VOUCH-AP: priVacy preserving Open-access 802.11 pUbliC Hotspot AP authentication mechanism with co-located evil-twins

Avinash Srinivasan

Department of Computer and Information Sciences
Temple University, Philadelphia, PA 19122
Email: avinash@temple.edu

Jie Wu

Department of Computer and Information Sciences
Temple University, Philadelphia, PA 19122
Email: jiewu@temple.edu

Abstract: Open-access 802.11 public Wi-Fi hotspots have become a basic necessity for hundreds of millions of mobile users' persistent on-the-go access to the *Internet*. 802.11 Wi-Fi networks are designed and deployed to support rudimentary *low-level authentication* at the link layer enabling an *AP* to decide whether to allow a client to associate. Similar authentication mechanisms are not provisioned for the clients. Hence, there is a fundamental *information asymmetry* at play in an 802.11 public hotspot, which tilts the balance in favor of an adversary intending to launch *AP*-based evil-twin attacks. Furthermore, link-layer authentication has little security since the link itself is completely open to numerous attacks. In this paper, we address this *information asymmetry* problem and propose a simple yet powerful solution for identifying and eliminating malicious APs, thereby providing users safe and private 802.11 public hotspots. Our proposed *AP* authentication framework is called *VOUCH-AP*, a portable and platform-independent solution. *VOUCH-AP* is, to the best of our knowledge, the first work to consider digital certificate based *AP* authentication. *VOUCH-AP* makes use of a modified version of a *X.509* digital certificate consisting of additional fields for provisioning robust security and privacy to counter evil-twin attacks. The proposed solution does not require any hardware upgrades or specialized hardware, unlike 802.11i (aka WPA2). Finally, through security analysis, we show the security robustness of the proposed *VOUCH-AP* framework to counter evil-twin attacks.

Keywords: Authentication, captive portal, evil-twin, identity theft, privacy, security, vulnerability.

Reference to this paper should be made as follows: Srinivasan and Wu '**VOUCH-AP:** pri**V**acy preserving **O**pen-access 802.11 p**U**bli**C** **H**otspot **AP** authentication mechanism with colocated evil-twins', *Int. J.* , Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Prof. Avinash Srinivasan is a faculty with of Computer & Information Sciences at Temple University. His research interests include network security & forensics, security in critical infrastructure, SCADA live forensics, and cross-domain security solutions. He has published 46 refereed conferences and journals articles. Dr. Srinivasan holds *CEH* and a *CHFI* certifications. He has trained law enforcement personnel and civilians in security and forensics topics. He is a *Fellow of National Cybersecurity Institute* in Washington D.C., and member of the EUROPOL Platform for Experts for the *Criminal Use of Information Hiding* and *Data Protection*.

Professor Jie Wu is a Laura H. Carnell Professor of Computer and Information Sciences at Temple University and a Fellow of the IEEE. Prior to joining Temple University, he was a program director at the NSF and Distinguished Professor at Florida Atlantic University. His research interests include wireless networks, mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. Dr. Wu's publications include over 600 papers in scholarly journals, conference proceedings, and books. He has served on several editorial boards, including IEEE Transactions on Computers and JJPDC. He was general co-chair for IEEE MASS-2006, IEEE IPDPS-2008, and IEEE DCOSS-2009 and was the program co-chair for IEEE INFOCOM-2011.

1 Introduction

Today, Wi-Fi is the predominant access technology for mobile devices. As more mobile devices become Wi-Fi enabled, the number of public hotspots increases in several orders of magnitude and user acceptance grows [1]. Consumers are increasingly using wifi to save on monthly cellular bills with exorbitant data rates. Therefore, growing customer needs and business pressure coupled with technology advances in wireless and mobile computing resulted in vendors' commoditizing wireless access points (*APs*) (Def. 1). Such commoditization is often accompanied by *default admin accounts*, *guest accounts*, *default disabled security settings*, etc, that result in little to no security for the device and the information. While both fixed and mobile hotspots [6] are in use, in this paper, we specifically focus on static hotspots.

The most serious vulnerabilities that threaten users' security and privacy stem from users' own complacence (Threat 1). Most users, if not all, tend to be primarily driven by a "myopic objective"; they solely focus on free Internet connection. The existing risks and vulnerabilities of 802.11 public hotspots (discussed in section 5) coupled with user complacence, and the lack of tools and techniques for client authentication *AP*s readily provide adversaries a fertile ground for hacking.

The adversary now simply needs to do introduce an *AP* that he controls – known as an "evil-twin" (AP_{et}) – with a *Service Set Identifier (SSID)* (Def. 3) that can successfully lure users as a legitimate and "free" *AP*. To further augment the attack's success rate, the adversary can spoof the *Media Access Control* (MAC) address of the legitimate *AP* (AP_{leg}). With an evil-twin *AP* under his control, the adversary can cause numerous other attacks such as – *Denial of Service* (DoS), *Address Resolution Protocol* (ARP) poisoning, and *Domain Name System* (DNS) poisoning.

Validation of a client wireless device by an *AP* is usually accomplished through a "*captive portal*," where the client is re-directed to the captive portal web page. Upon successful authentication and a payment if needed, the captive portal allows the client device to access the Internet. However, captive portals are themselves vulnerable to *man-in-the-middle* (MITM) attacks – details of which has been discussed in [13]. This results in serious security and privacy concern, as presented in Threat 3. It is possible that authentication credentials and payment information submitted on the captive portal may be secured with a *run-of-the-mill* security mechanism, but such security measures are easily exploited by a sophisticated adversary.

Furthermore, the hotspot is highly vulnerable with unencrypted link-layer transmissions prior to client association with the *AP* (Threat 2). Therefore, the captive portal's authenticity itself must be verified prior to the submission of any sensitive information. Therefore, Threats 2 and 4 collectively justify the need to empower users to make informed and conscientious decision when connecting to *AP*s in public hotspots. Therefore, providing users with appropriate security tools and mechanisms that can help offset the information asymmetry is critical.

Existing security solutions are primarily designed for enterprise class networks with powerful security appliances on the *Wireless Local Area Network* (WLAN) and significant computing resources. Such resource/computational intensive solutions are not suitable for 802.11 public hotspots operating in unique environments presented in Threat 5.

Finally, there in one other feature on most wireless/mobile client device *Operating System* (OS). This is the ability of the device OS to automatically re-associate with a known/previously used *AP*, presented in Threat 6. Unfortunately, this feature is usually enabled by default. With the automatic re-association feature enabled, the attacker can easily deploy an *AP* with the same identity – *SSID*, *MAC* address, etc. – as one of a client's previously-used and trusted *AP*s.

While the fix to this problem is a simple change in the device settings to disable this feature. If not disabled, this particular feature significantly increases the threat of *evil-twin* attacks to the owner of the vulnerable device. The attacker can use jamming and other denial-of-service attack techniques to specifically prevent targeted clients from associating to secure *AP*s, thereby passively coercing the clients to associate with an evil-twin managed by the adversary.

In this paper, we propose a robust authentication protocol for securing information exchanged between a client device and an 802.11 public Wi-Fi hotspot through an unsecured *AP*. We present the design, implementation, and operations details for the proposed solution framework that enable the user to select an authentic *AP* to access the Internet through a 802.11 public hotspot.

To the best of our knowledge, no known work aims to equip end users with a portable and lightweight authentication tool to authenticate a public hotspot *AP* before associating with it. Our proposed *VOUCH-AP* solution is designed around *Public Key Infrastructure* (PKI) and primarily driven by *Digital Certificates* (Def. 7).

The assurance of security (authenticity) is conveyed to the user through a certificate that the *AP* possesses. This digital certificate is issued by a verifiable, trusted third party called a *Certification Authority CA* (Def. 8). The *CA* validates the identity of the $AP's$ certificate by digitally signing the certificate to confirm it has not been forged or altered in any way.

1.1 Threat Model and Assumptions

The threats presented by malicious *AP*s cannot be eliminated or even substantially mitigated using security mechanisms like *firewalls, WPA2, 802.1x, NAC, anti-virus* or *wired side scanners*. The adversary is aware of the security measures that are in place to detect and isolate unknown and malicious APs in 802.11 public hotspots.

The adversary is also aware of the proposed certificate based AP authentication mechanism. The adversary is assumed to be adaptive and intelligent, capable of posing as a regular client and engaging the AP in message exchanges. Through the exchanged messages and broadcast beacon frames, the adversary will attempt to identify potential attack vectors specific to the attack surface.

Additionally, we assume that the adversary can, at will, attempt any attack against the $CL \leftrightarrow AP$ communication channel – *replay, modification, sniffing, packet injection, etc.* However, the adversary has two limitations:

1. *no means of forging an authentic AP certificate*;
2. *no way of responding correctly during the Challenge-Response authentication process.*

Nonetheless, the adversary can attempt "guessing," "brute force," or any other type of attack against the communication channel in an effort to break the encrypted channel.

