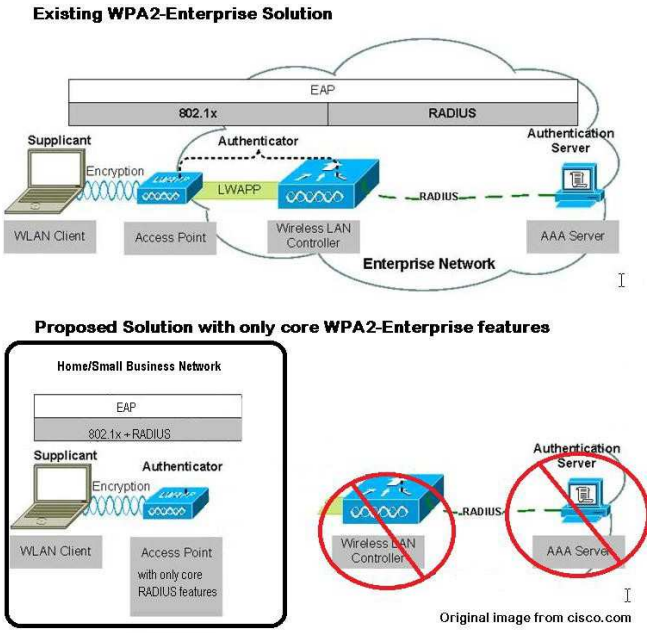

Fig. 7. WPA2-Enterprise versus Proposed solution [7]

We found that the PSK key generation logic was already in the Kali Linux security tool called Aircrack-ng. An open-source full-featured RADIUS server package called FreeRadius was also available for the same platform. Also, an HTTPS-enabled free Apache web server represented the public-key based SSL handshake used within WPA2-Enterprise. Using these applications allowed us to simulate the core components without needing actual wireless devices or dedicated equipment.

Installing the FreeRadius server package on that same Kali Linux VM required only two commands and a username to the config file. This contained a test script that sends an actual RADIUS authentication request, which we also measured using time. The syntax for the FreeRadius test scripts is: "radtest username password RADIUSServerName/IP someid RADIUSSecret"

Finally, The WPA2-Enterprise handshake requires a public key certificate exchange from the server to the client. To represent this, we added the Apache2 package to the Kali VM, then generated and installed a self-signed certificate using OpenSSL and configured Apache for HTTPS using the Apache SSL module. The Apache install required us to find and follow a guide. We used "wget," a command-line 'browser' utility, and did HTTPS request a non-existent file for a client.

VI. Mathemaical Model

To approximate the system's overhead difference between WPA2-PSK, WPA2-Enterprise, and our solution, a subset of WPA2-Enterprise functionality, we will decompose the encryption process as described in the previous sections. Therefore, we have determined three major pieces needed for this encryption process: authentication, key exchange, and the subsequent encryption. We will describe their benefits for each of these three pieces and then calculate the more advanced WPA2-Enterprise protocol's relative overhead.

The Authentication component for WPA2-PSK is trivial and is done using the shared key to decrypt a message [51]. The more advanced authentication methods in WPA2-Enterprise provided by an external RADIUS server are very diverse [52], which implies significantly more overhead. We will still cover three major areas related to this topic: username/password, client certificate, and the server certificate. The actual user authentication in WPA2-Enterprise does not improve security by itself, but it is necessary since it adds the public-key encrypted secure channel for key exchange and introduces the random seeds to generate unique keys. That difference in key generation and exchange appears to have the most important benefits over PSK. The dynamic keys are unique to every session, which prevents easy decryption of data. A public- key encrypted secure channel allows clients to authenticate the access point's identity to prevent evil twin attacks. It should be noted that the wireless clients still have to configure extra settings to only accept known certificates to benefit from this evil twin prevention [53]. As we will see in the results below, this key generation variation does not imply a significant difference in overhead for a basic use case. Finally, both variants generate a 128-bit symmetric key, which is used to encrypt the traffic. Therefore, while the key exchange is different and more secure in WPA2-Enterprise than in WPA2-PSK, there is no difference in the data-transfer part of the protocol's encryption strength.

