Pro-active Key Distribution using Neighbor Graphs

Arunesh Mishra, Min-ho Shin, William A. Arbaugh

{arunesh, mhshin, waa}@cs.umd.edu
Department of Computer Science
University of Maryland
College Park, Maryland 20742, USA

***Abstract*—**
User mobility in wireless data networks is increasing because of technological advances, and the desire for voice and multimedia applications. These applications, however, require handoffs between base stations (or access points) to be fast to maintain the quality of the connections. In this paper, we introduce a novel data structure, the Neighbor Graph, which dynamically captures the mobility topology of the network, and we show how neighbor graphs can be utilized to reduce the authentication time of an IEEE 802.11 hand-off from 1.1 seconds (full EAP/TLS) to 50 ms without loss of security.

I. Introduction

Wireless networks, specifically those based on the IEEE 802.11 standard (Wi-Fi), are experiencing rapid growth due to their low cost and unregulated bandwidth. As a result of this tremendous growth, pockets of connectivity have been created similar to the first few years of the cellular systems. The logical next step for Wi-Fi based networks is support for fast roaming within the same administrative domain and then eventually between different administrative domains. Finally, we expect that roaming between networks of differing physical layers, i.e. vertical hand-offs, will occur once multi-mode (Wi-Fi and GSM/CDMA) handsets become more available which will in turn change how Wi-Fi networks are used.

Previous studies of wireless network mobility have shown that users tend to roam in what we call *discrete mobility* where the user utilizes the network while stationary (or connected to the same base station) and before moving the user ceases operation only to continue using the network after moving to a new location [1], [2], [3], [4]. That is the users do not usually move while using the network because the majority of current network applications and equipment do not easily lend themselves to what we call *continuous mobility*, where the user moves while utilizing the network.

Voice based applications are the pre-dominant application in *continuous mobility* as seen in the current cellular networks, and we expect voice and multimedia applications will serve as the catalyst for *continuous mobility* in Wi-Fi networks much as they did for the cellular networks once multi-mode handsets and end-user applications become more widely available.

Supporting voice and multimedia with continuous mobility, however, implies that the total latency (layer 2 and layer 3) of handoffs between base stations must be small. Specifically, the overall latency should not exceed 50 ms to prevent excessive jitter [5]. Unfortunately, the vast majority of Wi-Fi based networks do not currently meet this goal with the layer 2 latencies contributing approximately 90% of the overall latency which exceeds 100 ms [6], [7].

Logically, a wireless hand-off is composed of four phases: probe, decision, association, and authentication. In the probe phase, the mobile station seeks to identify a candidate set of next access points via active or passive means. Once the candidate set of next access points has been identified, the station selects the next access point and performs any needed housekeeping, *i.e.* flushing buffers etc., in the decision phase. Next, the mobile station begins the association phase with the selected access point. Finally, authentication or reauthentication is completed.

In this paper, we focus on improving the authentication delay incurred during a horizontal hand-off within the same administrative domain. The current draft for the IEEE 802.11 security architecture recommends that this authentication process be completed using EAP/TLS [8], and EAP/TLS has become the defacto standard by its inclusion in Windows XP. Unfortunately, a complete EAP/TLS hand-shake, including RADIUS [9] messages, requires on the order of 1.1 seconds – a number far too large to support any form of streaming media. To answer this question, the IEEE included "Pre-authentication" in the draft which permits a mobile station to "pre-authenticate" itself to the next access point (see the related work section for a more complete description). Unfortunately, pre-authentication has several short comings. First, a station can only pre-authenticate to another access point on the same local area network, i.e. the station can not authenticate beyond the first access router. This, obviously, prevents Wi-Fi networks from reaching much of the previously discussed vision.

To solve this problem, we designed, implemented, and tested a solution which only requires only small changes at the AAA server and access point and supports all of the same security properties as EAP/TLS and pre-authentication but at significantly reduced latencies. Combining our key distribution method with a novel algorithm for dynamically identifying and maintaining the mobility topology of the network, *Neighbor Graphs*, results in an efficient key distribution method that amortizes the cost of the initial EAP/TLS authentication across all hand-offs within the same administrative domain without loss of security. By using pro-active key distribution, we reduced the latency of the authentication phase from an average of 1.1 sec to an average of 50 ms[1].

II. IEEE 802.11i Authentication Overview

The authentication framework developed by the IEEE Task Group I (Security) is a complex combination of several different protocols. While a thorough understanding of each of these protocols is not required, basic knowledge of each will assist in understanding the problems we are addressing as well as our solution.

As in any architecture, the trust assumptions are key to the correct operation of the system. TGi makes the following trust assumptions:

- The AAA server is trusted.
- The access point to which a mobile station is associated is trusted– Non-associated AP's are not trusted.