1.2 Motivation and Contributions

Through the implementation of certificates or other means, a robust wireless network validation model to authenticate public hotspots is a necessary tool to combat cyber-crimes. Cyber-crimes leverage the lack of such tools and the trusting nature of users. Our proposed approach is the first attempt to authenticate public APs to counter "evil-twin" attacks leveraging digital certificates.

With the proposed certificate-based robust authentication mechanism, the user can authenticate APs that it identifies during the discovery phase and choose to connect to an AP of choice based on several different parameters including – signal strength, service provider, and service fees. Using our robust authentication mechanism, users can protect themselves against "evil-twin" attacks when accessing the Internet in 802.11 public hotspots.

The security boot-strap is intended to ensure confidentiality before sensitive information, such as authentication credentials or encryption keys, is exchanged on the link. There is currently no known work that aims to equip end users with a portable and lightweight tool to authenticate a public hotspot AP before associating with it.

1.3 Paper Organization

The rest of this paper is organized as follows. In section 2, we provide discussions on relevant background. We also present a detailed review of digital certificates, which is at the heart of the proposed authentication mechanism. In section 3 we provide a summary of all threats lurking in public Wi-Fi APs. Necessary preliminaries and the problem statement are presented in Section 4, followed by a review of 802.11 public Wi-Fi hotspots in Section 5. In Section 6, we discuss our proposed authentication mechanism and follow with a detailed analysis of solution's security robustness to popular attacks in Section 7. We present a detailed review of relevant work in Section 8 and conclude the paper in Section 9. Formal definitions of important technical terms are presented in Section 10.

2 Background

2.1 Overview of Digital Certificates

A digital certificate is issued by a CA who is a trusted third party serving as a administrative entity within a cryptographic infrastructure referred to as a PKI. Some popular CAs include *VeriSign Inc.* and *Thawte Consulting*, but a CA can also be a government agency or a smaller company such as *Go Daddy.*

A CA issues a certificate with its *digital signature* (Def. 12), i.e., encrypted with its *private key (K_{pri})* (Def. 14), containing the principal's *public key (K_{pub})* (Def. 15) and a variety of other identification information such as – *principal's name*, *serial number*, *issue date*, and *expiration date.* Note that a message can be encrypted with either the K_{pub} or the K_{pri} and subsequently decrypted with the other key. Signing the certificate with the CA's private key enables the recipient of the certificate to verify the authenticity of the certificate using the CA's public key. Today, the most widely used standard for certificates is the $X.509$ standard.

2.2 Digital Certificates & Wireless Security

- provides participants with secure and reliable means to obtain necessary public keys
- fundamental for most communication systems that require both a high level of trust between the communicating parties and security assurance
- used to provide one or more of the following key security services – *Confidentiality, Integrity, Authentication,* and *Non-repudiation*
- decouples the dependence on *pre-shared keys* (PSKs) and *out-of-band* information exchange to communicate over an unsecured link
- suitable for robust security solutions since they scale very well with little overhead, suit distributed operations, and are very effective in adding and removing nodes
- very effective in countering attacks stemming from *Sybil* nodes

3 Summary of Threats to User Security and Privacy in Public 802.11 Hotspots

Threat 1 ***User Complacence.*** *Users' are extremely complacent when it comes to the security and privacy of their data. This is especially magnified when accessing the Internet through public hotspots primarily focused on free Internet access.*

Threat 2 ***Vulnerable Low-level Authentication.*** 802.11 *authentication is not only vulnerable to snooping attacks, since all link-layer traffic is relayed unencrypted, but also to attacks leveraging information asymmetry since an AP can execute the in-built low-level authentication to accept/reject a client, but not vice versa.*

Threat 3 ***Malicious Captive Portals.*** *With control over a malicious AP positioned as a MITM, the adversary can redirect user traffic to a malicious captive portal.*

Threat 4 ***Lack of Security Tools and Mechanisms for*** CL ***Authentication of*** AP***.*** *Even if a user is security and privacy conscious and wants to verify the authenticity of an AP, no security tools or techniques available to this aim.*

Threat 5 ***Unique Operational Environment.*** *Open access* 802.11 *public hotspots (Def. 10) have a unique operational environment. Some unique aspects include resource constraints – bandwidth, processor, and physical memory, make providing required levels of security and privacy difficult.*

Threat 6 ***Automatic Client (Re)association with Known APs.*** *When a wireless client device returns to a previously-used open* 802.11 *hotspot, the OS on the client device automatically scans the area for available APs through active probing and* re-associates *to one of the APs, unbeknownst to the user.*

Threat 7 ***802.11i with 802.1X/EAP (Extensible Authentication Protocol) impractical.*** 802.11*i compatibility requires a mandatory upgrade of AP hardware.* 802.1*X/EAP also makes an unrealistic assumption that the connection between the client and the AP is secure, which is incorrect in wireless medium.*

4 Preliminaries & Problem Statement

Let $H = \langle h_1, h_2 \cdots h_x \rangle$ be the set open 802.11 public Wi-Fi hotspot at a given location l_p from the set $L = \langle l_1, l_2 \cdots l_y \rangle$. Let $AP = \langle ap_1, ap_2 \cdots ap_z \rangle$ be the set of access points for a given hotspot $h_q \in H$ at location $l_p \in L$. Finally, let $CL = \langle cl_1, cl_2 \cdots \rangle$ be the universal set of wireless client devices that can access the *Internet* for a given pair of the form $\langle l_p, h_q \rangle$. Now, the problem addressed in this paper can be formally stated as follows –

Problem 1 *How can a client $cl_i \in CL$ reliably identify a legitimate access point $ap_x \in AP$ that is secure, robust to attacks, and preserves cl_i's privacy at all costs when accessing internet for the pair $\langle l_p, h_q \rangle$? Further, can the client $cl_i \in CL$ be equiped with appropriate security mechanisms to verify the authenticity of any* 802.11 *public hotspot $ap_x \in AP$ for the pair $\langle l_p, h_q \rangle$, prior to submitting sensitive and private information?*

Axiom 4.1 *Given a hotspot h_p at a location l_q, there exists a* ***many-to-one*** *relation between the hotspot and all available access points at location l_q.*

Axiom 4.2 *Given a geographic location l_q, there exists a* ***many-to-one*** *relation between the location and all available hotspots.*

Axiom 4.3 *A unique fingerprint exists for any given quadruple $\langle l_p, h_q, ap_r, ssid_s \rangle$.*

Lemma 1 *Any given pair $\langle ap_r, ssid_s \rangle$ always exhibits a strict* one-to-one *relation.*

Proof 1 *Consider two access points ap_i and ap_j for the pair $\langle l_p, h_q \rangle$. Now, if $i \neq j$ and $\langle ap_i^{ssid} == ap_j^{ssid} \rangle$, then there will be ambiguity during the discovery phase when client devices attempt to scan the hotspot in l_k and receive two probe responses from two different APs with the same SSID. Such situations, if not properly addressed to eliminate the ambiguity, make it easy for the attacker to execute evil-twin attacks. Therefore, with a security policy that strictly enforces only one AP with a given SSID for any given pair $\langle l_p, h_q \rangle$, either equation 1 or equation 2 will hold true in this scenario.*

$$(i == j) \implies ap_i^{ssid} == ap_j^{ssid} \tag{1}$$

OR

$$(i \neq j) \implies ap_i^{ssid} \neq ap_j^{ssid} \tag{2}$$

Axiom 4.3 states that any violation of the strict one-to-one *relationship exhibited by every pair $\langle ap_r, ssid_s \rangle$ serves as an indicator of threat from evil-twin(s).*

Lemma 2 *Two access points ap_i and ap_j can have the same SSID if they are in two non-overlapping hotspots H_x and H_y, i.e., $ap_i^{ssid} == ap_j^{ssid}$ if and only if $x \neq y$.*

Proof 2 *Let ap_i be an access point in hotspot h_x. Let ap_j be an access point in hotspot h_y. Now, if $x \neq y$, then $ap_i^{ssid} == ap_j^{ssid}$. This follows from Lemma 1. Alternately, "If $[ap_i^{ssid}, h_x]$ and $[ap_j^{ssid}, h_y]$," then we have the result shown in equation 3.*

$$ap_i^{ssid} \neq ap_j^{ssid} \implies (x == y) \tag{3}$$

5 802.11 **Public Hotspot Security Posture**

5.1 Why not use 802.11i?

802.11i is an amendment to 802.1x proposed by NIST to provide security for WLAN communications. 802.11i provides improved encryption for networks that use the popular 802.11a, 802.11b (which includes Wi-Fi), and 802.11g standards. However, this is not a feasible solution since the majority of APs are legacy devices and are not capable of supporting the computation intensive 802.11i standards. Following are some key limitations preventing the adoption of 802.11i:

- Does not specify what authentication protocols to use since the primary focus of 802.11i is the link layer while most authentication protocols run above the link layer in the communication stack.
- Offers improved security by using the symmetric *Advanced Encryption Standard* (AES) encryption algorithm. AES is a more secure alternative to the *Rivest Cipher 4* (RC4) stream cipher used by *Wired Equivalent Privacy* (WEP) and *Wi-Fi Protected Access* (WPA), but it adds significant overhead on a node's resources. Supporting AES, which is the building block of 802.11i, requires a dedicated chip to suport computing needs. This mandates a hardware upgrade for existing Wi-Fi networks with non-compatible hardware, which is neither practical nor feasible given the expanse and requirements of public Wi-Fi hotspots.
- Requires the client to obtain a *Maser Key* from the *Authentication Server* (AS) ahead of time that will be used to derive a *Pairwise Master Key* (PMK). As for the Wi-Fi hotspot, the AS provides the *AP* with the PMK. Having the same PMK, the client and *AP* now start a communication session using the PMK as the session key.
- Confidentiality service only applies to data frames, not to management frames. This can be a significant vulnerability since information contained in management frames can be leveraged to launch a variety of attacks.
- Supports client authenticating to the *AP* by establishing a *Secure Sockets Layer* (SSL)/ *Transport Layer Security* (TSL) connection. While the information transmitted during an SSL session cannot be viewed by a third party, the *Internet Protocol* (IP) address of the sender and receiver, the DNS request to resolve the hostname, the port numbers used, and the quantity of data sent, are all visible. Also, during the SSL/TLS session bootstrap phase, the attacker can potentially sniff the cryptographic keys because the communication link between the client and *AP* is unencrypted.
- Ensures secure communication by employing 802.1X authentication with EAP. However, for public 802.11 hotspots with unsecured APs, this is not a feasible solution because of the security concern noted in Threat 7.

5.2 Unsecured 802.11 *Public Hotspots*

In accordance with the IEEE 802.11 standards specification, the process of connecting to a WLAN subsystem through a wireless *AP* is a three-phase process: *i) the probe phase*, *ii) the authentication phase (Def. 17)*, and *iii) the association phase (Def. 17)*. In addition, 802.11 standard specifies the following two types of *link-level* authentication:

5.2.1 Open Key Authentication.

Open key authentication is a two step process:

1. an authentication request from the client with its ID (typically the MAC address)
2. an authentication response from the *AP* with a *success* or *failure* message

The client proceeds to the *association* phase only if the response message is a *success* message. While this authentication is feasible in the scenario of public hotspots, it is unsecure and biased in favor of the *AP* and therefore, indirectly in favor of the adversary controlling an *evil-twin.*

5.2.2 Shared Key Authentication.

With this type of *link-layer* authentication, either a *shared key* or a *passphrase* is established offline and manually pre-loaded on both the client device and the *AP*. Note that several types of *Small Office Home Office* (SOHO) WLAN shared key authentication mechanisms are available, including WEP, WPA, and WPA2. However, this authentication mechanism is not practical for public hotspots because of the difficulty of establishing a pre-shared key or pass-phrase offline.

The client device and the *AP* will exchange a series of 802.11 management frames in order to get to an *authenticated* and *associated* state. Consequently, there are three distinct possible states that a given *client* device can be in (relative to a given *AP*), and the three states denoted as a pair are:

1. ⟨*Unauthenticated, Unassociated*⟩
2. ⟨*Authenticated, Unassociated*⟩
3. ⟨*Authenticated, Associated*⟩

5.3 Client Authentication & Association to AP

A client device must be in a ⟨*Authenticated, Associated*⟩ state before the corresponding *AP* can grant the client access to the Internet. This complete process of authentication and association to the public Wi-Fi access point has been captured and illustrated as a finite state machine in Figure 1. Below is a detailed step-by-step description of this process.

Step-1. Client sends a *probe request* advertising its capabilities and requirements as a layer-2 broadcast message. The broadcast message is addressed to the BSSID (Def. 4) of ***0xff:ff:ff:ff:ff:ff***.

Step-2. All *AP*s receiving the *probe request* broadcast message send back a *probe response* message in which each *AP* advertises its *SSID* and capabilities, such as encryption and data rates.

Step-3. Client reviews all the received *probe response* messages mainly for the capabilities, and then chooses a compatible network from the available options.

Step-4. Client now attempts a 802.11 *low-level authentication* by sending an authentication frame. Within the frame, it sets *"authentication = open"* and the *"sequence* = 0*x*0001*"*. No data encryption or security is available at this stage since the client cannot opt for shared-key authentication (see Threat 7) unless it has subscribed to the hotspot service provider and has the necessary pre-shared key(s). Most importantly, 802.11 low-level authentication is **not the same** as as WPA2 (802.11*i*) or 802.1*X* authentications that occur post *authentication* and *association*, leveraging the resources and protocols available on the wired side of the *AP*.

Step-5. Upon receiving a client's authentication request, *AP* responds with a frame by simply changing the sequence number in the received frame to *"sequence* = 0*x*0002*"*.

Step-6. *AP* will tag a client's state as *"unauthenticated & unassociated"* if it receives any frame other than a *probe request* or *authentication*. Additionally, as a fail-safe mechanism, it responds with a *"de-authentication"* frame.

Step-7. Client begins the *association* process after successful 802.11 *low-level authentication*. Note that 802.11 does allow wireless client devices to be low-level authenticated to multiple *AP*s simultaneously. However, a client can still be only actively associated and transferring data through only a single *AP* at any given time.

Step-8. If a client is authenticated to multiple *AP*s, then it determines which *AP* it would like to associate to, and then send an *"association request."* This decision is primarily based on the capabilities of the *AP* and the corresponding hotspot.

Step-9. Now, if the *AP* receives frames from a client that is *"authenticated"* but not yet *"associated"*, *AP* will respond with a *"disassociation"* frame. *AP* also tags that client's state as *"authenticated & unassociated."*

Step-10. Once a client's association request is successful, *AP* will create an *"Association ID"* for that client and respond with an *"association successful"* response message along with the *"Association ID."*

Step-11. Once the client receives the *"Association ID,"* it is actively associated with that *AP*. At this point, the client has *Internet* access and data transfer can begin.

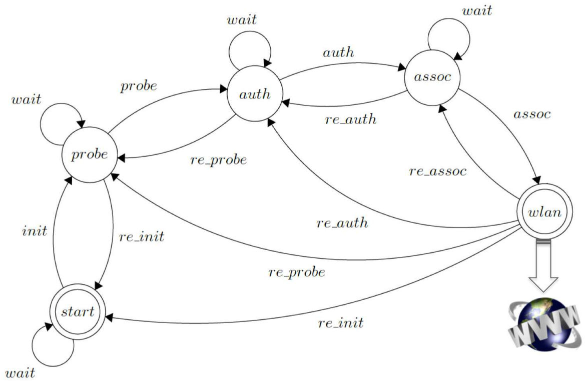

Figure 1 A wireless client device accessing the Internet through a public hotspot.

6 *VOUCH-AP* Authentication Mechanism

Our proposed *VOUCH-AP* is two-stage authentication mechanism. In the first stage, the client authenticates an *AP* that it intends to associate with. To this aim, the client requests the *AP* for its digital certificate using which it can verifies the authenticity of the *AP*. The stage one authentication concludes with the client and the *AP* having negotiated a session key in a secure manner, following which the second stage can be executed in a secure environment.

Note that stage one authentication is a critical prerequisite to stage two authentication since Wi-Fi traffic is unencrypted and vulnerable to an array of other MITM attacks in public hotspots. During the stage two authentication, the client authenticates to the service provider or to another entity like the client's protected enterprise network. During this stage, a secure *captive portal* is often the choice for authenticating to the local service administration entity or the the service provider. Remote connections to enterprise or other protected environments are typically via WPA or WPA2 with 802.1X-EAP/*Temporal Key Interchange Protocol* (TKIP).

Once at the log-in portal, new users are further redirected to a registration page where they are required to register to access the internet via that *AP*. Existing registered users can directly access the internet by authenticating themselves at the log-in page. When the user fails to provide valid authentication credentials n times within a time period Δ_{try}, further attempts will be blocked for a time period Δ_{wait}, to prevent online brute force attacks. The user has to then undergo the above process again to connect to the *AP* and access the internet.