The actual formulas for calculating the keys in a WPA2-PSK authentication are: “Function 1: PMK = PSK = PBKDF 2(passphrase, ssid, sidLength, 4096, 256) Function 2: PTK = PRF-X(PMK, “Pairwise key expansion” || min(AA, SA) || max(AA, SA) || min(ANonce, SNonce) || max(ANonce, SNonce)), where PBKDF 2 (Password-Based Key Derivation Function) is a key derivation function defined in RSA Laboratories’ Public-Key Cryptography Standards (PKCS) series. PRFX is a pseudorandom function based on a keyed-Hash Message Authentication Code using SHA-1 (HMAC-SHA1) that generates a PTK of size X bits, as defined in 802.11i section 8.5.1.1." [54]. The main issue with this is that this is and must be a pseudorandom function for both sides to independently derive the keys without anything but the shared key/passphrase, and readily available values visible in the handshake are easy to intercept.

WPA2-Enterprise encryption always includes a server-side public key certificate along with either a client public key certificate or a client username/password. In a full WPA2-

Enterprise implementation, a full RADIUS would support a local database or pass the user credentials on to another external server like LDAP or Active Directory. Still, those measures do not factor into encryption strength. They are not included in this 'small office/home' solution, so our solution limits the username/password combination stored on the AP. Similarly, there will be no support for client certificates, as the complexity is not appropriate for this use case.

The main benefit of using client certificates is not related to encryption but merely to limit access to the network while also protecting user credentials [55]. By removing all of the features not associated with actual cryptography, our formula for this limited implementation of WPA2-Enterprise becomes basic username/password authentication plus asymmetric key exchange using public keys plus symmetric key generation. This retains the WPA2-Enterprise secure tunnel made possible by public-key encryption to exchange the more efficient symmetric keys used to encrypt the actual data and eliminates the WPA2- PSK shared key vulnerability.

As stated previously, the symmetric encryption overhead is 128 bits for both WPA2-PSK and WPA2-Enterprise. This is an important aspect of our solution. The actual encryption method and strength are the same. By offering a reduced subset of WPA2-Enterprise and leveraging the simplest authentication method plus dynamic key exchange, we bring Enterprise-level encryption to home and small business users. It is important to note that a related work, WPA3- Enterprise, allows for an increased symmetric key size of 192 bits, but only for Enterprise [56].

VII. Simulation Results

Based on our simulation of the operation stated above, we determined that our solution will only nominally increase the AP's load. We have discovered that the key exchange happens at the initial authentication and then at subsequent re-key intervals of 60 minutes [57]. Furthermore, the initial authentication, initial key generation, and subsequent key regeneration only occur at a trivial frequency for a small environment with a small number of devices such as a home or small business; nevertheless, the emulation also included the previous formulas to approximate the impact. We used Kali Linux VM (version: kali-Linux-2020.1-VMWare-amd64), which already had the Aircrack-ng package installed to simulate the three aforementioned operations, which we used to perform WPA2-PSK calculations. We also used the OpenSSL package along with Apache2 with HTTPS and "wget" [58] to approximate SSL overhead. As for RADIUS protocol overhead, we installed the FreeRadius package. Readily- available step-by-step tutorials guided us through the configuration of these packages using only a few commands. Furthermore, as mentioned above, there is no need to compare the symmetric key encryption portion. The data's actual encryption is the most compute-intensive portion of a complete WPA2 session, but it is common to both WPA2-PSK and WPA2-Enterprise.

All tests were performed on the same machine with nothing else running except VMware Workstation Player and a single Kali Linux virtual machine. Each test was run 1,000 times using a simple shell script [59]. Without a better way to measure overhead, the Unix time command acted as a precise stopwatch and allowed us to give an approximation of the increase in processing required.

The Aircrack-ng test was our baseline for the existing WPA2-PSK overhead and was also used as a generic hashing and key generation estimate for WPA2-Enterprise. The aircrack distribution contained a WPA2 test script and a sample wireless capture file containing 499 packets. Using the key corresponding to the WPA2 capture as the only attempted key, aircrack implemented all of the PSK calculations plus some overhead for the tool. We observed an average of 33.4 milliseconds for each WPA2 handshake as seen in Figure 8.

