

Secure Access Control for Location-Based Applications in WLAN Systems

YounSun Cho, Michael Goodrich and Lichun Bao

1 Introduction

With the growing number of high speed wireless portable devices, wireless LAN (WLAN) systems based on IEEE 802.11 a/b/g are becoming the prevailing access technologies in the commercial industries. Among the services offered through WLANs, location-based service provisioning is of great interests to wireless Internet service providers (WISPs) to deliver value-added services, such as service advertisements, product marketing, to the network users according to their geographic locations. Unfortunately, WLAN systems are vulnerable to misuses and abuses because of their open-air transmissions in untethered environments, and easily become the targets of free-riders and attackers. Therefore, a natural resort in WLAN systems is to exercise network access control to authenticate network access according to various user certificates.

User identity-based access control is a promising approach. A user's identity [30] can be based on a password, a token, a ticket, an administered access control list (ACL), or biometrics [17, 23]. SecureID is a token-based authentication scheme for a user remotely logging into corporate networks using a combination of both a password and a random number token [29]. Kerberos is another widely used ticket-based network authentication protocol today [17]. It is designed to provide strong authentication for client/server applications based on the secret key cryptography, which is derived from the Needham-Schroeder key distribution protocol [23]. Likewise, the access control list (ACL) is commonly used for access control by modern operating systems [32].

However, identification-based access control does not satisfy certain security requirements that depend on user information such as the location as we mentioned before. Moreover, identification-based approaches may require user-agreement, key distribution, communication overheads in order to procure the related identities.

Various location sensing schemes have been reported. Usually, the location information is derived from direct interactions between the infrastructure network and the wireless mobile devices. One approach is to estimate the position of a given source based on the received signal strength. A variety of ranging and positioning techniques with different technologies such as RF, ultrasound or infrared, have been proposed to solve this problem [9, 12].

The Cricket system is a decentralized indoor location-support system that requires the combination of RF (radio-frequency) and ultrasonic signals in order to trace user locations and to provide location services to users and applications [26]. PAC is based on the Cricket system for location tracing, and adopts the INS/Twine [4] architecture for scalable resource discovery [22]. In PAC, The client first acquires a Location ID (LID) along with a time-varying Location Code (LIDCODE) from its surrounding access points' beacons, then sends them to a location authentication server to get a service-granting ticket. PAC

requires synchronization between beacons and the location authentication server, and keeps track of the corresponding LIDCODEs as they change with time.

Sastry *et al.* described an Echo protocol to compute node location based on the round-trip latency of messages and ultrasonic signals in location computations [28]. They proposed the concept of *Region of Acceptance* in order to combat malicious location-provers from submitting location claims that overstate the true processing delay. Water *et al.* proposed a similar protocol for proving the location of tamper-resistant devices, based on the exchange RF messages [34].

Zhang *et al.* [35] proposed the use of location-based keys using identity based public-key cryptography (ID-PKC) , which solves the Bilinear Diffie-Hellman Problem (BDHP) [6, 5].

Although exact location information meets the goal of location-based access control mechanisms, such localization is not required in many cases. Instead, coarse location information, such as the areas enclosed within airport, Internet cafe, hotel *etc.*, is sufficient to provide location-based access control. These areas can be easily defined by a set of access points, and are much more static than the previous ones. In these coarse location scenarios, we suggest that the access to a WLAN system is granted if and only if the clients are located within the areas concurrently covered by multiple access points. Using sectored antennas, we can further specify the desired shapes of the areas.

We propose a location authentication and network access authorization protocol, called LBAC (Location-Based network Access Control), based on coarse location information. LBAC securely authenticates the location claims of mobile wireless users in the areas, then securely distributes the shared keys for data encryption purposes. In LBAC, the location areas are defined by the shared coverage of multiple wireless access points (APs). The fact that a mobile node is located at certain places is proved by the mobile node collecting and presenting all the key information from the corresponding access points. Using Diffie-Hellman algorithm, LBAC authenticates location claims, and derives the shared keys for encryption purposes between each mobile node and access point pair. LBAC eliminates the dependence on Global Positioning System (GPS) or ultrasonic devices in order to localize the mobile devices. We enumerate possible attacks to the system and analyze their countermeasures. The computational, communication, and the memory requirement are evaluated, and further validated using simulations.

The rest of the chapter is organized as follows. Section 2 provides an overview of the algorithms and protocols used by LBAC. Section 3 specifies the network assumptions and the protocol operations of LBAC. We evaluate the efficiency and security features of LBAC in Section 4. Section 5 summarizes the chapter.

2 Background Review

2.1 Diffie-Hellman Key Exchange

Diffie-Hellman key exchange scheme is used to agree on a shared key, K_{AB} , securely between two parties/nodes A and B [8, 21].

