This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2011 proceedings

Secure MAC-Layer Protocol for Captive Portals in Wireless Hotspots

Jihyuk Choi*, Sang-Yoon Chang*, Diko Ko†, Yih-Chun Hu*
*Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, USA
{jchoi43, chang6, yihchun}@illinois.edu
†School of Computer Science & Engineering
Seoul National University
Seoul, Korea
diko@mmlab.snu.ac.kr

***Abstract*— Wireless access points largely fall into three categories: home and small business networks, enterprise networks, and hotspots. Wi-Fi Protected Access (WPA) provides solutions to home, small business, and enterprise networks, but hotspots typically are not secured at the Medium Access Control (MAC) layer because they are open to the public. In this paper, we present a scheme that establishes a secure wireless connection between a client device and an access point in these open environments. In our approach, we use hierarchical identity-based cryptography, and each user uses its MAC address as its public key. Our scheme ensures confidentiality and integrity even in the presence of colluding attackers.**

I. Introduction

IEEE 802.11 wireless networks are widely deployed today for governmental, commercial, and personal uses. As the demand for wireless communication increases, securing wireless systems against malicious behavior has taken on increasing importance. Previous protocols have been designed to use a shared key or an authentication server to provide link confidentiality, including Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), and IEEE 802.11i (also called WPA2). Though WPA and WPA2 are quite robust in environments where the base station and client can share a key, such as home, small business, and enterprise settings, they cannot provide the same properties in open hotspot environments because of the lack of a properly shared secret key. In particular, in an open and public environment, such as coffee shops, bookstores and restaurants, a single pre-shared key may be known by an attacker, and individual shared secrets are difficult to distribute in a small and public environment.

Some hotspots are owned or administered by a service provider such as AT&T that also owns and administers many other access points. In these environments, a subscriber to that service provider may be able to use WPA2-Enterprise to gain secure access. However, users that do not subscribe to that service provider do not have a shared key with that provider. Furthermore, unlike in cellular systems, the various service providers do not have a common roaming mechanism [1]. Thus a user that connects to an access point but subscribes to a service other than locally administrating service cannot establish a secure MAC layer connection to that access point. A final problem with trying to deploy WPA2-Enterprise in a hotspot environment is that ease of use and configuration is key to a successful deployment [2]. In general, due to the abilities of users, the large number of Wi-Fi service providers, and the inherent need for open access, existing solutions to MAC-layer security in Wi-Fi are not applicable to many deployments of commercial hotspot service.

Due to the lack of applicable MAC-layer security solutions, current 802.11-based hotspots choose one of two security strategies. The first strategy is to use no security whatsoever, so that any user can connect directly to the Internet. Small coffee shops often use this strategy, sometimes in combination with a single WEP key that is distributed to all of the customers of that coffee shop. The second strategy is to use a captive portal. A captive portal is a router or a gateway host that will not allow traffic to pass until a user has authenticated himself [3]. In a captive portal environment, a client device acquires an Internet Protocol (IP) address using Dynamic Host Configuration Protocol (DHCP) and any web request from the client device is redirected to the captive portal. The captive portal presents a web page, the user authenticates himself to the web page, possibly paying an access fee, the portal stops redirecting that client's traffic, so the client can now access the rest of the Internet.

In this paper, we study the security of public wireless hotspots that use captive portals. Our techniques are also applicable to other environments, but in this paper, we focus on their use in captive portal environments. We aim to address the problem that captive portals encrypt only the authentication phase, where the user supplies login credentials or other payment data, and transmits user data in the clear [3]. As a result, many service providers recommend that their users use a Virtual Private Network (VPN) to secure their traffic, and some previous research attempts to use VPN to develop a secure public hotspot service [4]. However, not all people are able to use VPN. In addition, these approaches attempt to solve a MAC-layer problem at the network layer, and are fundamentally unable to address attacks at the MAC layer [3].

One important property for a MAC-layer security mechanisms is that they must be resilient to attacker *collusion*. In a collusion attack, attackers share information in an attempt to break the security of a victim node. In a commodity wireless system, an attacker can easily purchase a large number of wireless network interfaces and access points, increasing the

This material is based upon work partially supported by USARO under Contract No. W-911-NF-0710287 and the NSF under Grant No. CNS-0953600.