These assumptions, which are different from those in a cellular network, are due to the nature of 802.11 equipment. Access points are low cost devices that are often placed in locations which lack proper physical security. Therefore, it is important to prevent the compromise of a single AP permitting a compromise of the entire network.

A. IEEE 802.1X

The IEEE 802.1X [10] standard provides an architectural framework to facilitate network access control at the link layer for various link technologies (IEEE 802.11, FDDI, Token Ring, IEEE 802.3 Ethernet, etc.). The standard abstracts the notion of three entities: the *supplicant*, the *authenticator* or the network port, and the *authentication* server. Figure 1 shows the communication setup. A *supplicant* is an entity that desires to use a service (link layer connectivity) offered via the notion of a *port* on the *authenticator* (such as a switch or an access point). Thus for a single network there will be many ports through which supplicants can authenticate themselves and obtain network access. An *authenticator* is in control of a set of ports, and a network might have multiple authenticators. As an example, an ethernet switch can be an authenticator, which controls network access on multiple physical ethernet ports available on the device. In the IEEE 802.11 scenario, a port corresponds to an association between a supplicant and the authenticator (access point).

The supplicant authenticates via the authenticator to a central *authentication server* which directs the authenticator to provide access after successful authentication. Typically the authentication server and the authenticator communicate using the *Remote Authentication Dial-In User Service* (RADIUS) protocol ([9], [11]). The RADIUS protocol contains mechanisms for per-packet authenticity and integrity verification between the AP and the RADIUS server– although these measures are not as strong as desired.

The authentication process between the authentication server and the supplicant (via the authenticator) is carried over an Extensible Authentication Protocol (EAP), which is described in the following section.

[1] The 50 ms value is for measurements made in our test-bed deployed throughout a building. We achieved an average of 20 ms in the laboratory, and have determined that the additional 30 ms delay is due to problems with the Power Over Ethernet cables used in our test-bed. We are currently repairing the problem and expect to have 20ms times for the test-bed in few weeks.

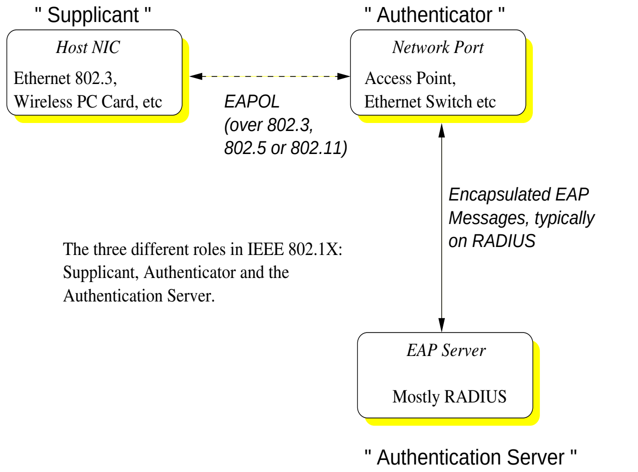

Fig. 1. *The entities in an IEEE 802.1X setup.*

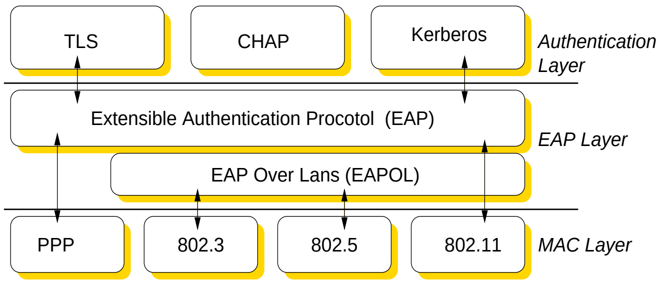

Fig. 2. *The EAP stack*

B. Extensible Authentication Protocol

The IEEE 802.1X standard employs the *Extensible Authentication Protocol* [12] to permit a variety of authentication mechanisms. Figure 2 shows the protocol layers for communication between the supplicant and the authenticator. EAP is built around the *challenge-response* communication paradigm. There are four types of messages: EAP *Request*, EAP *Response*, EAP *Success* and EAP *Failure*. The EAP Request message is sent to the supplicant indicating a challenge, and the supplicant replies using the EAP Response message. After multiple exchanges of the Request/Response messages the EAP Success/Failure message is used to notify the supplicant of the outcome. The common authentication mechanisms used are EAP-CHAP, EAP-MD5, and for our scenario EAP-TLS (discussed later).

The EAP messages do not have an addressing mechanism and are thus encapsulated. The *EAP Over Lan* (EAPOL, [10]) protocol carries the EAP packets between the authenticator and the supplicant. The EAPOL protocol also provides for the four-way handshake mechanism (discussed later). Between the authenticator and the authentication server, the EAP messages are carried over the RADIUS protocol as an attribute in a RADIUS packet.