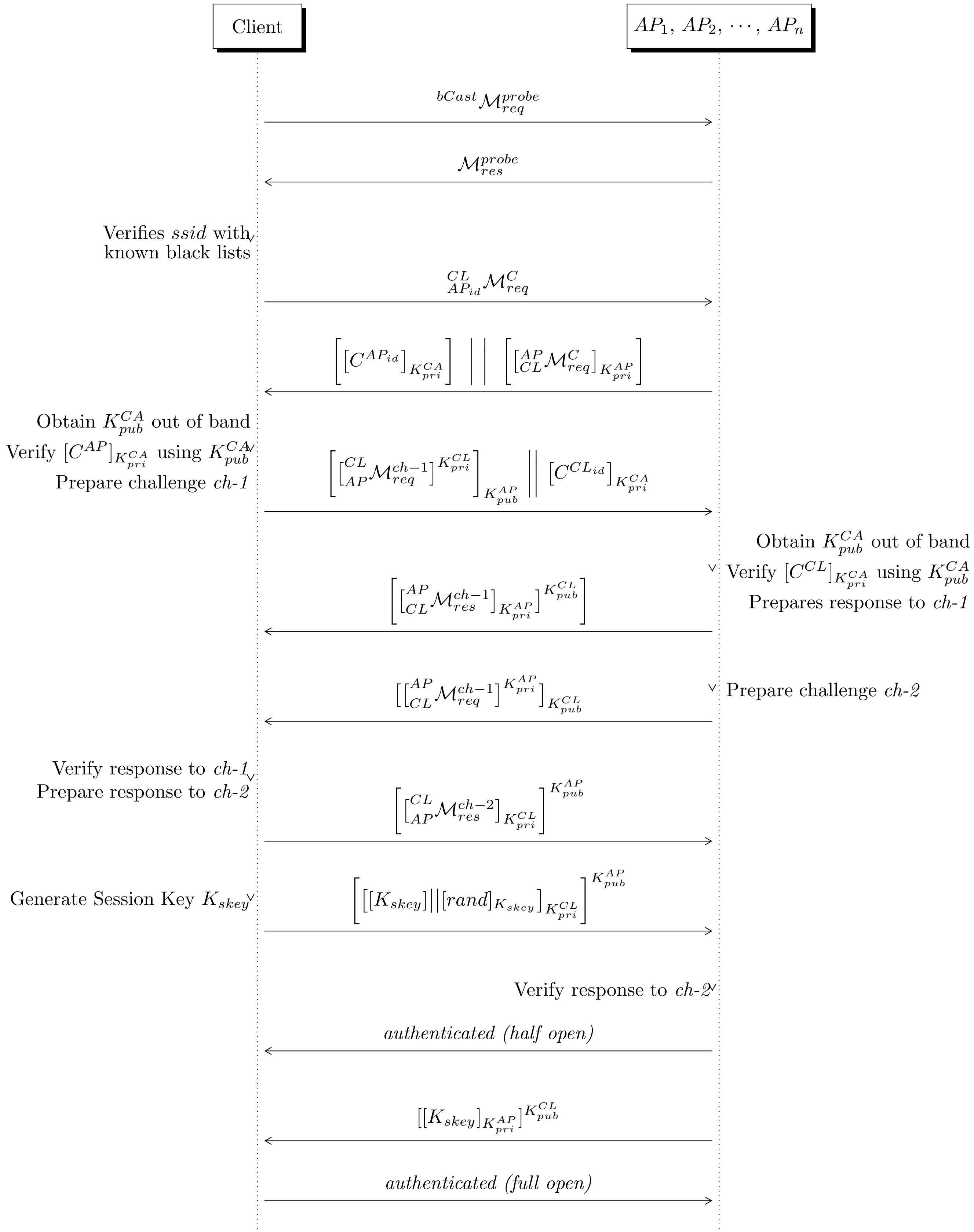
Client
$AP_1, AP_2, \cdots, AP_n$
${}^{bCast}\mathcal{M}^{probe}_{req}$
$\mathcal{M}^{probe}_{res}$
Verifies *ssid* with known black lists
${}^{CL}_{AP_{id}}\mathcal{M}^{C}_{req}$
$\left[[C^{AP_{id}}]_{K^{CA}_{pri}}\right] \,||\, \left[[{}^{AP}_{CL}\mathcal{M}^{C}_{req}]_{K^{AP}_{pri}}\right]$
Obtain K^{CA}_{pub} out of band
Verify $[C^{AP}]_{K^{CA}_{pri}}$ using K^{CA}_{pub}
Prepare challenge *ch-1*
$\left[[{}^{CL}_{AP}\mathcal{M}^{ch-1}_{req}]^{K^{CL}_{pri}}\right]_{K^{AP}_{pub}} \,||\, [C^{CL_{id}}]_{K^{CA}_{pri}}$
Obtain K^{CA}_{pub} out of band
Verify $[C^{CL}]_{K^{CA}_{pri}}$ using K^{CA}_{pub}
Prepares response to *ch-1*
$\left[[{}^{AP}_{CL}\mathcal{M}^{ch-1}_{res}]_{K^{AP}_{pri}}]^{K^{CL}_{pub}}\right]$
$[[{}^{AP}_{CL}\mathcal{M}^{ch-1}_{req}]^{K^{AP}_{pri}}]_{K^{CL}_{pub}}$
Prepare challenge *ch-2*
Verify response to *ch-1*
Prepare response to *ch-2*
$\left[[{}^{CL}_{AP}\mathcal{M}^{ch-2}_{res}]_{K^{CL}_{pri}}\right]^{K^{AP}_{pub}}$
Generate Session Key K_{skey}
$\left[[[K_{skey}]||[rand]_{K_{skey}}]_{K^{CL}_{pri}}\right]^{K^{AP}_{pub}}$
Verify response to *ch-2*
authenticated (half open)
$[[K_{skey}]_{K^{AP}_{pri}}]^{K^{CL}_{pub}}$
authenticated (full open)

Figure 2 *CL-AP* authentication handshake process using digital certificates.

Table 1 Contents in the special Digital Certificate issued to an *AP*.

Notation	Description	Notation	Description
C	Digital Certificate	t_s	Certificate validity *start*
t_e	Certificate validity *end*	$\mathcal{H}[\mathcal{C}]$	Hash of a certificate
t_{ca}	Time CA issued the certificate	$\mathcal{H}$	Hash Function
D_{type}	Device Type	$\mathcal{E}$	Encryption Function
D_{id}	Device ID	$\mathcal{D}$	Decryption Function
$K^{D_{id}}_{pub}$	Device public key	SP$_{name}$	Wi-Fi Service Provider Name
$\mathcal{C}^{D_{id}}$	Certificate issued to Device D_{id}	SP$_{id}$	Wi-Fi Service Provider ID
$R^{D_{id}}_{T_x}$	Receiver devices in transmission range T_x		

6.1 Certificate-based AP Authentication Protocol

In this section, we elaborate on the *AP* authentication process using digital certificates. Note that the certificate contains many simple attributes to validate identity and some special attributes to overcome security risks. The unique public key of the access point to which the certificate is issued helps keep track of the access point. In case of illegal or cyber-criminal activities, it helps trace back the involved *AP* and the individual or organization that obtained it. Another important attribute is the digital signature of the *CA*. This digital signature provides much-needed assurance to the user that the *AP* is secure and legitimate.

Discovery Phase. A wireless client device looks for all available APs (SSIDs) within its range. A client device may discover available APs and their corresponding security capabilities by either passively monitoring the Beacon frames or by actively probing every channel.

AP **Authentication Phase.** A wireless client device requests that each *AP* send its certificate. The request can be further refined to only APs with strong signals, meaningful SSIDs, etc. The user verifies the authenticity of the certificate submitted by each *AP* by invoking *Validate_Cert()* (see Algorithms 1 and 2). When *Validate_Cert()* completes successfully, the user will verify the information contained inside the certificate by invoking *Verify_Cert()* (see Algorithms 3). As shown in equation 4, the *AP*'s certificate has a special format with the following four additional critical pieces of information: "D_{type}," "D_{id}," "SP_{name}," and "SP_{id}," where D denotes a generic device compatible with 802.11 Wi-Fi networks, and SP denotes a service provider offering an 802.11 public Wi-Fi hotspot. A client device will reject any probe response frames from APs who do not have valid certificates in the specified special format.

$$\Big[\langle D_{type} \| \; D_{id}\rangle \; \| \langle SP_{name} \| \; SP_{id}\rangle \; \| \; \langle K^{D_{id}}_{pub}\rangle \; \| \\ \langle \mathcal{H}[\mathcal{C}^{D_{id}}]\rangle \; \| \; \langle \mathcal{E}, \mathcal{D}, \mathcal{H}\rangle \; \| \; \langle \; t_s \; \| \; t_e \; \| \; t_{issue} \; \rangle \Big] \quad (4)$$

Security Bootstrap Phase. Upon successful completion of *Verify_Cert()*, the user initiates a *Challenge-Response* (*C-R*) protocol in order to thwart any potential replay and MITM attacks. During the execution of the *C-R* protocol, the *AP* too can challenge the user to mitigate any potential user-centric attacks. Client device will respond to *AP*'s challenge only if *AP*'s successfully responds to the client challenge. This has been presented in Algorithms 4 and 5. If more than one authentic *AP* is available, then the client device can use any of the following parameters – *signal strength, service provider, service fees, etc.*, – to choose an *AP* to associate with. Depending on the implementation of the hotspot, the user may be redirected to a log-in/registration captive portal. Note that with the *VOUCH-AP* authentication mechanism, the client device will reject any *AP* that attempts to redirect the client to a *captive portal* before the successful completion of the *C-R protocol* during the

Procedure 1 Certificate Validation

```
Input: $K^{CA}_{Pub}$; $[C^{AP_{id}}]_{K^{CA}_{pri}}$
procedure VALIDATE_CERT(X, Y)
    $X \leftarrow K^{CA}_{Pub}$
    $Y \leftarrow [C^{AP_{id}}]_{K^{CA}_{pri}}$
    $result \leftarrow$ VERIFY_DIGISIG($X, Y$)
    if $result ==$ "Authentic'' then VERIFY_CERT(X, Y)
    else
        return Corrupt
    end if
end procedure
```

Procedure 2 Certificate Signature Verification

```
Input: $K^{CA}_{Pub}$; $[C^{AP_{id}}]_{K^{CA}_{pri}}$
procedure VERIFY_DIGISIG(X, Y)
    $X \leftarrow K^{CA}_{Pub}$
    $Y \leftarrow [C^{AP_{id}}]_{K^{CA}_{pri}}$
    if $Y$ decrypts without errors then $C^{M}_{id} \leftarrow Y \oplus X$
        VERIFY_CERT($[C^{M}_{id}]$)
    else return Corrupt Cert
    end if
end procedure
```

Procedure 3 Certificate Verification

```
Input: $C^{M}_{id}$
procedure VERIFY_CERT(X)
    $X \leftarrow C^{M}_{id}$
    $X.D_{type} == AP$                ▷ quit if anything else
    $X.D_{id} == ssid$                ▷ as advertised in $\mathcal{M}^{probe}_{res}$
    $X.SP_{name}$ is valid            ▷ can't be left blank
    $X.D_{id}$                        ▷ as advertised in $\mathcal{M}^{probe}_{res}$
    if all fields valid then return Authentic Cert
    else return Invalid Cert
    end if
end procedure
```

security bootstrap phase.

