Practical Defenses for Evil Twin Attacks in 802.11

Harold Gonzales[†], Kevin Bauer[†], Janne Lindqvist[‡], Damon McCoy[§], Douglas Sicker[†]
[†]University of Colorado {gonzaleh, bauerk, sicker}@colorado.edu
[‡]Carnegie Mellon University janne@cmu.edu
[§]University of California–San Diego dlmccoy@cs.ucsd.edu

***Abstract*—Open-access 802.11 wireless networks are commonly deployed in cafes, bookstores, and other public spaces to provide free Internet connectivity. These networks are convenient to deploy, requiring no out-of-band key exchange or prior trust relationships. However, such networks are vulnerable to a variety of threats including the *evil twin attack* where an adversary clones a client's previously-used access point for a variety of malicious purposes including malware injection or identity theft.**

We propose defenses that aim to maintain the simplicity, convenience, and usability of open-access networks while offering increased protection from evil twin attacks. First, we present an evil twin detection strategy called *context-leashing* that constrains access point trust by location. Second, we propose that wireless networks be identified by uncertified public keys and design an SSH-style authentication and session key establishment protocol that fits into the 802.1X standard. Lastly, to mitigate the pitfalls of SSH-style authentication, we present a crowd-sourcing-based reporting protocol that provides historical information for access point public keys while preserving the location privacy of users who contribute reports.

I. Introduction

Open-access 802.11 wireless networks that do not offer any link-layer security are a popular way to provide Internet connectivity at universities, cafes, and commercial hotspots. In fact, a recent study found that 42% of wireless access points (APs) are left open [1]. Convenience often discourages AP operators from enabling security, but this convenience comes with a price. Clients that associate with open APs are vulnerable to a number of trivial threats from eavesdropping and data injection such as DNS hijacking, phishing, altering HTTP, and many other attacks.

However, beyond these immediate threats, there is a more subtle and long term future threat whereby an attacker can disrupt an existing secure association or prevent a client from initially associating to a secure 802.11 AP using well-known denial-of-service attacks [2]. The attacker can then trick the client into automatically associating to an *evil twin* of an open AP that the client had previously used. This attack leverages preferred network lists, which enable automatic re-association with previously-used networks. The evil twin attack can be highly effective and has even been observed in the wild at high traffic areas such as airports [3].

Our primary objective is to provide simple, convenient, and usable techniques to mitigate the threat posed by evil twin APs. To this end, we first present an evil twin detection strategy that constrains AP trust by location. Next, we argue that wireless networks be identified by public keys that can authenticate APs and bootstrap session keys to secure data delivery.

Constraining trust by location. Our first approach, called *context-leashing*, records all of the APs that are visible to a client when that client first associates to an AP. These recorded APs act as wireless landmarks so that the client can recognize the correct context (or location) for that network. During subsequent associations, the client should automatically associate to this AP *only if* its observed context matches the previously recorded context. This approach is simple and practical since it requires only client-side modifications, no additional or specialized hardware, no explicit user interaction, and is conceptually similar to wireless device localization methods that have already proved successful [4]. We perform an empirical evaluation using real traces taken from several wireless hotspot locations and show that context-leashing achieves a high evil twin detection rate while minimizing false positives against a variety of attack models.

Establishing cryptographic identity. While context-leashing can detect anomalies during AP selection, it cannot make any strong assertions about an AP's identity. To address this, we propose that AP identities be bound to self-signed public keys using a new authentication module designed for the 802.1X extensible authentication protocol (EAP). The Simple Wireless Authentication Technique (*EAP-SWAT*) follows the principle of *trust-on-first-use* (TOFU) to authenticate the AP and bootstrap session keys to secure data delivery. In the TOFU model, clients may be vulnerable to impersonation attacks when they associate to an AP for the first time. However, TOFU ensures that the AP is the same for each subsequent association. This provides stronger protection against evil twin attacks without requiring pre-shared secrets (as in WPA-PSK) or other prior trust relationships.