C. Transport Layer Security

The Transport Layer Security protocol as described in RFC-2246 [13], provides strong authentication and encryption at the

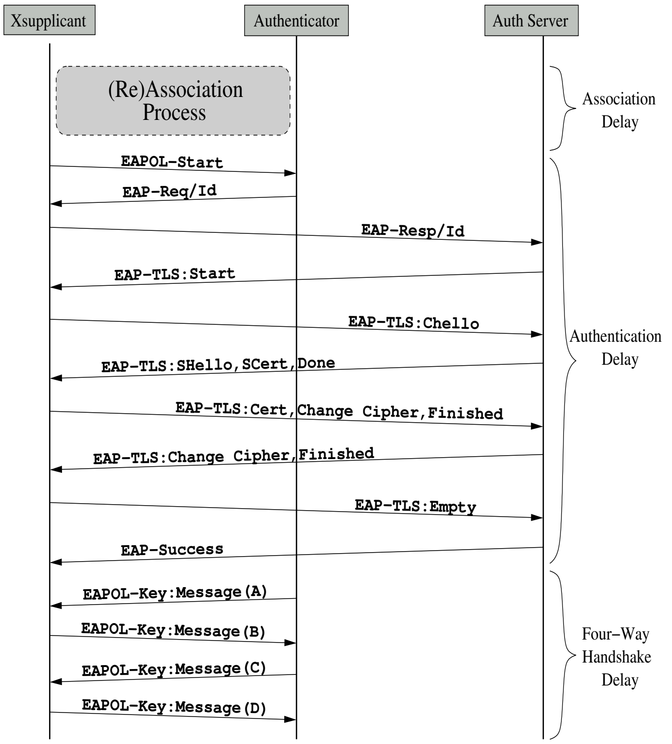

Fig. 3. *Figure shows the complete set of messages exchanged during the (re)association process. In particular, it shows the EAP/TLS authentication messages, and the four-way handshake.*

transport level. It is divided into two protocols : the *handshake* protocol which handles the communication for the authentication and derives strong key material for the data transfer which is carried over the *record* protocol. The authentication part of the TLS has been exported as an authentication mechanism over EAP in the EAP/TLS RFC2716 [8]. This is the most commonly used authentication mechanism over EAP, and fits into the IEEE 802.1X model.

In the application of TLS to IEEE 802.1X, the supplicant and the authentication server have a certificate from a common trusted certificate authority (CA). The mutual authentication process based on these credentials achieves the following: (i) mutual authentication of the client and the server, (ii) a strong shared secret master key (MK) (iii) an initialized sets of pseudo-random functions (PRFs) which can be utilized for generating further key material. Let TLS-PRF denote the PRFs generated as a result of the authentication. The MK is used to derive a Pairwise Master Key (PMK) by using equation 1.

$$PMK = \textit{TLS-PRF(MK, "client EAP encryption"} \mid \textit{clientHello.random} \mid \textit{serverHello.random)} \quad (1)$$

The PMK is used along with certain cipher methods to derive four Pairwise Transient Keys which are used various purposes as shown in figure 4 [2]. The first key EAPOL-MIC key and the EAPOL-Encr. keys are used to provide data origin authenticity and confidentiality for the four-way handshake discussed later.

[2] The interested reader is referred to [14] for a detailed description.

The other two keys are used for link layer encryption and authenticity depending on the cipher suite being employed.

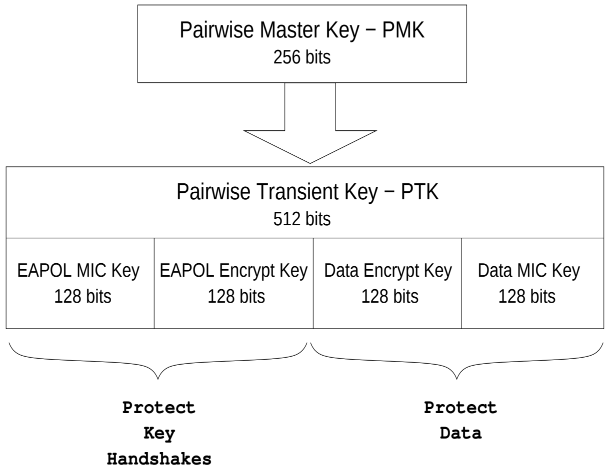

Fig. 4. *The key structure: PMK and the derived PTK.*

D. Four way hand-shake

The IEEE 802.11 Task Group I defines an IEEE 802.1X protocol called a four-way handshake. This protocol is used to confirm the liveness of the AP and the STA, guarantees the freshness and synchronizes the shared session key and binds the PMK to the MAC address of the STA. The communication is carried using EAPOL key messages[15].