Session Key Establishment Phase. After successful completion of the security bootstrap phase, CL and AP are assured of a communication channel that is a secured communication link. It is well know that asymmetric (aka public key) cryptosystems are compute-intensive and will not suit the unique operational environment of 802.11 public hotspots. Therefore, computationally light symmetric encryption will be the ideal choice. However, these encryptions require a shared secret key or a session key which can be generated in one of two ways:

1. one end system (e.g., the client) generates the session key K_{skey} and transmits it to the the other end systems (e.g., the AP) using asymmetric encryption to ensure confidentiality and integrity;
2. both end systems contribute parts of the session key. The session key is then exchanged and combined by both end systems to generate the same unique session key, with exchanged information secured with asymmetric encryption.

The latter way of generating the key is typically achieved through an asymmetric key negotiation/exchange protocol such as *Diffie-Hellman.* This type of protocol depends on the mathematical property of *"discrete logarithm"* for its security. For simplicity, since the focus is on the security and privacy of user data, we assume that the client will generate K_{skey}, sign it K^{CL}_{pri}, and then encrypt it with K^{AP}_{pub} resulting in equation 5. Client also includes a time stamp identifying K_{skey}'s valid time window.

7 Analysis of *VOUCH-AP*

In this section, we analyze the security and privacy robustness of the proposed *VOUCH-AP* authentication mechanism. Note that, all through the paper, our discussion has focused solely on the security and privacy concerns that stem from the following:

- *The lack of a secure communication environment and asymmetry during the* 802.11 *low-level link layer authentication;*
- *The lack of security tools and techniques allowing a wireless client to authenticate the AP in a public hotspot.*

In light of the first concern, all advanced and sophisticated authentication mechanisms relying on WPA and WPA2 (802.11i) with 802.1X/EAP are futile since the very start of the process is vulnerable.

7.1 Snooping Attack (Packet Sniffing)

Attack Overview. To execute a snooping attack, the adversary will strategically position the malicious AP between the sender (client) and the receiver (AP). Such positioning enables him to snoop all the traffic being exchanged between the client and the AP. Wireless snooping attacks are not a very serious class of threats. However, during the 802.11 low-level authentication, all transmitted data are unencrypted, and therefore snooping proves to be a critical security and privacy threat.

Procedure 4 C-R Challenge Message

```
procedure Prep-CH-Request
    ${}^{D_1}\mathcal{M} \leftarrow challenger\ device$
    ${}_{D_2}\mathcal{M} \leftarrow responder\ device$
    $CH.request \leftarrow \Big[ [\ {}^{D_1}_{D_2}\mathcal{M}^{ch}_{req} \ || \ \langle t^{ch}_{start},\ t^{ch}_{end} \rangle\ ]_{K^{D_1}_{pri}} \Big]_{K^{D_2}_{pub}}$
return $CH.request$
end procedure
```

Procedure 5 C-R Response Message

```
procedure Prep-CH-Response
    ${}^{D_1}\mathcal{M} \leftarrow challenger\ device$
    ${}_{D_2}\mathcal{M} \leftarrow responder\ device$
    $CH.reply \leftarrow \Big[ [\ {}^{D_2}_{D_1}\mathcal{M}^{ch}_{res} \ || \ t^{ch}_{res}\ ]_{K^{D_2}_{pri}} \Big]_{K^{D_1}_{pub}}$
return $CH.reply$
end procedure
```

Attack Impact. Since unencrypted traffic is being sniffed, and it is very likely that the user will be submitting sensitive information to the captive portal, there is a glaring confidentiality breach directly impacting user privacy. Some of the more serious attacks include *accidental information disclosure, identity theft*, and *privacy leaks.*

$$CL \rightarrow AP :: \Big[\big[K_{skey} \ || \ [t^{start}_{skey}] \ || \ [t^{end}_{skey}]\ \big]_{K^{CL}_{pri}} \Big]^{K^{AP}_{pub}} \tag{5}$$

***VOUCH-AP*'s Defense.** In our proposed *VOUCH-AP* authentication mechanism, the adversary does not succeed in obtaining any useful information by intercepting unencrypted traffic for the following reasons:

- Unencrypted frames transmitted by the AP prior to the completion of the security bootstrap are only one of two possible frames listed below. Both types of frames are primarily for advertising an AP's SSID and capabilities –

  ${}^{AP}_{bcast}\mathcal{M}^{bcn}$ :: beacon frames transmitted by an AP to all clients in its transmission range; or

  ${}^{AP}_{CL}\mathcal{M}^{probe}_{res}$ :: the message from AP in response to the client's probe request.

- Unencrypted frames transmitted by a client prior to the successful completion of the security bootstrap phase are the network discovery probe requests seeking SSIDs and the capabilities of all available *AP*s.

- *VOUCH-AP* is proposed to replace the vulnerable 802.11 low-level authentication with a secure link layer asymmetric channel that can be further leveraged to establish a secure symmetric key session.

None of the information contained in the *probe frames* or the *beacon frames* is sensitive or critically privacy-revealing. This information can also be obtained by the adversary just as easily as any other client device, and snooping is counter-intuitive in this operational scenario.

7.2 Replay Attacks – Packet, Certificate, Session

Attack Overview. The attacker can choose to replay select packets or the entire stream containing the certificates exchanged. This can also include the authentication credentials and/or financial information submitted by the client to the captive portal upon redirection from the *AP*.

Attack Impact. The impact of a replay attack is significant if successfully executed in an open access 802.11 low-level "open key" authentication in public hotspots with unsecured *AP*s. Some of the more serious attacks include *session hijacking*, *session replay*, *replay of authentication packets*, and *replay of certificate packets*.

***VOUCH-AP*'s Defense to Replay Attacks.** The communication between the client's wireless device and an *AP* in a public hotspot is secure from information disclosure attacks that compromise the confidentiality and privacy of the client. All data exchanged over an unsecured channel, prior to completion of the security bootstrap phase, are primarily non-private in nature and can be easily obtained by anyone simply by using one of many available free tools.

Even if an SSH/VPN tunnel is established for securing the communication, since the 802.11 low-level open authentication is vulnerable to interception and inference attacks, there are no security guarantees for the VPN tunnel.

Lemma 3 *Users can detect evil-twin APs at* 802.11 *public hotspots that are positioned as MITM attempting replay attacks.*

Proof 3 *User transmits a probe request during the discovery phase and potentially receive probe responses from multiple AP's. The client then requests the AP(s) for their digital certificate. An AP responds to the client's request by forwarding its certificate signed by the issuing CA's private key, as shown in equation 6.*

$$AP \xrightarrow{[C^{AP_{id}}]_{K^{CA}_{pri}}} CL \tag{6}$$

Assuming there exists an AP with a certificate issued by a CA that the user trusts, the user will verify the certificate, especially the validity period of the certificate, D_{type}, and D_{id}. The D_{id} in the cert should be same as the SSID advertised by the corresponding AP in its probe response and/or beacon frames.

During the security bootstrap phase, any information relayed from CL to AP is first encrypted with the CL's private key K^{CL}_{pri} (this encryption is optional) and then enciphered with the AP's public key – K^{AP}_{pub}. This two-layered encryption is a critical security requirement because encrypting messages with K^{CL}_{pri} provides integrity and non-repudiation services, whereas encrypting messages with K^{AP}_{pub} provides confidentiality and privacy services. Note that even with such security measures, messages are vulnerable to inference attacks. Inference attacks try to deduce the source and destination of an intercepted message and further try to guess the type of information being exchanged.