To address the potential vulnerabilities when associating to an AP for the first time, we present a collaborative reporting system based on Wifi-Reports [5] that allows users to easily submit reports on the stability of the public keys for the APs that they use. The reporting system tolerates misbehaving users by rate limiting report submissions and preserves participating users' location privacy by employing a blind signature-based submission protocol.

Contributions. This paper offers three primary contributions:

1) We propose a method to detect evil twin attacks using contextual information and we empirically evaluate it.
2) We argue that wireless networks should be securely identified by public keys. To this end, we design a trust-on-first-use authentication and session key establishment protocol that can secure wireless networks with minimal changes to the existing 802.1X infrastructure.
3) We describe a collaborative reporting system that reduces the risks of the trust-on-first-use model.

II. Evil Twin Detection

When a wireless client returns to previously-used 802.11 network, their operating system's network management tools commonly attempt to automatically re-associate to one of

these networks through active probing, often without the user's knowledge.[1] Automated re-association increases the threat of *evil twin attacks*, where an adversary deploys an AP with the same identity, in terms of network name (SSID) and MAC address, as one of a client's previously-used and trusted APs. To make matters worse, adversaries can use jamming and other denial-of-service tactics [2] to prevent targeted clients from associating to secure APs, thereby encouraging clients to associate with an evil twin.

In order to reduce the threat of evil twins, we propose that each wireless client attempt to answer the following question: *Am I in the same location as when I first used this network and added it to my list of preferred networks?* Following this intuition, we propose *context-leashing* to detect evil twin APs that are deployed in different locations than the original, authentic AP. Context-leashing is based on observing *contextual information*, or the list of all access points that are visible to a client when it is associated with a particular wireless network at a familiar location. Furthermore, context-leashing is practical, since it requires only commodity wireless hardware, stock drivers with minimal client-side modifications, and can be deployed without any changes to, or cooperation from, the APs.

A. Context-Leashing Methodology

We now formally describe the context-leashing approach to evil twin detection. First, when the client associates to an AP for the first time, the client observes the initial context vector $C_i = \{(c_1, r_1), ..., (c_j, r_j), ..., (c_n, r_n)\}$, which is a set of tuples that contain all SSIDs c_k and their respective received signal strength indication (RSSI) values r_k that are visible from location i when associated with an access point identified by SSID c_j.[2] The client subsequently maintains sets of learned contexts for each SSID to which it associates.

Next, when a client encounters SSID c_j again in the future, it should compare the context in which c_j is observed with the previously learned context for that AP. To match an observed context O with one of the learned contexts C_i, it is necessary to apply a metric that can capture set dis-similarity to determine if the AP in question is an evil twin. We next present two metrics for detecting evil twins.

SSID set difference. The first metric simply compares the context of APs that are present when the client wishes to associate to an AP to the context when the client first associated to that AP. Before associating to an AP identified by SSID c_j, we compute the Jaccard distance [6] between the set observed SSIDs O and the previously learned SSIDs in the expected context C_i (for $c_j \in C_i$), as shown in Equation 1.

$$J = 1 - \frac{|O \cap C_i|}{|O \cup C_i|}, \text{ for } c_j \in C_i \tag{1}$$

If J is larger than a threshold τ, then the context in which c_j is observed is too different than the expected one for that particular SSID. Consequently, the access point should be classified as an evil twin.

Signal strength difference. A determined adversary can defeat the set difference approach simply by observing the correct context and replicating it when launching an evil twin attack. In fact, commodity wireless cards often support virtual interfaces, making it possible to setup several APs with a single wireless card on a laptop. To mitigate this simple attack, we propose another method that computes set difference, but also considers the APs' expected RSSI values.

The signal strength difference approach computes a set difference that is weighted by the RSSI values from the APs in the observed context O and the learned context C_j. We define $RSSI(X, ssid)$ to be the RSSI value for the AP *ssid* from the set X. If *ssid* is not present in set X, then $RSSI(X, ssid)$ returns 0. Equation 2 calculates the cumulative differences in each AP's RSSI values between the learned and observed contexts.

$$K = \frac{\sum_{i=1}^{n} |RSSI(O,s_i) - RSSI(C_j,s_i)|}{2}, \quad \text{for } n = |O \cup C_j|, \; s_i \in O \cup C_j \tag{2}$$

Note that penalties are dealt if an AP is missing from one of the sets or if there is significant difference in an AP's RSSI values between sets.[3] Similar to the SSID set difference approach, if $K > \tau$, then the observed context is sufficiently different and the AP is classified as an evil twin. In the next section, we show that context-leashing is an effective and practical approach to evil twin detection by conducting an evaluation using real wireless traces.