In Diffie-Hellman algorithm, two publicly-known numbers are distributed beforehand — a prime number p , and a generator g of the cyclic group Z_p^* . In order to derive a shared key between nodes A and B , node A chooses random private key value $X_A \in Z_{p-1}$, then computes a public key value $Y_A = g^{X_A} \bmod p$. Similarly, node B chooses random private key value $X_B \in Z_{p-1}$, and computes $Y_B = g^{X_B} \bmod p$. Nodes A and B now exchange the public keys Y_A and Y_B of each other explicitly, and derive the shared secret key

using the modulo arithmetic as follows:

$$K_{AB} \equiv Y_A^{X_B} \equiv g^{X_A X_B} \equiv Y_B^{X_A} \equiv K_{BA} \pmod{p}. \quad (1)$$

Once the shared key is established, secure communication between two parties can be established using any symmetric key encryption scheme, like Data Encryption Standard (DES).

2.2 IEEE 802.11i Overview

After the interim WEP (Wired Equivalent Privacy) standard in the original IEEE 802.11 [1], the IEEE 802.11i standard was released to address the WEP's weaknesses [2]. The IEEE 802.11i separates the user authentication process from the message protection process in order to meet the goals of RSN (Robust Security Network) [18, 10]. It contains the following components:

1. Authentication protocol, which defined two modes of authentications: IEEE 802.1x EAP (Extensible Authentication Protocol) mode and the pre-shared key (PSK) mode. EAP is required in IEEE 802.11i.
2. AES-based encryption protocol, CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol), to provide confidentiality, integrity and origin authentication.

EAP was a port-based access control mechanism specified in IEEE 802.1x [3], which was originally designed for the Point-to-Point Protocol (PPP), as in MODEM connections and wired LANs. EAP requires an authentication server, such as a RADIUS (Remote Authentication Dial In User Service) server, and is extensible to support several other authentication protocols.

PSK mode does not require an authentication server, but requires an static pre-shared keys between access points and mobile stations. A pairwise master key (PMK) is obtained directly from a pre-shared key (PSK) with pseudo-random functions.

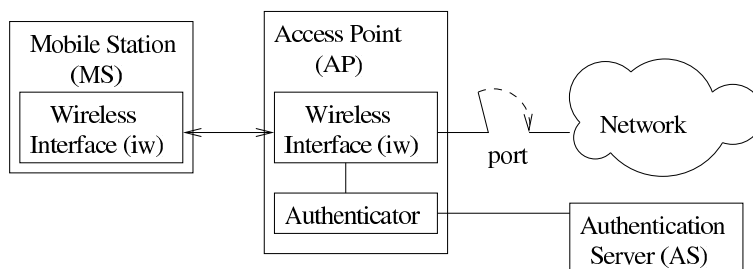


Figure 1: Three Entities for Port-Based Access Control in 802.11i.

The EAP authentication process in IEEE 802.11i involves the three entities — the mobile station (MS), the access point (AP) and the authentication server (AS) as shown in Figure 1. The AS resides in the network, and the MS, who initially does not have access to the network, is connected to the AP. The AP initially blocks the MS's access to the network, and also serves as a broker between the MS and the AS during the authentication process. Only after the MS is authenticated by the authenticator on the AP to the AS, can the MS access the network. The IEEE 802.1x EAP exchange provides the shared PMK (Pairwise Master Key).

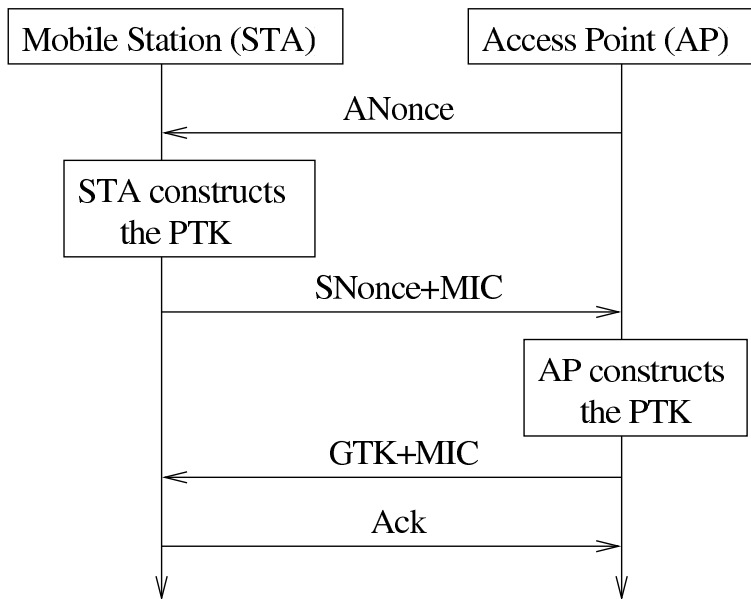


Figure 2: Four Way Handshakes for Mutual Authentication and Key Generation.