Fig. 8. WPA2 Key generation - Aircrack-ng with known PSK tests

We assumed that an actual RADIUS authentication using FreeRadius plus an SSL handshake using "wget" to the Apache HTTPS server on the same machine is roughly equivalent to approximate our solution a WPA2-Enterprise RADIUS call, which contains a similar public key exchange. Those two pieces, which average 38.5 milliseconds as seen in [Figure 9] and 24.8 milliseconds as illustrated in [Figure 10] when added to the WPA2-PSK key generation baseline average of 33.4 milliseconds, give our solution an average of 96.7 milliseconds. This ends up being a little less than a 3-times increase: 96.7/33.4 = 2.9 as seen in [Table 1]. An increase of 3 times sounds bad, but since this authentication/handshake

only occurs when initiating a connection and during a periodic re-key, which is typically once per hour, that increase should effectively not have an impact.

```
mo@gx:~/aircrack-ng-1.6/test$ time bash test1000rad > rad.log

real    0m38.534s
user    0m30.276s
sys     0m7.435s
mo@gx:~/aircrack-ng-1.6/test$ cat test1000rad
for i in {1..1000}
do
  echo $i
  radtest giovanna mypassword localhost 0 testing123
done
mo@gx:~/aircrack-ng-1.6/test$ radtest giovanna mypassword localhost 0 testing123
Sent Access-Request Id 197 from 0.0.0.0:56256 to 127.0.0.1:1812 length 78
        User-Name = "giovanna"
        User-Password = "mypassword"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 0
        Message-Authenticator = 0x00
        Cleartext-Password = "mypassword"
Received Access-Accept Id 197 from 127.0.0.1:1812 to 127.0.0.1:56256 length 20
mo@gx:~/aircrack-ng-1.6/test$
```

Fig. 9. FreeRadius clear text authentication tests

```
mo@gx:~/aircrack-ng-1.6/test$ time bash test1000ssl > ssl.log 2>ssl.err

real    0m24.794s
user    0m15.668s
sys     0m5.640s
mo@gx:~/aircrack-ng-1.6/test$ cat test1000ssl
for i in {1..1000}
do
  echo $i
  wget --no-check-certificate https://127.0.0.1:443/not_there
done
mo@gx:~/aircrack-ng-1.6/test$ wget --no-check-certificate https://127.0.0.1:443/not_there
--2020-05-08 19:17:29--  https://127.0.0.1/not_there
Connecting to 127.0.0.1:443... connected.
WARNING: The certificate of '127.0.0.1' is not trusted.
WARNING: The certificate of '127.0.0.1' doesn't have a known issuer.
The certificate's owner does not match hostname '127.0.0.1'
HTTP request sent, awaiting response... 404 Not Found
2020-05-08 19:17:29 ERROR 404: Not Found.

mo@gx:~/aircrack-ng-1.6/test$
```

Fig. 10. SSL handshake using wget and Apache2 tests

Moreover, aside from the transaction overhead, there is also a memory and storage footprint issue to consider when adding this new functionality to low-cost access points. Low-end APs will not have extra resources to run a full-featured RADIUS server. The overall size for the FreeRadius installation on Kali Linux was under 10 MB of hard drive space, and it consumed 67 MB of RAM when running. That seems like a large amount, but the documentation states that FreeRadius includes many enterprise features such as Active Directory integration, LDAP integration and support for third-party relational databases like Oracle, all of which are not required in this case. Removing those enterprise features from the code is likely to reduce storage and memory requirements for our solution. Nonetheless, even with all of the overhead from enterprise features, extremely low- end Linux devices such as the Raspberry Pi have been used to run the full-featured version of FreeRadius from as far back as 2013 [60]. We believe that the actual size of the required code, when pruned to only the core functionality needed in our solution, will be suitable for use on a low-end commodity access point.

VIII. CONCLUSION

We believe that our solution would bring enterprise-level encryption to more devices in the home and small business environments based on our research and testing. The solution leverages existing protocols and creatively applies them to take nominal development efforts and impose only a minumal overhead. Lastly, all of this should be possible without incurring an external RADIUS server's administrative overhead and will still be convenient for non-technical users.

Possible future work would focus on obtaining a more accurate model. Our solution is based on existing code and protocols, so the next step consisting of actually coding the solution, would accurately determine overhead. Using a subset of the open-source FreeRadius code on a small dedicated device like a Raspberry Pi would allow for accurate measurement of the extra load and resources consumed instead of using a virtual machine. Additional future work would be to merge that code with the readily-available code to add access point functionality to Linux machines, including the aforementioned Raspberry Pi [46].

The last item that would need to be considered in future work is the proper observation of the public key certificate portion. In our test, we used a self-signed certificate generated with the built-in OpenSSL package. In order to obtain the benefits of man-in-the-middle attack prevention, there would have to be a way to manage actual certificates.