III. Evil Twin Detection Evaluation

To demonstrate the efficacy of the context-leashing techniques for evil twin detection, we present a series of experiments conducted at several real WiFi hotspot locations. First, we present a number of different attack models, each with varying strategies and abilities. Next, we describe the methodology used to collect the real context information and RSSI values. Finally, we present evil twin detection rates along with an analysis of the potential for false positives, context stability, and parameter tuning.

A. Attack Models and Client Assumptions

To understand the effectiveness of context-leashing in a variety of adversarial scenarios, we construct five different adversaries with increasingly intelligent strategies and powerful abilities. For all attack models, we assume that an attacker can inject access points into a target environment using commodity WiFi hardware, but does not have the ability remove access points.[4] Finally, we assume that current public WLAN localization services such as Skyhook (that use techniques similar to context-leashing) are vulnerable to relatively simple attacks [7].

[1] These active probing messages include the names of the networks to which the client will automatically associate.

[2] We believe it is acceptable to trust the context on the client's first association, since the association is performed manually by the user.

[3] Equation 2 is divided by two in order to ensure that $K \in [0, 1]$.

[4] While it is possible to jam the beacons and all traffic from an access point, to perform selective jamming of a particular access point requires expensive non-standard hardware that can quickly change from receive to transmit mode. An attacker could completely jam the channel and only allow itself to transmit, but this would also result in blocking the victim's transmissions.

Adversaries. The weakest adversary we consider is called the *single* attacker, who only deploys an evil twin of an official access point from another location. This is a simple attack that can be executed by listening to the probe requests from the client and transmitting beacons for one of these networks. Freely-available rogue AP deployment tools such as Airsnarf [8] can implement this attack.

The second adversary, called *mismatching context*, has knowledge about the context of the location where the legitimate access point is located. The adversary deploys a set of access point beacons transmitted at a constant power level that do not exactly match the context that the client recorded when she initially connected to the access point, but rather from a different measurement taken on one of the other days at a slightly different position in that location.

A slight variation on the mismatching context adversary is one that, instead of transmitting beacons at a constant power level, can match the RSSI values from that measurement in addition to the access point set. We refer to this attack as *tx power mismatching context*.

The last type of adversary possesses the set of access points and RSSI values that the client recorded when initially connecting to an access point. This could be done through an exploit that tricks the client into disclosing this information or by malware installed on the client. The first adversary in this class, which we call *matching context*, uses this information to transmit beacons at a constant power level to match the exact set of access points as the initial set that the client recorded. The adversary that can also match the RSSI values from the client's exact initial context is called *tx power matching context*.

Client assumptions. The client is a standard laptop computer that does not have the ability to localize itself using GPS[5] or GSM techniques. We assume that the client records their initial measurement at a location and that they have added all of the "official access points" from six different locations to their preferred network list. The client moves to another one of the six locations where an adversary attempts to trick them into using an evil twin access point at a different location. The client then measures the current context and uses one of the context-leashing methods described in Section II-A to determine if this network is an evil twin.

B. Experimental Setup

In order to understand how the context-leashing detection methods perform in practice, we collected context measurements from 19 different wireless hotspot locations in Boulder, CO (shown in Figure 1). We used Kismet [9] to identify all 802.11 networks near a particular location and their RSSI values by passively recording broadcast beacons. We also used hotspot data from Seattle, WA (shown Figure 2) that is publicly available in the CRAWDAD wireless trace repository [10]. For the Seattle data, 8–13 measurements were collected at each location (one or two measurements per day) over the course of seven days in October 2008. For the Boulder data, 14 measurements were collected (one per day) over the course of two seven day periods. The first measurement period occurred in January 2008 and the second in May 2009.[6] Finally, at each location we note which access point is the official access point for that hotspot location.