1) *Message (A) Authenticator* $\longrightarrow$ *Supplicant*: This is the first EAPOL-Key message and is sent from the authenticator to the supplicant. It contains ANonce – a nonce value generated by the authenticator. Once the supplicant has received this message it can compute the four temporal keys.
2) *Message (B) Supplicant* $\longrightarrow$ *Authenticator*: This message contains SNonce – a supplicant generated nonce and a MIC over the message to protect its integrity. The authenticator uses SNonce to generate the temporal keys, and verifies the MIC.
3) *Message (C) Authenticator* $\longrightarrow$ *Supplicant*: This message includes the earlier ANonce and a MIC check which can be verified by the supplicant proving that the authenticator has a matching PMK.
4) *Message (D) Supplicant* $\longrightarrow$ *Authenticator*: This message signifies the completion of the four-way handshake and signals the installation of the keys by both entities for the data communication.

The four-way handshake protocol is used during a full-authentication and during re-authentication, and hence this cost (i.e. the overhead incurred) will be present in both situations. We also do not include the cost of the hand-shake in the timings of EAP/TLS. In this work, we do not implement the handshake for the above reason, instead we have implemented a simpler two-way handshake mechanism for demonstration purposes.

E. TGi Trust Relationships

One of the interesting, and disappointing, problems with TGi's new 802.11 security architecture are the trust relation-

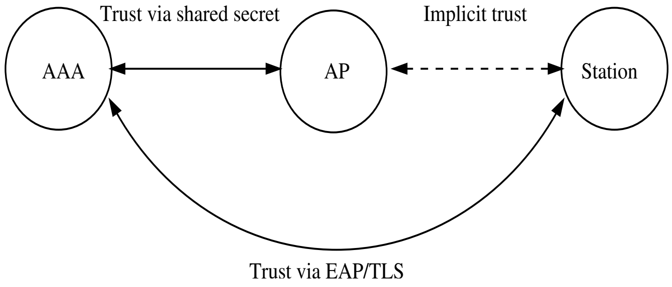

Fig. 5. *The Trust relations in TGi.*

ships in an operational network. Many people believe that the access point is a trusted party, and this isn't completely correct.

Figure 5 depicts the trust relationships within TGi. The solid arrows represent an explicit mutual trust relationship while the dotted line represents an implicit trust relationship that MUST be created in order to make security claims about the communications path. This trust relationship between the AP and the STA is transitive and derived from the fact that the station trusts the AAA server and the AAA server trusts the AP. This, unfortunately, is not ideal since in many cases the trust relationship between the AAA server and the AP will not exist if shared keys are not used to protect the RADIUS traffic. However, the majority of the AP vendors in TGi had a strong desire for an inexpensive AP and be more of a relay than a participant in the communications.

F. Properties of a Successful Authentication

After the successful completion of the EAP/TLS authentication phase the following statements hold:

1) The mobile station's identity has been proven.
2) Based on the above identity, the mobile station's access to the network has been granted by the AAA server.
3) The mobile station and the AAA server share a strong master secret, MK.
4) The mobile station, the AAA server, and the associated access point all share a common secret, pairwise master key or PMK, derived from the MK.
5) A session key, PTK, is derived from the PMK using the four-way handshake and is only shared between the mobile station and the associated access point.

III. Neighbor Graphs

In this section, we describe the notion of the neighbor graph datastructure, and the abstractions they provide. Neighbor graphs are used to determine the candidate set of access points that a roaming STA could potentially reassociate to. Usually this candidate set is a small fraction of the total number of APs forming the wireless network. Hence schemes which proactively transfer STA context and key material to this candidate set of APs prior to reassociation become feasible.

A. Definitions

Reassociation Relationship: Two APs, say, ap_i and ap_j are said to satisfy a reassociation relationship if it is possible for a STA to perform an 802.11 reassociation through some path of motion between the physical locations of ap_i and ap_j.

Consider the placement of APs in a simple in-building scenario as shown in figure 6. The dotted lines show a potential path of motion. The APs A and E satisfy the reassociation relationship, because there exists a path of motion (as can be seen from the figure) by which an STA can reassociate between A and E.

The reassociation relationship depends on the placement of APs, signal strength and other topological factors and in many cases corresponds to the physical distance (vicinity) between the APs. The reassociation relationship between APs forms the basis for the construction of the neighbor graph datastructure as discussed below.

AP Neighbor Graph: Define a undirected graph $G = (V, E)$ where $V = \{ap_1, ap_2, \ldots, ap_n\}$ is the set of all APs (constituting the wireless network under consideration), and there is an edge $e = (ap_i, ap_j)$ between ap_i and ap_j if they satisfy a reassociation relationship.