Similarly, during the security bootstrap phase, any information relayed from AP to CL is first enciphered with K^{AP}_{pri} (this encryption is optional) and then enciphered using K^{CL}_{pri}. Both the CL and the AP obtain each other's public key from the certificates exchanged. The public key of the certificate owner is included in the certificate issued by the CA. Additionally, the certificate itself is encrypted with the CA's private key K^{CA}_{pri}. Therefore, anyone can verify the authenticity of the certificate using the CA's public key K^{CA}_{pub}. This process has been presented in detail in Algorithm 2.

Now, the client has access to AP's public key K^{AP}_{pub} and uses it to encrypt all subsequent messages. Hypothetically, if the adversary were to intercept this communication and get a copy of the certificate $C^{AP_{id}}$, there is very little benefit to the adversary. He can certainly decrypt the certificate using K^{CA}_{pub}, but all the information within the certificate is non-private in nature.

With the certificate encrypted with K^{CA}_{pri}, the adversary cannot modify the certificate contents, say for instance, to falsely bind his public key to the AP and replay the certificate at a later time. The process of certificate validation is presented in Algorithm 1.

Additionally, to prevent replay attacks, the client will send a challenge ch as shown in equation 7:

$$CL \xrightarrow{[[\mathcal{M}^{ch}_{req}]]_{K^{AP}_{pub}}} AP \tag{7}$$

For added security against replay attacks, client can send the challenge in the form shown in equation 8:

$$CL \xrightarrow{\left[[\mathcal{M}^{ch}_{req}]_{K^{CL}_{pri}}\right]_{K^{AP}_{pub}}} AP \tag{8}$$

If the above transmission is a certificate replay attack from an attacker-controlled evil-twin, then the attacker should not be able to respond to the challenge because the attacker cannot access K^{AP}_{pri}. This can be summarized as follows: there cannot be a situation with two AP's –

$AP_i \in \langle h_p, l_q\rangle$ and $AP_j \in \langle h_p, l_q\rangle$ which can be formally presented as shown in equation 9:

$$K^{AP_i}_{pub} == K^{AP_j}_{pub} \ for \ i \neq j \tag{9}$$

AP will respond to the challenge with another challenge-response message encrypted as shown in equation 10:

$$CL \xleftarrow{[\mathcal{M}^{ch}_{res}]_{K^{CL}_{pub}}} AP \tag{10}$$

For added security against replay attacks, AP's response message – to CL's challenge (equation 7) – will be in the form as shown in equation 11:

$$CL \xleftarrow{\left[[\mathcal{M}^{ch}_{res}]_{K^{AP}_{pri}}\right]_{K^{CL}_{pub}}} AP \tag{11}$$

The attacker can still try to guess the response if he has apriori knowledge about the type/format/size of response message expected by the requester. However, this is not a realistic attack vector.

Lemma 4 *An attacker's effort toward a session replay attack can be thwarted effectively under the* VOUCH-AP *authentication mechanism.*

Proof 4 *Session replay attacks can be thwarted by exchanging messages as presented in equation 5. The session key along with the start and end times of the valid corresponding session will enable the AP to readily detect if its is replayed by a MITM. Furthermore, since the message is first digitally signed by the client and then encrypted with the AP's public key, the message also provides source anonymity. This also prevents the attacker from replacing the signed message with a message that the attacker signs since the message is encrypted at the outermost layer using AP's public key.*

7.3 Man-in-the-middle Attacks

Attack Overview. In a MITM attack, the attacker strategically positions himself (and/or his attack hardware/software) between the CL and the AP as follows: $CL \leftrightarrow AP_{et} \leftrightarrow AP_{leg}$.

With such positioning, the adversary can intercept messages traveling in either direction. After intercepting the communication between the CL and the AP, the adversary can do several things:

- *Relay packets*: In this case, the adversary acts simply as a router relaying the packets. He can certainly snoop the information being relayed, but attacker does not gain much useful information other than knowing what is inside the certificate.
- *Delay or Drop packets*: The adversary does not gain anything from either delaying or dropping the packets other than causing a temporary DoS attack. Such temporary DoS attacks are almost impossible to thwart.
- *Modify packets*: Adversary cannot successfully execute packet modification attack since he cannot access the private key used by the end parties to digitally sign the packets. The adversary can still modify packets using brute-force techniques but such modifications will be readily detected with integrity verification mechanisms in place on the end systems.

Attack Impact. The type and extent of a MITM attack depends on how the adversary leverages his position and the intercepted communication against the system and its components. At the very least, a sophisticated attacker can succeed in stealing cookies, hijacking sessions, active snooping, passive snooping, masquerading, and capturing cryptographic keys.

***VOUCH-AP*'s Defense.** The proposed Certificate-based robust AP authentication mechanism effectively thwarts MITM attacks. We evaluate the security robustness of the solution in the following lemma.

Lemma 5 VOUCH-AP *effectively counters an attackers attempt to execute a MITM attack successfully by luring client devices to connect his"evil-twin" AP at a public hotspot.*

Proof 5 *For the attacker to succeed as a MITM, he has to succeed in either of the following:*

1. *replace the target AP's public key K^{AP}_{pub} with another public key $K^{AP'}_{pub}$ for which he has the corresponding private key $K^{AP'}_{pri}$;*

OR

2. *replace the target AP's certificate C^{AP} with an alternate certificate $C^{AP'}$ binding $K^{AP'}_{pub}$ to that certificate owner.*

Replacing K^{AP}_{pub} is not an option since the key is bound to the legitimate AP in the certificate and encrypted with the issuing CA's private key K^{CA}_{pri} (known only to the issuing CA). Similarly, obtaining a legitimate "special" certificate from a legitimate CA that identifies the attacker as a Wi-Fi Service Provider (SP) binding the public key $K^{AP'}_{pub}$ to the attacker is impossible without a colluding malicious CA, which is beyond the scope of this paper.

Another way for this scenario to occur is if a legitimate AP turns rogue due to node capture attack, which again, is beyond the scope of this paper. The previous two scenarios in which the CA or AP is rogue, are manifestations of insider threats that cannot be addressed explicitly with the proposed solution.

Note that an AP that is not in possession of a valid and legitimate certificate gets isolated by the CL, who will choose a certified AP to connect to the hotspot. At best, the adversary can sniff the traffic by relaying it, but he gains nothing by doing so. This traffic can

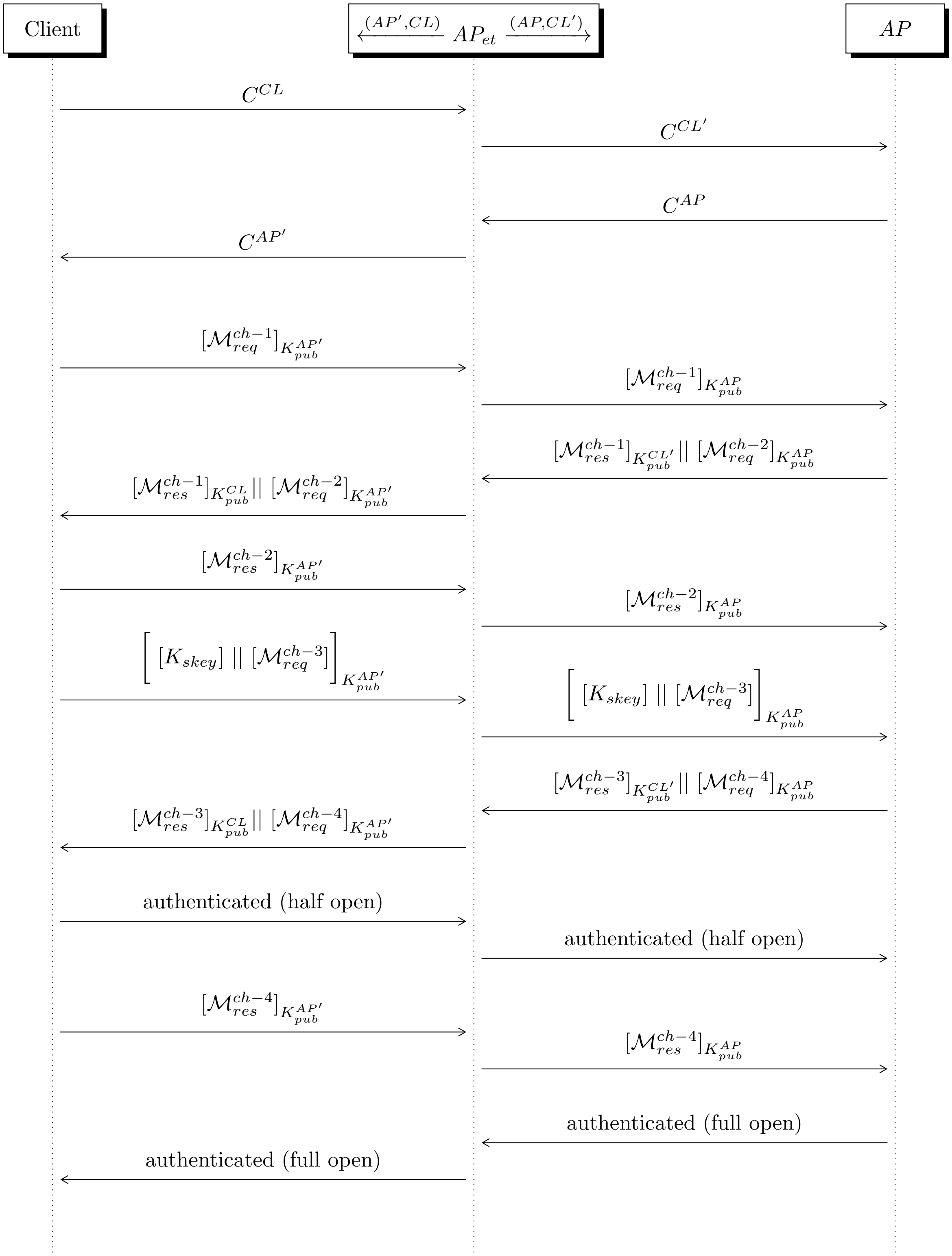
Client
$\xleftarrow{(AP',CL)} AP_{et} \xrightarrow{(AP,CL')}$
AP
C^{CL}
$C^{CL'}$
C^{AP}
$C^{AP'}$
$[\mathcal{M}^{ch-1}_{req}]_{K^{AP'}_{pub}}$
$[\mathcal{M}^{ch-1}_{req}]_{K^{AP}_{pub}}$
$[\mathcal{M}^{ch-1}_{res}]_{K^{CL'}_{pub}} || \ [\mathcal{M}^{ch-2}_{req}]_{K^{AP}_{pub}}$
$[\mathcal{M}^{ch-1}_{res}]_{K^{CL}_{pub}} || \ [\mathcal{M}^{ch-2}_{req}]_{K^{AP'}_{pub}}$
$[\mathcal{M}^{ch-2}_{res}]_{K^{AP'}_{pub}}$
$[\mathcal{M}^{ch-2}_{res}]_{K^{AP}_{pub}}$
$\Big[[K_{skey}] \ || \ [\mathcal{M}^{ch-3}_{req}] \Big]_{K^{AP'}_{pub}}$
$\Big[[K_{skey}] \ || \ [\mathcal{M}^{ch-3}_{req}] \Big]_{K^{AP}_{pub}}$
$[\mathcal{M}^{ch-3}_{res}]_{K^{CL'}_{pub}} || \ [\mathcal{M}^{ch-4}_{req}]_{K^{AP}_{pub}}$
$[\mathcal{M}^{ch-3}_{res}]_{K^{CL}_{pub}} || \ [\mathcal{M}^{ch-4}_{req}]_{K^{AP'}_{pub}}$
authenticated (half open)
authenticated (half open)
$[\mathcal{M}^{ch-4}_{res}]_{K^{AP'}_{pub}}$
$[\mathcal{M}^{ch-4}_{res}]_{K^{AP}_{pub}}$
authenticated (full open)
authenticated (full open)