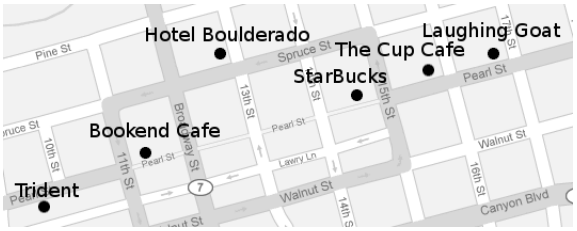


Fig. 1. Measured hotspot locations near Pearl Street, Boulder, CO

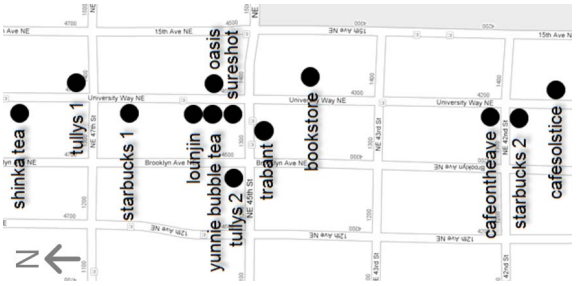


Fig. 2. Measured hotspot locations near University Avenue, Seattle, WA

We quantify the performance of the context-leashing detection using the following metrics:

- *True Positive Rate* (TPR): The fraction of evil twins correctly detected (*e.g.,* sensitivity).
- *False Positive Rate* (FPR): The fraction of legitimate APs identified as evil twins (*e.g.,* false alarms).

We initially assume that evil twins attempt to attack clients in a different location from the access point's true location. In Section IV, we discuss methods to mitigate evil twin attacks that are launched in the same location as the true access point.