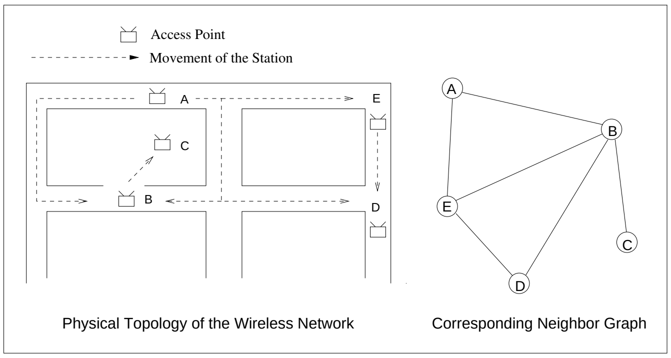

Fig. 6. *Figure shows an example placement of APs and the corresponding neighbor graph.*

Association Pattern: Define the *association pattern* $\Gamma(c)$ for client c as $\{(ap_1, t_1), (ap_2, t_2), \ldots, (ap_n, t_n)\}$, where ap_i is the AP to which the STA reassociates (new-AP) at time t_i and $\{(ap_i, t_i)\ , (ap_{i+1}, t_{i+1})\}$ is such that the handoff occurs from ap_i to ap_{i+1} at time t_{i+1}; the STA maintains continuous logical network connectivity from time t_1 to t_n.

The neighbor graph and the association pattern are related according to the following observation. We define the *Locality of Mobility* principle to state that for a client c, with association pattern $\Gamma(c)$ as defined above, the neighbor graph $G = (V, E)$ captures the *locality* (of motion) in the association pattern i.e. for any two successive APs, say, ap_i and ap_{i+1} in $\Gamma(c)$ the edge $e = (ap_i, ap_{i+1}) \in E$. This concept of locality is the abstraction captured by the neighbor graph as a datastructure.

B. Implementation Issues

The neighbor graph can be autonomously learned and maintained by a wireless network without the need for any manual configuration. Also the datastructure can be maintained either in a distributed fashion by the APs themselves [16], or in a centralized manner at the authentication server as in this paper. In this application of neighbor graphs for proactive key-distribution, we construct and maintain the datastructure at the authentication server (RADIUS).

1) *Edge Creation*: Edges can be created either on the receipt of an 802.11 *reassociation request* frame by an AP or explicitly by APs themselves on re-authentication. Also if the APs implement the IEEE 802.1f Inter-Access Point Protocol, the receipt of a *Move-Notify* message can also induce an edge in the graph.
2) *Edge Deletion*: Unused and stale edges (i.e. reassociations paths which rarely occur) can be deleted over time in an LRU fashion. This is necessary in order to delete incorrectly added edges. One situation where this could happen is a client that goes into the power save mode, and potentially wakes up in a different location to reassociate to any arbitrary AP on the wireless network.

The autonomous generation also eliminates the need for any survey or other manual construction methods. As a result, this also makes the datastructure adaptive to changes in the reassociation relationship which might occur because of topology changes (i.e. changes in AP placements, physical topology, etc).

IV. Pro-active Key Distribution

Pro-active key distribution seeks to reduce the latency of the authentication phase by pre-distributing key material ahead of a mobile station. Our approach provides all of the same properties of a full EAP/TLS authentication, but at significantly less cost in terms of latency and computational power of the mobile station.

A. PMK Trees

In the current, 802.11i framework the PMK is derived from the MK by equation 1. Pre-distributing this PMK, which is currently permitted in the current TGi draft as PMK caching, violates the current TGi trust assumptions[3]. Rather than pre-distribute this PMK, we change the derivation of the PMK to the recurrence shown in equation 2, where n represents the n^{th} reassociation for $n >= 0$.

$$\begin{aligned} PMK_0 &= \textit{TLS-PRF(MK,"client EAP encryption"} \mid \\ &\quad \textit{clientHello.random} \mid \textit{serverHello.random)} \\ PMK_n &= \textit{TLS-PRF(MK, } PMK_{n-1} \mid \textit{AP\_MAC} \\ &\quad \mid \textit{STA\_MAC)} \end{aligned} \tag{2}$$

The recurrence shown in equation creates a *PMK* tree with the reassociation pattern, $\Gamma(STA)$, a path within the tree as shown in figure 7. In figure 7, the reassociation pattern is $\Gamma(STA) = A, B, C, D$.

B. PMK Synchronization

There are two conditions that can exist when a mobile station arrives at an access point with respect to the pre-distribution of the correct PMK: either the AP and the mobile station share the same PMK, or they do not. The handshake (two-way in our case and four-way in the case of TGi) determines which of these cases exist. This also ensures both *liveness* and *freshness* of the key.

[3]Yes. TGi knows they are doing this.