Figure 3 Illustration of MITM attack scenario in a 802.11 hotspot with co-located "evil-twin" APs.

Table 2 Cryptographic keys used in 802.11 hotspots for different security services.

Source	Destination	Confidentiality	Authentication/ Non-repudiation
CL	AP	K_{pub}^{AP}	K_{pri}^{CL}
AP	CL	K_{pub}^{CL}	K_{pri}^{AP}
CA	AP	K_{pub}^{AP}	K_{pri}^{CA}
AP	CA	K_{pub}^{CA}	K_{pri}^{AP}

be protected against sniffing by encrypting it with a session key K_{skey} that the CL and AP can negotiate as previously discussed.

Furthermore, the adversary cannot launch a "packet modification attack" since every packet is encrypted before transmissions – as soon as the CL and AP exchange their certificates. The encryption key used for message encryption depends on the packet type as well as its source and destination. We have listed in Table 2 the encryption key(s) that will be used for encrypting various packets in our proposed VOUCH-AP*.*

When the CL requests AP for its certificate, the AP responds by sending the certificate C^{AP} issued by CA. C^{AP} should identify the AP as a hotspot service provider and must be encrypted with K_{pri}^{CA}. Optionally, the AP can also include a hash of the certificate $\mathcal{H}(C^{AP})$ in its response message, along with the certificate. Inclusion of $\mathcal{H}(C^{AP})$ helps detect any integrity attacks on the certificate. Finally, for providing non-repudiation service, AP encrypts the hash with K_{pri}^{AP}.

On receiving the C^{AP}, the CL will first verify the validity of the certificate as presented in Algorithm 1. If the certificate is valid, then the CL decrypts the certificate using K_{pub}^{CA}. Then the CL extracts K_{pub}^{AP} from C^{AP}. If AP has attached a hash of the certificate ($\mathcal{H}_{C^{AP}}$), then the CL computes the hash of the C^{AP}, decrypts it with K_{pub}^{CA} and then compares the two hash values. If the two hash values match, then CL is assured of the AP's identity and authenticity as a public hotspot service provider.

$$CL \xrightarrow{\left[\left[K_{skey}\middle|\middle|\langle t_s,\ t_e\rangle\middle|\middle|[\mathcal{M}_{req}^{ch-2}]_{K_{skey}}\right]_{K_{pri}^{CL}}\right]_{K_{pub}^{AP}}} AP \tag{12}$$

$$CL \xleftarrow{\left[\left[\mathcal{M}_{res}^{ch-2}\middle|\middle|\mathcal{H}(\mathcal{M}_{res}^{ch-2})\middle|\middle|\langle t_s,\ t_e\rangle\right]_{K_{pri}^{AP}}\right]_{K_{pub}^{CL}}} AP \tag{13}$$

Now, the user will generate a random session key K_{skey}, which he will encrypt with the AP's public key. In this message, the CL will embed a second challenge message encrypted using K_{skey}. Finally, CL will also attach a signed hash of the session key within the encrypted message (see equation 12).

On receiving this message, the AP will extract the session key, decrypt the challenge, and then respond with another encrypted message. The response message from the AP is first encrypted using the AP's private key and then with CL's public key. The message also contains a hash of the encrypted response, and a valid time window with start (t_s) and end times (t_e), as shown in equation 13.

At this point, a shared secret key has been successfully negotiated between the CL and the AP, and all messages hereinafter will be encrypted with the session key K_{skey} until the end of the session at t_e. Alternately, upon verification of the AP, the client can also supply a password associated with the service it is trying to access.

7.3.1 Countering evil-twin Attacks

The danger of evil-twin attacks can be eliminated by requiring remote clients to establish VPN connections with VPN gateways prior to gaining access to network resources. However, remote access VPN connection setup requires a pre-shared key on the client and VPN gateways. This pre-shared key is never transmitted during authentication, which defeats the evil-twin's ability to copy credentials and key information. This helps improve the security of wireless networks by protecting vulnerable management frames that are the root cause for many wireless DoS attacks.

8 Related Work

Commonly used identifiers for IEEE 802.11 APs, such as network name (SSID), MAC (BSSID), or IP address can be trivially spoofed. Impersonating existing APs with faked ones to attract their traffic is referred to as the *evil-twin attack*. It allows an attacker with little effort and expenditure to fake a genuine AP and intercept, collect, or alter data [9].

One existing work that attempts to filter unknown/malicious APs during automatic device association is presented in [5]. In this work, authors present a scheme that establishes a secure wireless connection between a client device and an AP in open 802.11 environments using hierarchical identity-based cryptography. However, in their approach, each user makes use of the device's MAC address as its public key, which is a critical security flaw given the ease with which MAC addresses can be spoofed. Furthermore, there are no empirical validation results or security analyses of their proposed approach to confirm their claims of achieving confidentiality and integrity even in the presence of colluding attackers.

Internet of Things' (IoT) nodes, like any other computing node, require connectivity to the network/Internet. The node may either connect directly to the network or through intervening IoT node(s). Perhaps, as one can imagine, there is a ready solution in the form of a WiFi access point. In this context, in [11], authors argue that the inherent vulnerabilities of the internet make it rather critical to address security and privacy issues before the IoT is widely deployed. They note that authentication and access control are two key techniques to prevent a computer or network component from being compromised. To this aim, their work analyzes existing authentication and access control solutions, and present a feasible design for IoT applications.

Al-Salihy and Samsudin [2] have proposed a new protocol of routers CA Certificate. According to their work, the router would be certified by a Certificate Authority by verifying router physically about the information given by the administrator. Then this router will be allowed to issue sub Certificates to the valid end nodes by keeping track of their MAC addresses so that these sub Certificates would help to overcome the replay, man-in-the-middle, and denial-of-service attacks. However, their work does not consider the ease with which such sub sertificates can be fabricated. Chen and Ito [3] have proposed using "End-to-Middle" security to protect against evil-twin APs. Their proposed model is a *end-to-middle* security to create a secure gateway on the internet that can be reached by mobile users. The user have to establish a secure channel with this virtual gateway so that all the user traffic relay through the gateway to the Internet.