978-1-61284-233-2/11/$26.00 ©2011 IEEE

importance of collusion-resistant protocols.

We design a protocol that allows a client to establish a secure connection with an access point in the presence of malicious entities. When a client connects to an access point, our protocol provides a secure authentication and key exchange process. Our scheme constructs a protocol for establishing a secure connection on top of hierarchical identity-based cryptography [5]. Our scheme is scalable, easy to deploy, and provides secure authentication of both the client user and the access point, and is resistant to attacker collusion.

The rest of the paper is organized as follows: in Section II we review some related work. In Section III we overview the hierarchical identity-based cryptography on which the key distribution of our proposed protocol based. We detail the proposed scheme in Section IV and conclude this paper in Section V.

II. Related Work

The IEEE 802.11 standard is widely deployed because of its convenience and lower cost. Wireless network interfaces implementing the standard operate in unlicensed spectrum, and as such are not subject to licensing fees, the interfaces are built for the mass market and thus easy to set up and use, and due to economies of scale, these network interface devices tend to be inexpensive.

The widespread deployment of 802.11, together with the relative ease of gaining access to an 802.11 network, has attracted attention to security concerns. The original IEEE 802.11 standard defined the Wired Equivalent Privacy (WEP) security protocol. The design goal of WEP is to provide the same level of security as a wired network. Fluhrer et al. showed that RC4, the encryption algorithm used by WEP, can be broken, compromising the session key [6]. Further attacks against WEP exploited both the encryption mechanism [7], [8] and the authentication scheme [9].

Subsequent protocols, such as Wi-Fi Protected Access (WPA) and IEEE 802.11i (also called WPA2), have been proposed to address the exploitable vulnerabilities of WEP. As described in Section I, WPA and WPA2 are not well-suited to the hotspot environment because of the difficulty of having pre-arranged authentication information or a secret pre-shared key. In particular, establishing such accounts or keys requires additional effort on the part of subscribers, which conflicts with the provider's business goal of increasing subscription [2]. To make the setup more convenient, the Wi-Fi Protected Setup scheme provides three different setup methods: the Push-Button method, the PIN method, and an Out-of-band channel [10]. Table I shows how Wi-Fi Protected Setup makes the configuration procedure more user-friendly. Though Wi-Fi Protected Setup can simplify setup in a home or small office environment, it does not solve the problem in wireless hotspots, because it is infeasible for a user to gain physical access to the access point, as needed for all three Wi-Fi Protected Setup options.

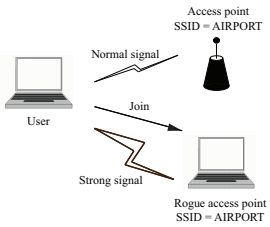

Fig. 1. Rogue access point attack

Because of the challenges of using WPA and WPA2 in hotspot environments, typical hotspot deployments use a captive portal. Captive portals use Secure Sockets Layer (SSL) to encrypt authentication and payment data, but afterwards passively route cleartext traffic to and from destination sites [3]. Previous work on securing data through a captive portal focus on network-layer approaches [11], [12]. However, many security vulnerabilities exist at the Medium Access Control (MAC) layer, such as MAC address spoofing, are fundamentally unsolvable at a higher layer. As a result, we address the MAC-layer security issues in a captive portal environment.

Yang et al. proposed the method of providing the secure connection on MAC-layer in wireless hotspot [13]. The suggested solution, dummy authentication key establishment, is based on WPA-Pre-Shared Key (PSK). Here dummy means that the user does not exactly authenticate. The proposed protocol provides link-layer data encryption in wireless hotspots and separate session encryption keys for different users.

Hole196 [14], a vulnerability in the WPA2, was recently reported. It is an insider attack which can be carried out by the attacker. The WPA2 defines two types of keys for data encryption; Pairwise Transient Key (PTK) and Group Temporal Key (GTK). While PTK is used to encrypt unicast data frames (e.g. user data), GTK is for the encryption of group addressed data frames (e.g. ARP request). Here, only an access point is supposed to transmit group addressed data traffic encrypted using the GTK and clients are supposed to decrypt that traffic using the GTK. However, an attacker can inject spoofed GTK-encrypted packets. An attacker might exploit this weakness in three ways; for ARP poisoning and man-in-the-middle attack; for injecting malicious code; and for a Denial-of-Service (DoS) attack.