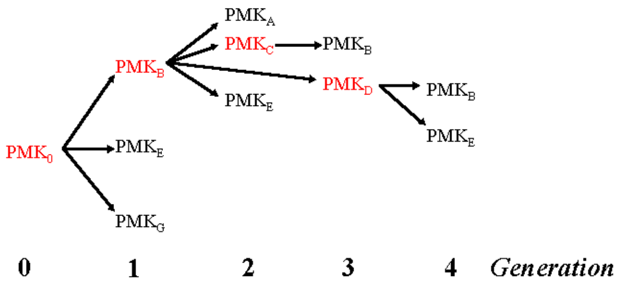

Fig. 7. *PMK tree*

C. PMK Distribution

Once a mobile station completes an initial full EAP/TLS authentication as denoted by by the AAA server sending an *ACCESS-ACCEPT* message to the access point indicating successful completion of the authentication process as well as PMK_0. At this point, the AAA server and the mobile station share the MK, and the AAA server, the access point, and the mobile station all share PMK_0. The AAA server now determines the neighbors of the associated access point and sends a *NOTIFY-REQUEST* that a specific mobile station may roam into the coverage area of each of the neighboring access points [17]. This message is advisory only, and an access point may or may not decide to request the security association, or PMK from the AAA server at this time. If the AP does decide to request the PMK, then the AP sends a *NOTIFY-ACCEPT* message. If not, then the AP sends a *NOTIFY-REJECT* message to the AAA server. Upon receiving the *NOTIFY-ACCEPT* message, the AAA server responds with an *ACCESS-ACCEPT* message which contains the appropriate PMK as well as authorization for the mobile station to remain connected to the network.

D. Two-way handshake

After the key distribution, the four-way handshake (discussed earlier) confirms the freshness of the keys being used by the AP and the roaming STA. In our implementation, we used a simpler two-way handshake (an EAPOL start message, and an EAP-Success message if the AP has the correct key) for purposes of demonstration. Since the four-way handshake is performed during both – a full authentication and the fast re-authentication, it does not effect the key distribution scheme.

V. Implementation and Experimental Results

In this section, we present implementation results to demonstrate the performance of the proactive key distribution scheme. We have implemented the fast re-authentication (using the key distribution scheme) and the standard full-authentication over an in-building wireless testbed network comprising of 9 access points spread over three floors. Since the four-way handshake process appears in both schemes after the key has been delivered, we did not implement the full version and we instead implemented a simple 2-way handshake to verify the key freshness. We measured 90 full EAP-TLS authentication latencies which result in an average of approx. 1.1 seconds. Using the proactive key distribution scheme for fast re-authentication we obtained an average latency of 48 ms (a 99.6% reduction). Also we measured the overhead incurred by two additional messages between the RADIUS server and the authenticator. With eight neighbors to distribute the key, the overhead was approx. 21 ms on average [4].

A. The Implementation

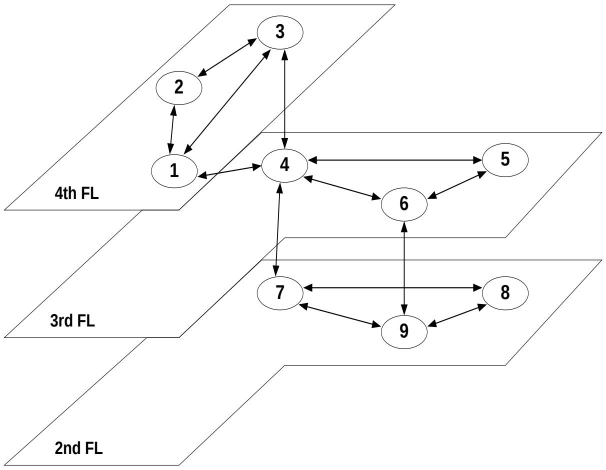

Fig. 8. *Figure shows the topological placement of the APs in our wireless testbed and the resulting structure of the neighbor graph.*

The wireless testbed network spans three floors (2nd, 3rd and 4th) of a university building and consists of nine APs as shown in figure 8. The access point is based on a NET4521 *Soekris* [18] board, which has a 133 MHz AMD processor, 64MB SDRAM, two PC-Card/Cardbus slots for wireless adapters and one *CompactFlash* socket. The board is powered using *Power Over Ethernet* through the ethernet cable. A 200mW Prism 2.5 based wireless card is used as the AP interface with a 1ft *yagi* antenna. OpenBSD 3.3 with access point functionality is used as the operating system.

The supplicant and the authenticator software is based on the *open1x* [19] implementation built here. We also use the *Freeradius* [20] software for the RADIUS server, modified to implement the key distribution scheme and maintain the neighbor graph datastructure. The RADIUS server is installed on a backend machine (PIII 551.247 MHz, 128 MB RAM). The *Xsupplicant* and the *authenticator* [19] software was modified to include the simple two-way handshake instead of the four-way handshake for purposes of demonstration.