TABLE I. TEST SUMMARY

Results for Multiple Test Cases		
Frequency (1000 operations)	***Time for 1000 Operations***	***Milliseconds per Transaction (Avg)***
WPA2 Key generation - aircrack-ng with known PSK	33.415	
FreeRadius clear text authentication	38.534	
SSL handshake using wget and Apache2	24.794	
Standard WPA2-PSK solution (just KeyGenerate)	33.415	33.4ms
Our solution (KeyGenerate+RADIUS+SS)	96.743	96.7ms
96.7 / 33.4 = 2.9x increase		

REFERENCES

[1] C. P. Kohlios and T. Hayajneh, "A Comprehensive Attack Flow Model and Security Analysis for WiFi and WPA3," Electronics 2018, vol. 7, no. 284, Oct. 2018.

[2] M. Ciampa, Security+ Guide to Network Security Fundamentals, 6th Ed. Cengage. 2017.

[3] N. Borisov, I. Goldberd, and D. Wagner, "Analysis of 802.11 Security or Wired Equivalent Privacy Isn't," in Mac-Crypto Conf. on Macintosh Cryptography, Cupertino, California, USA, 29-31 Jan. 2001.

[4] "Enterprise Mobility 4.1 Design Guide," Cisco. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Mobility/emob41dg/emob41dg-wrapper/ch4_Secu.html. A. Tripathi, "Relative Encryption Overhead in 802.11g Network," in 2008 Int. Symp. on Telecommunications, Tehran, Iran, 27-28 Aug. 2008.

[5] D. Rapp, "Keys, Keys, and Even More Keys!" Daleswifisec. [Online]. Available: https://dalewifisec.wordpress.com/2013/06/03/keys-keys-and-even-more-keys/amp/.

[6] "802.11 Network Security Fundamentals," Cisco. [Online]. Available: https://www.cisco.com/en/US/docs/wireless/wlan_adapter/secure_client/5.1.0/administration/guide/C1_Network_Security.html#wp1050709.

[7] M. Vanhoef and F. Piessens, "Release the Kraken: New KRACKs in the 802.11 Standard," in Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, 2018, pp. 299–314.

[8] M. Vanhoef, D. Schepers, and F. Piessens, "Discovering Logical Vulnerabilities in the WiFi Handshake Using Model-Based Testing," in Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, New York, NY, USA, 2017, pp. 360–371.

[9] I. Reddy, V. Srikanth, "Review on Wireless Security Protocols (WEP,WPA,WPA2 & WPA3)," IJSRCSEIT, vol. 5, no. 4, pp. 28–35, Aug. 2019.

[10] K. Lounis and M. Zulkernine, "Bad-Token: Denial of Service Attacks on WPA3," in Proceedings of the 12th International Conference on Security of Information and Networks, New York, NY, USA, 2019.

[11] M. Vanhoef and E. Ronen, "Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd," in IEEE Symposium on Security and Privacy, San Francisco, California, 2020.

[12] R. Heartfield et al., "A taxonomy of cyber-physical threats and impact in the smart home," Computers & Security, vol. 78, pp. 398–428, 2018.

[13] A. Bartoli, E. Medvet, A. De Lorenzo, and F. Tarlao, "(In) Secure Configuration Practices of WPA2 Enterprise Supplicants," in Proceedings of the 13th International Conference on Availability, Reliability and Security, New York, NY, USA, 2018.

[14] S. Brenza, A. Pawlowski, and C. Pöpper, "A Practical Investigation of Identity Theft Vulnerabilities in Eduroam," in Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks, New York, NY, USA, 2015.

[15] M. Vanhoef and F. Piessens, "Advanced WiFi Attacks Using Commodity Hardware," in Proceedings of the 30th Annual Computer Security Applications Conference, New York, NY, USA, 2014, pp. 256– 265.

[16] A. Rahman and M. Ali, "Analysis and Evaluation of Wireless Networks by Implementation of Test Security Keys," in Emerging Technologies in Computing, Cham, 2018, pp. 107–126.

[17] F.-C. Cheng, "Automatic and Secure WiFi Connection Mechanisms for IoT End-Devices and Gateways," in Emerging Technologies in Computing, Cham, 2018, pp. 98–106.

[18] K. Heinäaro, "Cyber attacking tactical radio networks," in 2015 International Conference on Military Communications and Information Systems (ICMCIS), Cracow, Poland, 2015, pp. 1–6.

[19] A. Bartoli, E. Medvet, and F. Onesti, "Evil twins and WPA2 Enterprise: A coming security disaster?," Computers & Security, vol. 74, pp. 1–11, 2018.