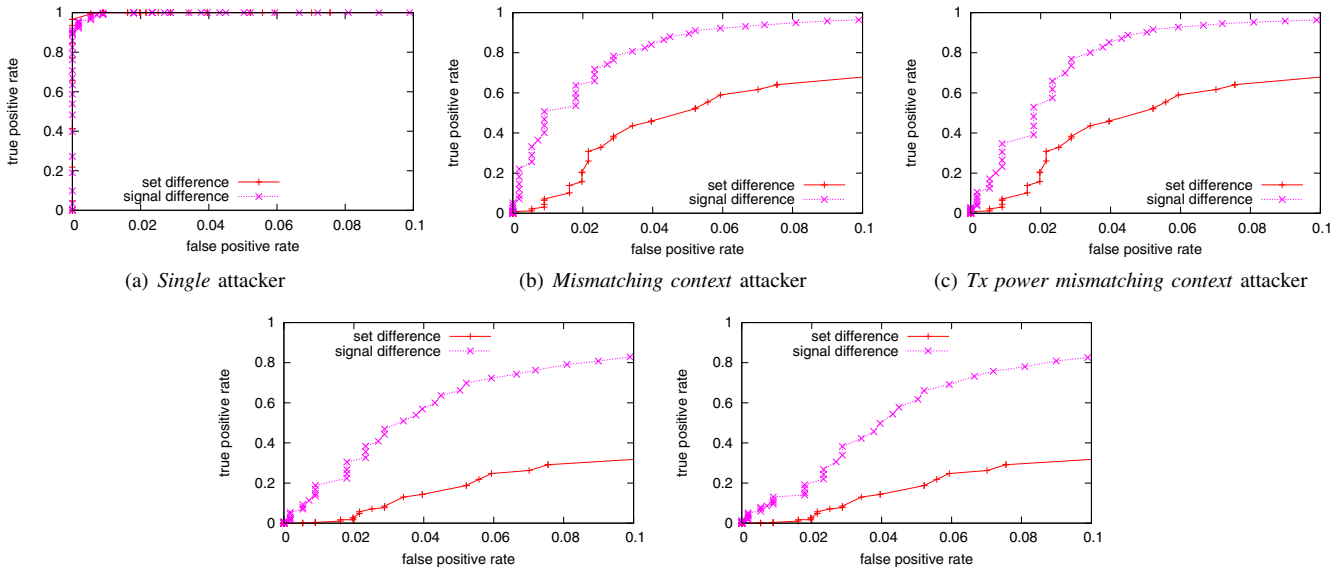
C. Results

Detection rates and false positives. Receiver operating characteristic (ROC) curves that illustrate the relationship between the TPR and FTP for the context-leashing detection methods in each adversarial scenario are given in Figures 3(a)-(e). Both detection methods perform almost perfectly against the *single* adversary as shown in Figure 3(a). This strong performance against the single adversary is encouraging, as it can be easily launched with tools like Airsnarf [8] and consequently may be the most common type of evil twin attack in the wild.

The results for the remaining adversaries show that with varying degrees of success, the context-leashing methods can detect more powerful attackers that have knowledge of legitimate access points and their correct contexts. For the remaining four adversarial scenarios, the signal strength difference approach gives a higher TPR than the set difference method when the FPR is less than 0.1.

[5]We believe this is a reasonable assumption, since GPS typically does not work indoors.

[6]The reason for the two sets of measurements collected in Boulder, CO is to evaluate the stability of the contexts over time.



(a) *Single* attacker (b) *Mismatching context* attacker (c) *Tx power mismatching context* attacker

(d) *Matching context* attacker (e) *Tx power matching context* attacker

Fig. 3. ROC curves for the five different adversarial scenarios

Even in the worst case, if we assume an extremely powerful adversary that can exactly match the set of access points and the RSSI values from the client's initial association, the signal strength difference method detects the evil twin 20% of the time when the FPR is fixed to 0.02 and 82% of the time when the FPR is set to 0.1 (see Figure 3(e)). However, since RSSI values are hard to predict due to various environmental and unobservable factors, we argue that it is extremely difficult for an attacker to reliably transmit packets such that they exactly (or even closely) match the expected RSSI values at the receiver. Thus, the *tx power matching context* attacker is mostly of theoretical interest. We believe that the most powerful practical adversary is *matching context*. Against this adversary, the signal strength difference method achieves 30% and 82% detection rates for FPRs of 0.02 and 0.1, respectively (see Figure 3(d)).

A more realistic attacker is one that can't replicate the *exact* set of access points in the client's initial context, but has access to slightly different, but similar measurements from the same location (this is the *mismatching context* adversary).[7] In this case, Figure 3(b) shows that the signal strength difference method can detect the attacker 63% and 96% of the time with a FPR of 0.02 and 0.1, respectively. The *tx power mismatching* attack experiences similar detection rates (see Figure 3(c)).

Threshold tuning and context stability. We next wish to answer two questions about the FPR: (1) Can we tune the detection threshold τ on one set of locations and achieve a similar FPR on another set of locations? (2) Are the contexts relatively stable over time?

To answer the first question, we find that if we select a τ value that achieves a particular fixed FPR for the Seattle locations and apply the same threshold to the Boulder hotspots, we can expect a similar FPR in Boulder. For example, an acceptably low FPR of 0.02 can be achieved for the set and signal strength difference methods in the Seattle locations with $\tau = 0.75$ and $\tau = 0.59$, respectively. These same τ values correspond to the same low FPR at the Boulder locations. Thus, the clients could simply hard-code these τ values.[8]

In order to answer the second question, we analyze the two sets of measurements that were collected three months apart at the same locations in Boulder. Out of a total of 113 access points detected at the six locations in Boulder, the set of APs from the January measurements included only three APs that were not detected in the May data set. The May measurements included five access points that were not visible in the January set of access points. Also, the RSSI measurements for each AP remained stable across the two time periods.

D. Discussion

Context-leashing gives end-users a simple way to protect themselves from evil twin attacks without requiring any changes to the wireless infrastructure. This property makes this approach very easy for a single client to deploy without coordination from any standards body. In addition, this technique works even when there are no other APs in a particular AP's context.