However, the PMK is designed to last for the entire session, and should be exposed as little as possible. Therefore, a four-way handshake is used to establish another key called the PTK (Pairwise Transient Key), and to authenticate the access point (AP) to the mobile station (STA), as shown in Figure 2. The PTK is generated through a cryptographic hash function with the concatenated product of the following attributes: PMK, AP nonce (ANonce), STA nonce (SNonce), AP MAC address and STA MAC address. The four-way handshake works as follows:

1. The AP sends a nonce-value to the STA (ANonce), which now has all the attributes to construct the PTK.
2. The STA sends its own nonce-value (SNonce) to the AP together with an MIC (message integrity code).
3. The handshake also yields the GTK (Group Temporal Key), used to decrypt multicast and broadcast traffic. The AP sends the GTK and a sequence number together with another MIC. The sequence number is the sequence number that will be used in the next multicast or broadcast frame, so that the receiving STA can perform basic replay detection.
4. The STA sends a confirmation to the AP.

The pairwise transient key (PTK) is then divided into three separate keys: 1) EAPOL-Key Confirmation Key (KCK) to compute the MIC for EAPOL-Key packets, 2) EAPOL-Key Encryption Key (KEK) to encrypt the EAPOL-Key packets, and 3) Temporal Key (TK) to encrypt the actual wireless traffic.

3 Location-Based Access Control (LBAC)

3.1 Network Assumptions

We address the network access control problem in infrastructure-based WLAN systems based on IEEE 802.11 [1], which involves two types of elements: the access points (APs) and the mobile stations (MSs). In addition, for authentication and key distribution purposes, we have another type of node, called the key server (KS), which provides similar functionalities as the authentication server in IEEE 802.1x.

We assume that the network access points are purposefully deployed such that the desired access-granted areas are covered by multiple access points. Specifically, the access-granted areas can be custom-made into special shapes according to customer requirements using directional antennas by adjusting the angle and distance of the signal propagation [25].

We also assume that mobile devices have sufficient computational and communication capacities to carry out the simple cryptographic operations required in LBAC.

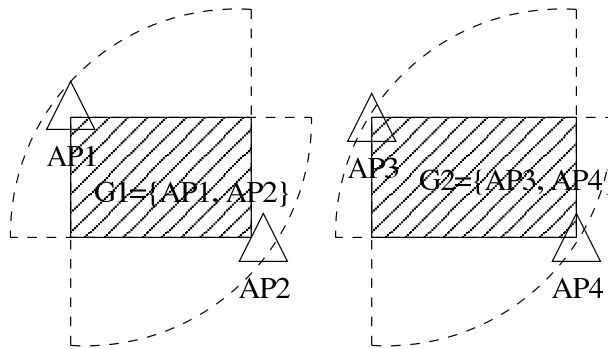


Figure 3: Defining Access-Granted Network Areas.

In Figure 3 illustrates two access-granted areas as defined by the shaded areas of two AP groups $G_1 = \{AP_1, AP_2\}$ and $G_2 = \{AP_3, AP_4\}$, where the directional antenna of each access point spreads 90° . The set of access points that cover an access-granted area is called the *location group* of the access-granted area. The location groups that an access point belongs to are designated by the network administrators or an automated bootstrapping process when the WLAN system is initially designed and deployed.

3.2 LBAC Protocol Operations

LBAC uses the Diffie-Hellman key exchange scheme for user location authentication, network-access authorization and data encrypting key distribution purposes. We employ the key server for the key exchange and location group management purposes.

Figure 4 shows the location-based access control architecture using an example WLAN system, which includes the three elements – mobile stations (MSs), access points (APs) and a key server (KS). In particular, Figure 4 illustrates that the mobile station MS is located in the access-granted area confined by the location group $G_L = AP_1, AP_2, AP_3$. The key server KS connects to each AP by *low-latency* and *secure* connections, provided by the infrastructure networks. The LBAC operational steps are marked by numbers, and the message flows are indicated by arrows. For simplicity, we omitted the location group descriptor in all of the messages in Figure 4.