B. Experimental Results

The experimental setup consisted of a supplicant roaming in the wireless testbed. A laptop with PIII 1.8 GHz, 256 MB RAM and a Prism 2.5 based *DemarcTech* wireless card [21] is used as the supplicant. Three experiments were done to measure three different latencies as detailed below:

[4]Note that this overhead plays no role in the re-authentication latency, and just adds to the load on the RADIUS server. We include it here for the sake of completeness.

1) *Measuring Full-authentication Latency:* The supplicant was made to roam from one AP to another in the wireless network, and a full IEEE 802.1X EAP TLS authentication was performed at each reassociation. We measured 90 such authentications resulting in an average latency of 1.1 seconds.
2) *Fast Re-authentication:* Fast re-authentication using proactive key distribution was enabled on the RADIUS and the authenticators. The RADIUS server was initialized with the neighbor graph shown in figure 8. We use a static neighbor graph for ease of demonstration. The graph used in our experiments was constructed by human observation of the reassociation messages. Autonomous construction methods detailed earlier should be used in order to keep the neighbor graph fresh and dynamic and this has no effect on the performance of the key distribution scheme. Figure 9 shows the authentication latencies. The first authentication (which occurs at the start of a session), is a full-authentication and hence incurs a high latency (approx. 800 ms); while all subsequent 18 re-authentications reflect the latency of the two-way handshake.
3) *Overhead at the RADIUS server:* In this experiment we measured the additional overhead incurred by communication required for distributing the keys proactive using the *Notify-Request*, *Notify-Accept* and the *Access-Accept* messages. We measured 80 authentications and obtained an average latency of 21 ms. This overhead does not increase the handoff latency.

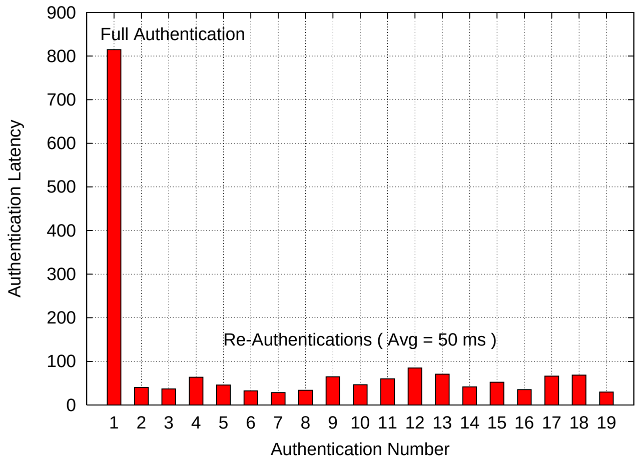

Fig. 9. *Figure shows the authentication latencies as observed by the roaming supplicant in the wireless testbed, with proactive key distribution enabled. As can be seen, the first authentication reflects the full-authentication latency and initiates the key distribution mechanism.*

VI. Related Work

Pack [22], [23] proposes a fast handoff scheme using a *predictive* authentication method based on IEEE 802.1X model. In their scheme, pre-authentication is performed to the k most likely next access points. The k stations are selected using a weighted matrix representing the likelihood (based on the analysis of past network behavior) that a station, associated to AP_i, will move to AP_j. The mobile station may select only the most likely next access points to pre-authenticate, or it may select all of the potential next access points [22], [23]. Pack uses the notion of a frequent handoff region (FHR) to represent the adjacent access points which is obtained by examining the weighted matrix. The weights within the matrix are based on an $O(n^2)$ analysis of RADIUS log information using the inverse of the ratio the number of handoffs from AP_i to AP_j to the time spent by the mobile station at AP_i prior to the handoff. In the paper [22], pre-authentication means the following. When a station authenticates to AP_i, authentication server (AAA server) sends security information not only to AP_i but also to other APs in FHR. As a consequence, the next handoff to one of APs in FHR does not require any message exchanges between the AP and the AAA server, because the AP already has the security information.

There are several issues with pre-authentication. Firstly, pre-authentication can not occur beyond the first access router due to the fact that EAPOL packets are used to carry authentication information. This severely limits the ability to pre-authenticate to single LANs only and prohibits WAN and Inter-network roaming. Secondly, the cost of a full reassociation is prohibitive for a capable device as in the laptop used in our experiments. Imagine the times for a small handset using a low powered processor. In addition, the authentication process must be accomplished to each potential neighbor. Thus, the cost is several seconds rather than milli-seconds. During the authentication time, by the way, the mobile station is on a different channel and unable to process traffic from or from the currently associated access point. Finally, unless there is a significant overlap in coverage pre-authentication will just not work due to the length of times cited earlier.