The dummy authentication key establishment is also exposed to the Hole196 vulnerability. The Hole 196 vulnerability in WPA2 network with the dummy authentication key establishment is expected to be more severe because the attacker can be easily an insider without authentication. Also, the dummy authentication key establishment is unclear how to handle revocation. Our proposed scheme is not exposed to the Hole196 vulnerabliltiy and is able to handle revocation.

Another attack against wireless hotspots targets the client device. Because the client device and access point lack a shared key, the client device cannot authenticate the access

TABLE I
COMPARISON OF THE SETUP PROCEDURE: WPA, WI-FI PROTECTED SETUP: PIN METHOD AND PUSH-BUTTON METHOD. HERE THE REGISTRAR IS THE NETWORK ENROLLMENT CENTER.

Step	WPA	PIN	Push-Button
1	Turn on Access Point (AP)	Turn on AP/registrar	Turn on AP
2	Access AP	Turn on client device	Turn on client device
3	Set network name	Access registrar	Push button on AP
4	Activate security	Enter PIN	Push button on client device
5	Set pre-shared key		
6	Turn on client device		
7	Select network name		
8	Enter pre-shared key		

point. Fig. 1 shows an attacker that makes a rogue access point that pretends to be a normal access point. In this example, when a client device attempts to join the network AIRPORT, the client device connects to the access point that has the strongest signal among all access points beaconing the same Service Set IDentifier (SSID). After the client device connects to the rogue access point, the attacker can conduct a phishing attack in an attempt to obtain a user's login credentials or payment information [1]. The rouge access point attack is effective in its ability to intercept all user traffic and is also difficult to detect.

Recent research efforts have aimed to detect rogue access points by sensing Radio Frequency (RF) [15], [16] and measuring Transmission Control Protocol (TCP)/ Internet Protocol (IP) traffic characteristics, e.g., inter-packet spacing [17], [18] and round trip time [19], [20]. These approaches are primarily of interest in enterprise environments where full-time network administrators can monitor for on-site rogue access points; in remote hotspot environments, these approaches may carry too much overhead to be conducted or understood by a normal hotspot user. In our scheme, a normal user can easily detect a rogue access point.

III. KEY DISTRIBUTION USING HIERARCHICAL IDENTITY-BASED CRYPTOGRAPHY

As we have mentioned previously, a commercial Wi-Fi hotspot is open for public access, and such environments are not conducive for sharing login credentials or secret key information between the client and the access point. In our scheme, we establish keys between an unproven user and an access point using identity-based cryptography (first proposed by Shamir [21]). Before we describe our key distribution scheme in Section III-B, Section III-A overviews the hierarchical identity-based cryptography on which we build.

A. Overview of Hierarchical Identity-Based Cryptography

In this section, we review the hierarchical identity-based cryptography proposed by Gentry and Silverberg [5]. Gentry and Silverberg's approach assumes the difficulty of the Bilinear Diffie-Hellman problem and treats each cryptographic hash function as a random oracle. They then generate a user's private key in a hierarchical manner, so that the root Private Key Generator (PKG) can delegate a subspace of its private key generating capabilities to lower-level PKGs, which can in turn delegate further subspaces. The PKG at any level can only generate private keys for elements of its delegated space, and can further delegate only to lower-level PKGs. Although the lower-layer PKGs generate private keys for end-users, only the root PKG's public parameters are needed to verify a user's public key. Once keys have been disseminated using the technique proposed by Gentry and Silverberg, they also provide an encryption and signature scheme that provide chosen ciphertext security.

B. Application to Our Scheme

We build our scheme on hierarchical identity-based cryptography. Because of its uniqueness, uniformity, and usefulness for identification, we use the IEEE Medium Access Control (MAC) address as the user's public key. IEEE MAC address is consists of a 24-bit Organizationally Unique Identifier (OUI) [22], which is assigned by the IEEE and corresponds to the manufacturer of a network device, and Network Interface Controller (NIC) specific portion, which the manufacturer assigns to the device and is unique across all devices sharing the same OUI.