However, the main limitation of this detection approach is that it does not provide authentication, confidentiality, and integrity that can prevent eavesdropping and data injection attacks. In fact, regardless of whether evil twin access points are detected or ignored, any arbitrary wireless device can inject wireless frames and eavesdrop near an unsecured wireless network. In addition, context-leashing cannot detect evil twins that are launched in the correct context. Finally, while context-leashing is ideal for single-AP networks (*e.g.,* cafe or home

[7]For instance, an attacker could query a war-driving database [11] to locate such information.

[8]However, a detailed analysis of parameter tuning across many diverse environments is beyond the scope of this paper.

networks), it may not support multi-AP corporate or university WLANs where users roam between APs. We next present a simple protocol that overcomes these limitations by authenticating wireless networks and establishing session keys without requiring any prior trust or out-of-band secrets.

IV. Simple Wireless Authentication Technique

In order to further reduce the evil twin threat and simultaneously mitigate a wide variety of other security threats including arbitrary data frame injection and eavesdropping, it is necessary to securely identify wireless networks and establish per-session keys to secure data delivery. However, existing solutions such as WPA-PSK require the client to obtain a secret pass-phrase through an out-of-band process to bootstrap the session keys. For wireless networks providing free Internet access at cafes or other public places, it's not always feasible to distribute network passwords to users.

We first argue that wireless networks should be identified by uncertified cryptographic credentials in a "trust-on-first-use" (TOFU) manner, similar to SSH-style authentication, where APs assert their identities with self-signed public keys. While the TOFU model is ostensibly vulnerable to impersonation attacks when the client associates to a network for the first time, public keys ensure that the client is using the same wireless network for all subsequent associations with the same SSID. These self-signed public keys can be used to bootstrap a session key without the need to exchange secrets out-of-band, and thus, can be used to secure open-access public wireless networks that serve transient clients.

To mitigate the potential risks associated with the TOFU security model, we next describe a collaborative reporting protocol that allows users to securely and anonymously submit reports on the stability of APs' public keys.

A. EAP-SWAT Design Overview

Before presenting the details of the protocol, we enumerate its desired features:

- *Identifies wireless networks by their self-signed public-keys, and therefore requires no certificate authorities, pre-shared secrets, or pre-existing trust relationships.*
- *Provides data confidentiality, authenticity, and integrity*
- *Integrates into the 802.1X authentication framework as an extensible authentication protocol (EAP) module.*
- *Mitigates the risks posed by the TOFU model.*

We define a new 802.1X EAP module called *EAP-SWAT* (Simple Wireless Authentication Technique) that is based on EAP-TTLS [12], which uses a signed certificate to authenticate the wireless network and bootstrap an AP-client session key. To associate to a wireless network securely with *EAP-SWAT*, the client sends an 802.11 authentication request message to initiate the 802.1X protocol [13]. Next, a one-way authenticated TLS session is created, which establishes a session key shared between the AP and the client, offering confidentiality with forward secrecy and data integrity. Note that *EAP-SWAT* differs from EAP-TTLS [12] since the client must decide whether or not to trust the AP's self-signed certificate. More details about *EAP-SWAT* can be found in [14].

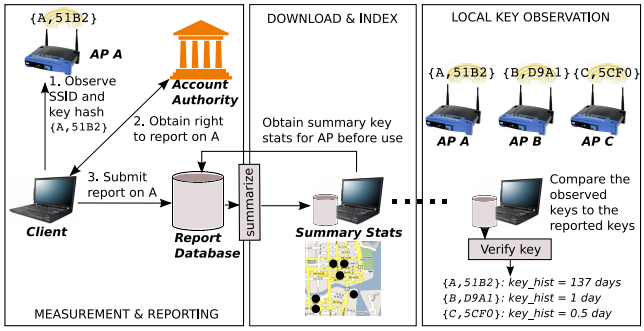


Fig. 4. The collaborative reporting protocol proceeds in three phases. *Phase 1, Measurement and Reporting*: Clients observe an AP's SSID and advertised public key, obtain blind signature tokens from the account authority, and submit reports to the database. *Phase 2, Download and Index*: Clients obtain reports on APs by querying the report database. *Phase 3, Local Key Observation*: Using key reports obtained from the database, clients decide whether to trust an AP's key for the first time based on its key history.