For the construction of FHR matrix, it requires $O(n^2)$ computation and space, where n is the number of access points in the network, and must be created at the authentication server (AS). Furthermore, the FHR notion does not quickly adapt to changes in the network topology. This is in contrast to our neighbor graphs which require $O(degree(ap))$ computation and storage space per AP and which quickly adapt to changes in the network topology. Additionally, neighbor graphs can be utilized either in a distributed fashion at each access point, or client, and in a centralized fashion at the AS.

VII. Conclusions

Wireless networking has changed considerably over the last decade, and the next decade will likely see the ubiquity of wireless network service achieved. Accomplishing this goal will require the inter-working of different administrative domains, and different physical layers. If Wi-Fi networks are to be participants in this vision, then the current hand-off latencies must be reduced significantly.

In this paper, we presented a novel data structure, neighbor graphs, along with the addition of new messages to RADIUS that enables the pre-distribution of the PMK ahead of a mobile station. We also demonstrated this approach provides the same level of security as a 1.1 second full EAP/TLS authentication, but at a significantly lower latency 20 ms as shown by laboratory and 50 ms as shown by test-bed experiments.

References

[1] D. Tang and M. Baker, “Analysis of a metropolitan-area wireless network,” in *Mobile Computing and Networking*, pp. 13–23, 1999.

[2] K. Lai, M. Roussopoulos, D. Tang, X. Zhao, and M. Baker, “Experiences with a mobile testbed,” in *Proceedings of The Second International Conference on Worldwide Computing and its Applications (WWCA'98)*, Mar 1998.

[3] A. Balachandran, G. Voelker, P. Bahl, and P. Rangan, “Characterizing user behavior and network performance in a public wireless lan,” 2002.

[4] M. Balazinska and P. Castro, “Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network,” in *International Conference on Mobile Systems, Applications, and Services (MobiSys)*, May 2003.

[5] International Telecommunication Union, “General Characteristics of International Telephone Connections and International Telephone Circuits.” ITU-TG.114, 1988.

[6] A. Mishra, M. Shin, and W. Arbaugh, “An empirical analysis of the ieee 802.11 mac layer handoff process,” in *Computer Communications Review (ACM SIGCOMM) (To Appear)*, 2003.

[7] R. Koodli and C. Perkins, “Fast Handover and Context Relocation in Mobile Networks,” *ACM SIGCOMM Computer Communication Review*, vol. 31, October 2001.

[8] B. Aboba and D. Simon, “Ppp eap tls authentication protocol,” *RFC 2716*, October 1999.

[9] C. Rigney, W. Willats, and P. Calhoun, “Remote Authentication Dial In User Service (RADIUS).” RFC 2869, June 2000.

[10] IEEE, “Standards for local and metropolitan area networks: Standard for port based network access control,” *IEEE Standard P802.1X*, October 2001.

[11] C. Rigney, W. Willats, and P. Calhoun, “Radius extensions,” *RFC 2869*, June 2000.

[12] L. Blunk and J. Vollbrecht, “Ppp extensible authentication protocol (eap),” *RFC 2284*, March 1998.

[13] T. Dierks and C. Allen, “The tls protocol version 1.0,” *RFC 2246*, January 1999.

[14] J. Edney and W. A. Arbaugh, *Real 802.11 Security*. Addison Wesley, 2003.

[15] IEEE, “Draft amendment to standard for telecommunications and information exchange between systems-lan/man specific requirements. part 11: Wireless medium access control and physical layer(phy) specifications: Medium access control (mac) security enhancements.,” *IEEE Standard 802.11i*, May 2003.

[16] A. Mishra, M. Shin, and W. Arbaugh, “Context caching using neighbor graphs for fast handoffs in a wireless network,” *CS Tech Report Number CS-TR-4477, University of Maryland, College Park*, 2003.

[17] W. A. Arbaugh and B. Aboba, “Experimental Handoff Extension to RADIUS.” Internet-Draft, May 2003.

[18] “Soekris Engineering.” URL: http://www.soekris.com.

[19] “An Opensource Implementation of the IEEE 802.1X standard..” URL: http://www.open1x.org.

[20] “The FreeRADIUS Server Project.” URL: http://www.freeradius.org.

[21] “Demarc Technologies Group.” URL: http://www.demarctech.com.

[22] S. Pack and Y. Choi, “Fast Inter-AP Handoff using Predictive-Authentication Scheme in a Public Wireless LAN,” *IEEE Networks 2002 (To Appear)*, August 2002.

[23] S. Pack and Y. Choi, “Pre-Authenticated Fast Handoff in a Public Wireless LAN based on IEEE 802.1x Model,” *IFIP TC6 Personal Wireless Communications 2002 (To Appear)*, October 2002.