Given a public key, that is, a MAC address, we construct the corresponding private key using the key distribution scheme described in Section III-A. Using hierarchical identity-based cryptography [5], the PKG generates each user's private key using the user's MAC addresses and the security parameters provided by the root PKG. For example, with an IEEE MAC address, the root PKG would be the IEEE, while the manufacturer to which IEEE assigns OUI would be the second-level PKG. Each manufacturer would then assign private keys to each device when it manufactures. Though we could use single-level identity-based cryptography and have the IEEE directly provide the keys for each manufactured device, and our scheme supports this type of key distribution, our preferred approach uses hierarchical PKGs in order to delegate the workload of generating private keys to each manufacturer. Once a network interface device has a private key, it can use that key for authentication and to establish a secure

TABLE II
NOTATION

A	Access point
C	Client device
d_X	X's private key
e_X	X's MAC address (public key)
ID_X	X's certificate signed by a CA
$k_{A,C}$	Shared key between A and C
$\{M\}k$	Message M enciphered with key k
$S_A[M]$	Digital signature of M with A's private key
t_X	Time stamp from X's local clock

connection, as described in Section IV.

We now show that our usage of hierarchical identity-based cryptography [5] is compatible with its design assumptions. The MAC address is an ideal public key, because it is is both uniquely assigned and in widespread use. Also, our use of Gentry and Silverberg's scheme ensures that each user's secret key is private, and collusion between multiple network interface devices, even from the same manufacturer, does not provide any advantage on another user's private key. Thus, in our work, malicious users cannot learn the private key of other users who attempt to access internet on Wi-Fi hotspot, even under a collusion attack. Furthermore, a MAC address spoofing attacker [3] cannot compromise the private key, and therefore cannot sign messages for the victim, because the identity-based cryptography we have chosen is Chosen Plaintext Attack (CPA) secure (Gentry and Silverberg also provide a scheme secure against Chosen Ciphertext Attack (CCA)).

As in other standards (e.g. 802.16e mobile WiMAX), our scheme requires factory installation of cryptographic key. In current wireless network, our scheme might be added as a new 802.11 amendment or new wireless network standard.

IV. Protocol Definition

In this section, we present our protocol for a client device and an access point in a captive portal environment. We use the scheme in Section III-B, in which the manufacturer of each wireless network interface assigns it a unique IEEE Medium Access Control (MAC) address as its public key, and the manufacturer uses its Private Key Generator (PKG) functionality to provide the network interface with a private key corresponding to its MAC address. Our protocol uses this public key to securely exchange a private key, which in turn is used to establish a secure connection between an access point and a client device. Our scheme protects against MAC address spoofing, rouge access points, and certain MAC-layer Denial-of-Service (DoS) attacks.

Our proposed protocol consists of up to eight messages (notation in Table II) :

$$C \rightarrow A: \quad e_C \tag{1}$$

$$A \rightarrow C: \quad e_A, S_A[e_A, e_C, t_A],\ ID_A \tag{2}$$

$$C \rightarrow CA: \quad \text{Inquiry to verify } e_A \text{ (optional)} \tag{3}$$

$$CA \rightarrow C: \quad \text{Response (optional)} \tag{4}$$

$$C \rightarrow A: \quad e_C, \{\{S_A[e_A, e_C, t_A], k_{A,C}\}d_C\}e_A \tag{5}$$

$$A \rightarrow C: \quad \text{Client Puzzle (optional)} \tag{6}$$

$$C \rightarrow A: \quad \text{Puzzle Solution (optional)} \tag{7}$$

$$A \rightarrow C: \quad \text{Virtual AP's SSID} \tag{8}$$

The client sends a request with its Medium Access Control (MAC) address to access point in (1). In (2), the access point uses its Secure Sockets Layer (SSL) certificate to sign its MAC address, which establishes the authenticity of the access point, e_C, and time stamp t_A. To use this authentication feature an access point must have an SSL certificate signed by a Certificate Authority (CA) trusted by the user. In addition, we assume that the access point has a private key corresponding to its IEEE MAC address. To prove that the access point belongs to a certain domain (e.g. t-mobile.com), the access point signs its MAC address with its SSL private key, and then sends the signed message ($S_A[e_A, e_C, t_A]$) with the certificate (ID_A) to the client C. Here e_C and t_A are used for preventing replay attack.