[20] C. McMahon Stone, T. Chothia, and J. de Ruiter, "Extending Automated Protocol State Learning for the 802.11 4-Way Handshake," in Computer Security, Cham, 2018, pp. 325–345.

[21] M. A. Garcia, D. J. S. Martinez, and J. Castillo-Velazquez, "Hardening Applied Over a WLAN SOHO Environment for Mitigation of Vulnerabilities," in 2018 IEEE 38th Central America and Panama Convention (CONCAPAN XXXVIII), San Salvador, El Salvador, 2018, pp. 1–6.

[22] J. Castillo-Velazquez, M. A. Garcia, and D. J. S. Martinez, "Hardening as a best practice for WLAN Security Meanwhile WPA3 is released," in 2019 IEEE 39th Central America and Panama Convention (CONCAPAN XXXIX), Guatemala City, Guatemala, 2019, pp. 1–5.

[23] L. Galeazzi Avalos, C. Barría Huidobro, and T. Villegas Berbesi, "Identifying Components Belonging to Wireless Connectivity Security," in Telematics and Computing, Cham, 2019, pp. 365–373.

[24] D. Schepers, A. Ranganathan, and M. Vanhoef, "Practical Side-Channel Attacks against WPA-TKIP," in Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, New York, NY, USA, 2019, pp. 415–426.

[25] M. Vanhoef and F. Piessens, "Predicting, Decrypting, and Abusing WPA2/802.11 Group Keys," in 25th USENIX Security Symposium (USENIX Security 16), Austin, TX, 2016, pp. 673–688.

[26] M. Fruhmann and K. Gebeshuber, "Radio Frequency (RF) Security in Industrial Engineering Processes," in Security and Quality in Cyber- Physical Systems Engineering: With Forewords by Robert M. Lee and Tom Gilb, S. Biffl, M. Eckhart, A. Lüder, and E. Weippl, Eds. Cham: Springer International Publishing, 2019, pp. 413–441.

[27] M. Vanhoef and F. Piessens, "Release the Kraken: New KRACKs in the 802.11 Standard," in Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, 2018.

[28] M. Vanhoef and F. Piessens, "Symbolic Execution of Security Protocol Implementations: Handling Cryptographic Primitives," in 12th USENIX Workshop on Offensive Technologies (WOOT 18), Baltimore, MD, 2018.

[29] T. Chang, C. Chen, H. Hsiao, and G. Lai, "The Cryptanalysis of WPA & WPA2 Using the Parallel-Computing with GPUs," in Mobile Internet Security, Singapore, 2018, pp. 118–127.

[30] D. Wang, X. Zhang, Z. Zhang, and P. Wang, "Understanding security failures of multi-factor authentication schemes for multi-server environments," Computers & Security, vol. 88, p. 101619, 2020.

[31] A. Bartoli, "Understanding Server Authentication in WPA3 Enterprise," Preprint, Jan. 2020.

[32] A. Abdelrahman, H. Khaled, E. Shaaban, and W. S. Elkilani, "WPA-WPA2 PSK Cracking Implementation on Parallel Platforms," in 2018 13th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 2018, pp. 448–453.

[33] J. Opio, "Wifi Protected Access, WPA2 Residential: Security Solution for Residential Application," Worcester Polytechnic Institute, March 2019. https://www.researchgate.net/publication/332069909

[34] M. Horowitz, "WPA3," Router Security. 11 Aug. 2018. Updated 11 April. 2019. https://routersecurity.org/wepwpawpa2.php

[35] "New features availablewith iOS 13." Apple. https://www.apple.com/ios/ios-13/features/

[36] "Support for WiFi Protected Access 3 (WPA3) on Intel® Wireless Adapters," Intel, updated 9 Mar. 2020. https://www.intel.com/content/www/us/en/support/articles/000054783/n etwork-and-i-o/wireless-networking.html

[37] "Release Notes for Cisco Wireless Controllers and Lightweight Access Points, Cisco Wireless Release 8.10.121.0," Cisco, 4 April. 2020. https://www.cisco.com/c/en/us/td/docs/wireless/controller/release/notes/ crn810mr2.html

[38] A. Sood et all, "Cisco Enterprise Wireless Intuitive WiFi starts here, " 2nd edition, Cisco. 2018. https://www.cisco.com/c/dam/en/us/products/collateral/wireless/nb-06- wireless-wifi-starts-here-ebook-cte-en.pdf

[39] "WPA3,"Aruba. https://www.arubanetworks.com/techdocs/Instant_83_WebHelp/Content/Instant_UG/Authentication/wpa3_authentication.htm