To establish a secure network identifier, APs should include a signature of their human-readable SSID (signed with their private key) and their public key as part of a vendor-specific information element in their beacon messages.

It is also necessary to support multi-AP networks where many APs broadcast the same SSID, such as large corporate or university networks. Multi-AP networks may simply advertise a single public key for their SSID, allowing users to roam between APs without re-authenticating for each AP they use.

B. Collaborative Reporting

To address the potential vulnerabilities of the TOFU model, we describe a collaborative reporting system that securely collects and aggregates information about the stability of APs' unverified public keys. This approach provides valuable information about the history of a network's public key, which reduces the threat of initially associating with a transient rogue AP. The collaborative reporting system involves a report submission protocol based on Wifi-Reports [5], an RSA blind signature-based [15] report submission framework.

The collaborative reporting system (illustrated in Figure 4) consists of an account authority, a report database, wireless clients, and access points. Clients submit reports on the public keys of APs that they use and the submission protocol ensures that reporting users' location privacy is preserved. The reporting protocol proceeds in three phases.

Phase 1: Measurement and reporting. To report on the stability of an AP's key, the client first obtains a blind signature "token" from the account authority. This token allows the user to submit a key report for that AP in a secure manner and preserves the client's location privacy across reports on multiple APs. Using this token, the client submits their report[9] containing a cryptographic hash of the AP's advertised public key to the report database.

Phase 2: Download and index. Clients wishing to obtain information on an AP's key history can query the report

[9]To prevent users from leaking network or transport layer identifiers, they should use an anonymizing overlay such as Tor [16] when communicating with the report database.

database. The report database responds to queries with the following information, obtained by majority consensus over the submitted reports to reduce the threat of malicious reporting: (1) the most recent public key (identified by its hash) and (2) how long that key has been used.

Phase 3: Local key observation. Clients decide whether to trust an AP based on the AP's public key history, associating if the key is stable. Otherwise, if there is little information about the key, the client should be suspicious of the AP.

Discussion. This design enables robust stability reporting, but some considerations remain. For instance, a network may change its SSID and public key. The network's administrators can submit a signed report verifying the updated information or provide a signed attestation of the update directly to users.

Another important requirement for the reporting protocol is to restrict any single user's ability to significantly influence the reports. To reduce this threat, multiple reports for the same APs from the same user are linkable, since the account authority limits each client to a single token per AP. The reporting database can remove previous votes submitted by that user for that AP when they receive an updated vote. In addition, this system mitigates Sybil attacks [17] by requiring that reporting users first register with an account authority that can perform micro-transactions out-of-band.

V. Related Work

Public keys as network identifiers. Aura *et al.* argue that public key network identifiers allow for PKI-based authentication when the infrastructure is available, yet can also identify networks when no signed certificates are available [18].

Trust-on-first-use authentication. The Secure Shell (SSH) authentication protocol [19] was one of the first widely adopted TOFU authentication protocols. This method of authentication is also used in the Session Initiation Protocol (SIP) [20] and Host Identity Protocol (HIP) [21], which are widely used for setting up Internet voice and video calls.

Collaborative monitoring to improve security. ConfiDNS [22] is a collaborative monitoring system that uses a large set of cooperative DNS resolvers to improve the security of DNS. This system measures the stability of name-to-IP mappings over time and agreement of mappings at multiple locations to detect local DNS injection and poisoning attacks. Perspectives [23] is another collaborative monitoring system that uses a number of hosts in different locations to monitor SSH and HTTPS self-signed certificates. It also maintains a record of the key's stability and age along with detecting man-in-the-middle attacks by reporting when a disagreement in keys between monitors occurs.

Rogue AP mitigation. Prior work has also focused on the secure device pairing problem. Roth *et al.* developed a simple protection mechanism to prevent users from connecting to an evil twin access point by equipping the "official" AP with a light capable of emitting different colors [24]. Yang *et al.* argue that security should be provided at the 802.11 MAC layer and present a "dummy authentication" key establishment algorithm based on public key cryptography [25]. Other approaches link identity to location to detect identity-based attacks [26].