In order to detect a rogue access point, client C checks the authenticity of access point A by first ensuring that ID_A is a valid certificate signed by a CA that the client trusts, checks that the certificate matches the access point's SSID, and then verifies the signature $S_A[e_A, e_C, t_A]$. Also, client checks whether e_C matches with it's own MAC address and also checks t_A. If each of these checks are successful, then the client trusts that the MAC address given (e_A) is associated with the access point to which the client wishes to connect. (This approach is similar to that used by HyperText Transfer Protocol Secure (HTTPS)). The client has not yet verified that the access point to which it is connected is in fact legitimately using that MAC address; the client will verify this fact later. In (3) and (4), client inquires optionally the validity of e_A to check accurately whether e_A has been revoked. We leave the details of this check to the implementation.

In (5), the client device sends the received signed message ($S_A[e_A, e_C, t_A]$) together with a shared key chosen by the client $k_{A,C}$ to the access point. In order to verify that this message is sent by C, the message is encrypted using C's MAC address. In addition, to ensure that the access point is legitimately using address e_A, the entire message is encrypted using A's MAC address, so that only A can obtain $k_{A,C}$. The shared key ($k_{A,C}$) can then be used as a shared secret key, for example as an WPA2 AES (Advanced Encryption Standard) key. The AP verifies the received message by checking that the MAC address of received message matches the one in the $S_A[e_A, e_C, t_A]$ and the current local time is within the validity period from timestamp t_A.

When all checks have passed, the access point can immediately provision resources that allow the client to establish a secure connection to the access point. However, the access

point may have limited resources, and an attacker that has obtained a number of wireless network interfaces could use them to take up all the resources of an access point, preventing legitimate users from connecting. Messages (6) and (7) allow the access point to send a puzzle to the client and receive a solution in response. These messages should be encrypted using the key $k_{A,C}$ to prevent man-in-the-middle attacks. The challenge should be of sufficient difficulty that an attacker cannot obtain all of a base station's secure connection resources. Our protocol is not sensitive to the type of challenge used; one challenge that can be used with our scheme is that of Juels et al. [23].

After the access point has received a puzzle solution, it creates a virtual access point. A virtual access point is a logical entity that exists within one access point [24]. Each virtual access point can have a distinct Service Set IDentifier (SSID) and security parameters (such as a WPA2 key). A single access point can support multiple virtual access points; a typical access point can support 64 or 128 virtual access points. The access point randomly generates an SSID specifically for this client and creates an virtual access point with the chosen SSID and shared key $k_{A,C}$. The access point then notifies the client of the SSID of the new access point in the final message of our protocol.

Our scheme does not depend on WPA pre-shared key mode, and provides better security, especially against Hole196 vulnerability described in Section II, in the sense that each client connects to a unique virtual access point. Also, our scheme is unrestricted from the security problem of WPA-PSK and RC4 that WPA still relies on. Moreover, our scheme can detect rogue access points and is resistant to certain types of denial-of-service attacks. Even if an attacker spoofs the MAC address of access point or client device or relays one of protocol messages, it does not learns the shared key ($k_{A,C}$) because it does not know the private key of the access point (d_A). Consequently, the attacker can neither eavesdrop on nor modify messages that are transmitted within a secure connection established using our scheme.

V. Conclusions

In hotspot environments or other scenarios where the distribution of authentication credentials and pre-shared information is difficult, our scheme establishes a secure connection in the presence of malicious entities. Two communication parties agree on a shared, secret key using the public key/private key pair generated based on hierarchical identity-based cryptography. Focusing on applications in public wireless hotspots, we studied the security vulnerabilities, especially at the Medium Access Control (MAC) layer, of captive portals, which are currently used in hotspots. Using the already available MAC address as the public key, our scheme is not only practical, i.e., easy to deploy and scalable, but also protects users' integrity and confidentiality against colluding attackers. In particular, we defend against the attacks of MAC address spoofing, rouge access point, and denial-of-service attack. The integration of our scheme within 802.11i is left as future work.

References

[1] B. Potter, "Wireless hotspots: Petri dish of wireless security," *Communications of the ACM*, vol. 49, no. 6, pp. 50–56, Jun. 2006.