Greenstein et al. [8] present the design and evaluation of an 802.11-like wireless link layer protocol – SlyFi – that obfuscates all transmitted bits to increase privacy. They show that SlyFi is nearly as efficient as existing schemes such as WPA for discovery, link setup, and data delivery despite its heightened protections; transmission requires only symmetric key encryption and reception requires a table lookup followed by symmetric key decryption, with a slight overhead introduced in packet delivery compared to WPA-CCMP encryption. With SlyFi, authors assume that clients and services have (possibly shared) cryptographic keys prior to communication. Authors also assume that most private services will be known beforehand such as a home 802.11 AP and can bootstrap keys using these methods. However, given the unique operational environment of 802.11 public Wi-Fi hotspots with unknown and unsecured APs, a protocol like SlyFi are not suitable.

In [10], authors propose an efficient two-factor localized authentication scheme for inter-domain handover and roaming in IEEE 802.11 based service-oriented wireless mesh networks. Their solution addresses some important aspects, such as resource-constrained Mobile Stations (MSs) and the ping-pong movement phenomenon during handover roaming across different hotspots. Cheng et al. [4] examine the privacy leakage in public hotspots from activities such as domain name querying, web browsing, search engine querying and online advertising. We discover that, from these activities multiple categories of user privacy can be leaked, such as identity privacy, location privacy, financial privacy, social privacy and personal privacy. Authors use real data from 20 airport data sets in four countries and discover that over two thirds (68%) of users leak private information while accessing Internet at airports.

Gonzales et al. [7] propose defense mechanisms against evil-twin attacks in three distinct stages. First, they present "context-leashing" an evil-twin detection strategy. The proposed technique constrains an AP's trust based on its location. Second, they propose identifying wireless networks using un-certified public keys and design an SSH-style authentication and session key establishment protocol to be compatible with 802.1X standard. Lastly, to mitigate the pitfalls of SSH-type authentication, they propose a crowd-sourcing-based reporting protocol that provides historical information for AP public keys while preserving the location privacy of users who contribute reports.

Service provider such as *AT&T* and *Verizon* often own and administer a large number of public 802.11 hotspots commercially. They also administer hotspots that are owned by other service providers. Such hotspots are commercially available for users with guaranteed secure access through WPA2 variants. However, these hotspots that guarantee security are available only to registered and subscribed users. Users who do not subscribe to that service provider will not have a pre-shared key with that provider. Hence, users connecting to an AP and subscribing to a service other than the one locally administrating the service cannot establish a secure connection to that AP with link layer security [12].

9 Conclusions

Open-access 802.11 public Wi-Fi networks have become a necessity in today's cyber-centric world where citizens feel the need to intermittently access the Internet. 802.11 public hotspots are easy to deploy. They have with little overhead, and most importantly, they require no out-of-band key exchange or prior trust relationships. However, the very same open-access 802.11 networks can become a nightmare if the adversary can successfully exploit the network's inherent vulnerabilities. One key attack vector adopted in the hacker community is deploying a malicious access point commonly referred to as an *"evil-twin."*

This paper presents a new approach for preserving the privacy of users accessing public hotspots through an AP that only provides 80.11 low-level authentication as un-encrypted text. Existing solutions primarily focus on securing the AP, but the client has to blindly rely on the information it receives from a broadcast beacon frame or a probe response frame to its request. The proposed AP authentication mechanism – *VOUCH-AP* – resolves this information asymmetry by enabling the user to authenticate an AP before authenticating and associating with it. Existing research predominantly focus on user authentication by AP, and detection and isolation of evil-twin AP-based attacks. The proposed approach, to the best of our knowledge, is the first work on the authentication of APs.

10 Definitions

In this sections, we provide formal definitions of important technical terms used in this paper.

Definition 1 *A wireless Access Point (AP) is a device through which wireless devices connect to a wired network, using Wi-Fi or other related wireless standards.*

Definition 2 *Information Asymmetry is a term from the field of "Contract Theory and Economics" that deals with the study of decisions in transactions where one party has more or better information than the other.*

Definition 3 *Service Set Identifier (SSID) is the name assigned to a wireless local area network (WLAN) gateway. This is also known as the AP.*

Definition 4 *BSSID is the MAC address of the wireless AP generated by combining the* 24 *bit Organization Unique Identifier and the manufacturer's assigned* 24*-bit ID for the radio chipset in the wireless AP.*

Definition 5 *A Media Access Control address (MAC address) is a unique identifier assigned to network interfaces for communications at the data link layer of a network segment. It is also known as hardware address.*

Definition 6 *A captive portal is a special purpose web page, which is often a login/registration page, that is presented to the user via the browser before (s)he can use the Internet.*

Definition 7 *Digital certificates are electronic credentials that bind the identity of a certificate owner to a pair (public and private) of electronic keys that can be used to encrypt and sign information digitally.*

Definition 8 *Certificate Authority (CA) in cryptography is an entity that issues digital certificates that certify the ownership of a public key by the named subject of the certificate.*

Definition 9 *A Certificate Revocation List (CRL) – in crypto systems such as PKI – is a list of certificates that have been revoked. Entities presenting those revoked certificates will no longer be trusted.*

Definition 10 *A hotspot is a physical location (geographic location with coordinates) where people can access the Internet through* 802.11 *WLAN connected to an Internet service provider backbone.*

Definition 11 *A public key infrastructure (PKI) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store, and revoke Digital Certificates and to manage public-key cryptosystems.*

Definition 12 *The digital signature of a message or a document is generated by encrypting the item with the sender's private key so that it can be verified by anyone with access to the sender's public key. This is a fundamental requirement for non-repudiation service.*

Definition 13 *A pre-shared key (PSK) is a secret shared between two or more parties. Ordinarily, the sharing or negotiation of the shared secret is performed out-of-band, preferably using secure means. The shared secret must be available to all parties ahead of time.*

Definition 14 *A private key is a cryptographic key that is meant to be a secret and should never be shared with anyone. No one other than the owner of the key should ever have any knowledge of the private key.*

Definition 15 *A public key is a cryptographic key that is meant to be publicized and shared with anyone who intends to communicate or conduct an electronic transaction with another individual/system/service.*

Definition 16 *Authentication is the process to determine whether someone is who (s)he claims to be or something is what it is declared to be. With* 802.11 *hotspots, it is the process of verifying the credentials of a client desiring to join the hotspot.*

Definition 17 *Association is the service used to establish* access point/client *mapping and enable client invocation of distribution system services.*

Definition 18 *Non-repudiation refers to a state of affairs where the author of a statement will not be able to successfully challenge authorship of the statement or validity of an associated contract.*

References

[1] Wi-fi, the new basis of maslow's pyramid of human needs?". February 2016.

[2] W. A. Al-Salihy and A. Samsudin. A new proposed protocol of router's ca certificate. In *2006 International Conference on Computing & Informatics*, pages 1–6. IEEE, 2006.

[3] E. Y. Chen and M. Ito. Using end-to-middle security to protect against evil twin access points. In *2009 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks Workshops*, pages 1–6, June 2009.

[4] N. Cheng, X. O. Wang, W. Cheng, P. Mohapatra, and A. Seneviratne. Characterizing privacy leakage of public wifi networks for users on travel. In *INFOCOM, 2013 Proceedings IEEE*, pages 2769–2777. IEEE, 2013.

[5] J. Choi, S. Y. Chang, D. Ko, and Y. C. Hu. Secure mac-layer protocol for captive portals in wireless hotspots. In *2011 IEEE International Conference on Communications (ICC)*, pages 1–5, June 2011.

[6] N. Do, Y. Zhao, C.-H. Hsu, and N. Venkatasubramanian. Crowdsourced mobile data transfer with delay bound. *ACM Trans. Internet Technol.*, 16(4):28:1–28:29, Dec. 2016.

[7] H. Gonzales, K. Bauer, J. Lindqvist, D. McCoy, and D. Sicker. Practical defenses for evil twin attacks in 802.11. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–6. IEEE, 2010.

[8] B. Greenstein, D. McCoy, J. Pang, T. Kohno, S. Seshan, and D. Wetherall. Improving wireless privacy with an identifier-free link layer protocol. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, MobiSys '08, pages 40–53. ACM, 2008.

[9] F. Lanze, A. Panchenko, I. Ponce-Alcaide, and T. Engel. Undesired relatives: protection mechanisms against the evil twin attack in ieee 802.11. In *Proceedings of the 10th ACM symposium on QoS and security for wireless and mobile networks*, pages 87–94. ACM, 2014.

[10] X. Lin, X. Ling, H. Zhu, P.-H. Ho, and X. S. Shen. A novel localised authentication scheme in ieee 802.11 based wireless mesh networks. *International Journal of Security and Networks*, 3(2):122–132, 2008.

[11] J. Liu, Y. Xiao, and C. P. Chen. Internet of things; authentication and access control. *Int. J. Secur. Netw.*, 7(4):228–241, Apr. 2012.

[12] B. Potter. Wireless hotspots: petri dish of wireless security. *Communications of the ACM*, 49(6):50–56, 2006.

[13] S. Siboni, A. Shabtai, N. O. Tippenhauer, J. Lee, and Y. Elovici. Advanced security testbed framework for wearable iot devices. *ACM Trans. Internet Technol.*, 16(4):26:1–26:25, Dec. 2016.