VI. Conclusion

Given the prevalence of open-access 802.11 networks, users are vulnerable to a variety of threats such as the evil twin attack. We present two simple defensive strategies to mitigate the evil twin attack in 802.11 access point selection. First, we present an evil twin detection technique called context-leashing that constrains access point trust by location. Next, we propose an SSH-style authentication protocol that can securely identify wireless networks and be used to bootstrap a session key exchange. To mitigate the risks of SSH-style authentication, we describe a crowd-sourcing collaborative reporting protocol that provides historical information on APs' keys and preserves the location privacy of the submitting users.

Acknowledgments. This work was supported in part by a Faculty Award from IBM. D. McCoy was supported in part by a CCC-CRA-NSF Computing Innovation Fellowship.

References

[1] A. J. Nicholson, Y. Chawathe, M. Y. Chen, B. D. Noble, and D. Wetherall, "Improved access point selection," in *MobiSys*, 2006.
[2] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions," in *USENIX Security*, 2003.
[3] T. Abdollah, "Ensnared on the wireless web," http://articles.latimes.com/2007/mar/16/local/me-wifihack16, 2007.
[4] A. Lamarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, "Place Lab: Device positioning using radio beacons in the wild," in *Pervasive*, 2005.
[5] J. Pang, B. Greenstein, D. McCoy, M. Kaminsky, and S. Seshan, "Wifi-reports: Improving wireless network selection with collaboration," in *MobiSys*, 2009.
[6] C. van Rijsbergen, *Information Retrieval. 2nd ed.* Butterworths, 1979.
[7] N. O. Tippenhauer, K. B. Rasmussen, C. Pöpper, and S. Čapkun, "Attacks on public wlan-based positioning systems," in *Mobisys*, 2009.
[8] "Airsnarf a rogue ap setup utility," http://airsnarf.shmoo.com/.
[9] "Kismet," http://www.kismetwireless.net.
[10] J. Pang, "CRAWDAD data set cmu/hotspot (v. 2009-04-15)," Downloaded from http://crawdad.cs.dartmouth.edu/cmu/hotspot.
[11] "WiGLE: Wireless geographic logging engine," http://www.wigle.net.
[12] P. Funk and S. Blake-Wilson, "RFC 5281: Extensible Authentication Protocol Tunneled Transport Layer Security," August 2008.
[13] "802.1X Standard," IEEE, July 2004.
[14] K. Bauer, H. Gonzales, and D. McCoy, "Mitigating evil twin attacks in 802.11," in *WIDA*, 2008.
[15] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko, "The one-more-rsa-inversion problems and the security of chaum's blind signature scheme," *Journal of Cryptology*, vol. 16, no. 3, pp. 185–215, 2003.
[16] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *USENIX Security*, 2004.
[17] J. R. Douceur, "The sybil attack," in *IPTPS*, 2002.
[18] T. Aura, M. Roe, and S. J. Murdoch, "Securing network location awareness with authenticated dhcp," in *Securecomm*, 2007.
[19] T. Ylonen and C. Lonvick, "RFC 4252: The Secure Shell (SSH) Authentication Protocol," January 2006.
[20] "RFC3261 SIP," http://www.faqs.org/rfcs/rfc3261.html.
[21] "Host identity protocol architecture," http://www.ietf.org/rfc/rfc4423.txt.
[22] L. Poole and V. S. Pai, "ConfiDNS: Leveraging scale and history to detect compromise," in *USENIX 2008 Annual Technical Conference*.
[23] D. Wendlandt, D. Andersen, and A. Perrig, "Perspectives: Improving ssh-style host authentication with multi-path network probing," in *USENIX Annual Technical Conference*, June 2008.
[24] V. Roth, W. Polak, E. Rieffel, and T. Turner, "Simple and effective defense against evil twin access points," in *WiSec*, 2008.
[25] Z. Yang, A. C. Champion, B. Gu, X. Bai, and D. Xuan, "Link-layer protection in 802.11i WLANS with dummy authentication," in *WiSec*, 2009.
[26] D. B. Faria and D. R. Cheriton, "Detecting identity-based attacks in wireless networks using signalprints," in *Wise*, 2006.