[2] P. Henry and H. Luo, "WiFi: What's next?" *IEEE Commun. Mag.*, vol. 40, no. 12, pp. 66–72, Dec. 2002.

[3] K. J. Hole, E. Dyrnes, and P. Thorsheim, "Securing Wi-Fi networks," *IEEE Computer*, vol. 38, no. 7, pp. 28–34, Jul. 2005.

[4] N. Sastry, J. Crowcroft, and K. Sollins, "Architecting citywide ubiquitous Wi-Fi access," in *HOTNET'07*, Atlanta, GA, Nov. 2007.

[5] C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography," *Lecture Notes in Computer Science*, vol. 2501, pp. 548–566, 2002.

[6] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," *Lecture Notes in Computer Science*, vol. 2259, pp. 1–24, 2001.

[7] A. Stubblefield, J. Ioannidis, and A. D. Rubin, "A key recovery attack on the 802.11b wired equivalent privacy protocol (WEP)," *ACM Transactions on Information and System Security*, vol. 7, no. 2, pp. 319–332, May 2004.

[8] A. Bittau, M. Handley, and J. Lackey, "The final nail in WEP's coffin," in *27th IEEE Symposium on Security and Privacy*, Oakland, CA, May 2006, pp. 386–400.

[9] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions," in *12th USENIX Security Symposium*, Washington, D.C., Aug. 2003, pp. 15–27.

[10] Wi-Fi Alliance, *Wi-fi protected setup specification*, 2007.

[11] H. Xia and J. e Brustoloni, "Detecting and blocking unauthorized access in Wi-Fi networks," *Lecture Notes in Computer Science*, vol. 3402, pp. 795–806, 2004.

[12] M. Brunato and D. Severina, "WilmaGate: a new open access gateway for hotspot management," in *WMASH'05*, Cologne, Germany, Sep. 2005, pp. 56–64.

[13] Z. Yang, A. C. Champion, B. Gu, X. Bai, and D. Xuan, "Link-layer protection in 802.11i WLANs with dummy authentication," in *WiSec'09*, Zurich, Switzerland, Mar. 2009, pp. 131–138.

[14] WPA2 hole196 vulnerability. [Online]. Available: http://www.airtightnetworks.com/WPA2-Hole196

[15] A. Adya, P. Bahl, R. Chandra, and L. Qiu, "Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks," in *MobiCom'04*, Philadelphia, PA, Sep. 2004, pp. 30–44.

[16] P. Bahl, R. Chandra, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill, "Enhancing the security of corporate Wi-Fi networks using DAIR," in *MobiSys'06*, Uppsala, Sweden, Jun. 2006, pp. 1–14.

[17] R. Beyah, S. Kangude, G. Yu, B. Strickland, and J. Copeland, "Rogue access point detection using temporal traffic characteristics," in *GLOBECOM'04*, Dallas, TX, Nov. 2004, pp. 2271–2275.

[18] W. Wei, K. Suh, B. Wang, Y. Gu, J. Kurose, and D. Towsley, "Passive online rogue access point detection using sequential hypothesis testing with TCP ACK-pairs," in *IMC'07*, San Diego, CA, Oct. 2007, pp. 365–378.

[19] C. D. Mano, A. Blaich, Q. Liao, Y. Jiang, D. A. Cieslak, and et al., "RIPPS: Rogue identifying packet payload slicer detecting unauthorized wireless hosts through network traffic conditioning," *ACM Transactions on Information and System Security*, vol. 11, no. 2, pp. 1–23, Mar. 2008.

[20] H. Han, B. Sheng, C. C. Tan, Q. Li, and S. Lu, "A measurement based rogue AP detection scheme," in *INFOCOM'09*, Rio de Janeiro, Brazil, Apr. 2009.

[21] A. Shamir, "Identity-based cryptosystems and signature schemes," *Lecture Notes in Computer Science*, vol. 196, pp. 47–53, 1985.

[22] IEEE OUI and company_id assignments. [Online]. Available: http://standards.ieee.org/regauth/oui/

[23] A. Juels and J. Brainard, "Client puzzles: A cryptographic countermeasure against connection depletion attacks," in *NDSS'99*, San Diego, CA, Feb. 1999, pp. 151–165.

[24] B. Aboba, *Virtual Access Points*. IEEE 802.11-03/154r1, 2003.