[40] S. Gollakota, N. Ahmed, N. Zeldovich, and D. Katabi, "Secure In-Band Wireless Pairing," in SEC'11: Proc. 20th USENIX conf. Security, Berkeley, CA, Aug. 2011. https://people.csail.mit.edu/nickolai/papers/gollakota-tep.pdf

[41] S. Gupta, N. Nanda , N. Chhikara3, N. Gupta, and S. Jain, "Mutual Learning in Tree Parity Machines Using Cuckoo Search Algorithm for Secure Public Key Exchange," in ICTACT Journal on Soft Computing, April 2018, Volume: 08, Issue: 03. http://ictactjournals.in/paper/IJSC_Vol_8_Iss_3_Paper_3_1663_1667.pdf

[42] M. M. Alani, "Applications of machine learning in cryptography: a survey," in Proc. 3rd Int. Conf. on Cryptography, Security and Privacy (ICCSP' 19). Association for Computing Machinery, New York, NY, USA, 2019.

[43] A. Klimov, A. Mityagin, and A. Shamir, "Analysis of Neural Cryptography," in Proc. 8th Int. Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT'02). Springer-Verlag, Berlin, Heidelberg, 288–298, 2002.

[44] "Configuring RADIUS Authentication With WPA2-Enterprise," Cisco Meraki. https://documentation.meraki.com/MR/Encryption_and_Authentication/Configuring_RADIUS_Authentication_with_WPA2-Enterprise#RADIUS_Configuration

[45] S. Lovely, "How to use your Raspberry Pi as a wireless access point," The Pi. 2017. [46] https://thepi.io/how-to-use-your-raspberry-pi-as-a-wireless-access-point/

[46] "Releases," FreeRADIUS. https://freeradius.org/releases/

[47] M Wuttke, "Tiny Radius: Java Radius library," Sourceforge. http://tinyradius.sourceforge.net/

[48] "Welcome to the OpenWrt Project," OpenWrt. Updated 2020. https://openwrt.org/

[49] "DD-WRT," DD-WRT.com. https://dd-wrt.com/

[50] "Understanding PSK Authentication," Juniper. 8 Jan. 2019. [Online]. Available: https://www.juniper.net/documentation/en_US/junos-space- apps/network-director3.1/topics/concept/wireless-wpa-psk-authentication.html

[51] " User Guide for Cisco Secure Access Control System 5.8," Cisco, updated 9 Aug. 2018. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/net_mgmt/cisco_secure_access_ control_system/5-8/user/guide/acsuserguide/eap_pap_phase.html

[52] L. Nussel," The Evil Twin problem with WPA2-Enterprise," version 1.1. SUSE Linux products GmbH. 19 Apr. 2010. [Online]. Available: https://users.suse.com/~lnussel/The_Evil_Twin_problem_with_WPA2-Enterprise_v1.1.pdf

[53] Z. Yang, A. C. Champion, B. Gu, X. Bai, and D. Xuan, "Link-Layer Protection in 802.11i WLANs with Dummy Authentication," The Ohio State University. 2009. [Online]. Available: http://web.cse.ohio-state.edu/~champion.17/pubs/09_wisec_ycgbx.pdf

[54] T. S. Hendershot, "Towards Using Certificate-Based Authentication as a Defense Against Evil Twins in 802.11 Networks," Brigham Young University Scholars Archive, 1 Nov. 2016. [Online]. Available: https://scholarsarchive.byu.edu/cgi/viewcontent.cgi?article=7115&conte xt=etd

[55] "Discover WiFi Security," WiFi. [Online]. Available: https://www.wi-fi.org/discover-wi-fi/security

[56] "Cisco Wireless Controller Configuration Guide, Release 8.5," Cisco, updated 17 Apr. 2020. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/wireless/controller/8-5/config-guide/b_cg85/wlan_security.html

[57] Sohail, "12 Practical Examples Of wget Command In Linux," Linuxandubuntu. 13 2019. [Online]. Available: http://www.linuxandubuntu.com/home/12-practical-examples-of-wget- command-on-linux

[58] "Bash for loop with range {#... #}," Askubuntu. [Online]. Available: https://askubuntu.com/questions/166583/bash-for-loop-with-range

[59] G. Krause, "Raspberry PI based FreeRadius Server with GUI," Binary Heartbeat. Dec. 2013. [Online]. Available: http://www.binaryheartbeat.net/2013/12/raspberry-pi-based-freeradius- server.html