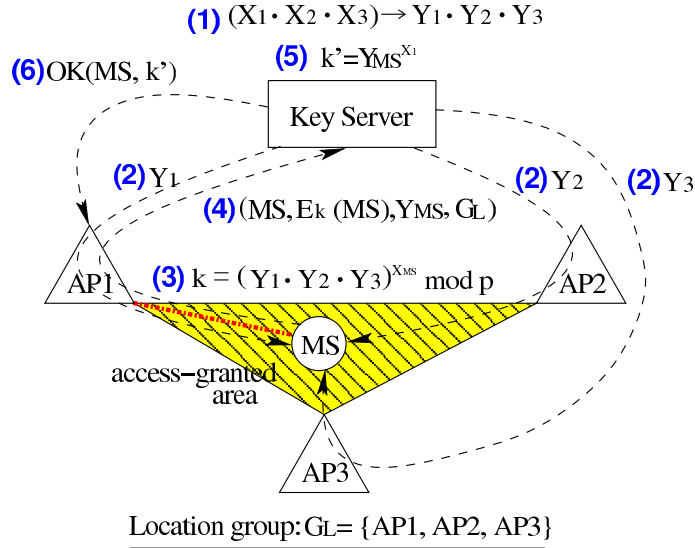


Figure 4: Key Distribution Protocol in LBAC.

The LBAC key server is the central point of control in the location-based access control process. We describe LBAC protocol operations in two phases according to Figure 4: the authentication phase in steps (1)-(5), and the key generation phase in step (6).

LBAC Authentication Phase

- (1) The key server generates three private keys $\{X_1, X_2, X_3\}$, and derives the corresponding public keys $\{Y_1, Y_2, Y_3\}$ using the Diffie-Hellman algorithm for the APs in location group $G_L = \{AP_1, AP_2, AP_3\}$, respectively. The public key Y_i is called the *location key* of the corresponding AP i . The private keys of the location keys are kept as secrets by the key server. The location keys are generated and distributed to the APs, periodically.
- (2) After receiving these location keys, the APs broadcast these keys via their beacon messages within their wireless cell or BSS (basic service set). When a mobile station is located within the access-granted areas, the station can listen to the channel and gather these location keys of the location group.
As shown in Step (2) of Figure 4, the MS collects the public keys $\{Y_1, Y_2, Y_3\}$ of the location group $G_L = \{AP_1, AP_2, AP_3\}$.
- (3) The MS derives the location claim key k by multiplying all the location keys from the APs of the location group G_L , then raising the product to the power of X_{MS} using modulo arithmetic, which gives the location claim key k as

$$k = (Y_1 \cdot Y_2 \cdot Y_3)^{X_{MS}} \equiv g^{(X_1 + X_2 + X_3) \cdot X_{MS}} \pmod p.$$

Such computation is similar to the shared key computation in the Diffie-Hellman algorithm, but with a little complication to the base. Such scheme first appeared similarly in [31]. X_{MS} is the private key kept by the mobile station MS .

- (4) After deriving the location claim key k , the MS composes the location claim with a four-element tuple $(MS, E_k(MS), Y_{MS}, G_L)$, which includes the MS identifier MS , the encrypted mobile station

ID $E_k(MS)$ using the location claim key k , the current MS public key Y_{MS} derived from X_{MS} , and the location group descriptor G_L . The location claim is sent to the key server via the currently contacted or associated access point, which is AP_1 in Figure 4.

The purpose of sending MS and its encrypted form $E_k(MS)$ is to show the key server that the MS has derived the location claim key, and can encrypt the plain-text using the key. A more complicated location claim is to include the BSS time stamp and location group indicator G_L in the encrypted message so as to defend against the message-replay attack.

- (5) The key server exams the location claim of the MS by deriving the same location claim key k . According to the location claim tuple, the key server first retrieves the private keys $\{X_1, X_2, X_3\}$ of the location group G_L , which were generated in Step (1) originally. Then the key server computes k using

$$k = Y_{MS}^{X_1+X_2+X_3} \equiv g^{X_{MS} \cdot (X_1+X_2+X_3)} \pmod{p}.$$

Then the key server KS encrypts MS using the key k . If the result is the same as $E_k(MS)$, the key server asserts that the MS has received the location keys of the location group G_L , and is located in the corresponding access-granted area. Therefore, the MS location claim is authenticated. Otherwise, the location claim is invalid, and data packets from and to the MS will be blocked in the future.

LBAC Key Generation Phase

- (6) If the MS location claim is authentic, the key server will generate the shared key k' for encrypting the AP-MS communication by again using the Diffie-Hellman algorithm. The shared key k' between MS and AP_1 in Figure 4 is

$$k' = Y_{MS}^{X_1} \equiv g^{X_{MS} \cdot X_1} \pmod{p},$$

in which, the public key belongs to the MS , and the private key belongs to the associated AP of the MS. The associated AP can be easily derived from who forwarded the location claim, or be explicitly indicated in location claim messages, which was omitted in Figure 4.

Afterward, the key server directly sends the key k' to AP_1 , along with an OK message for the AP to communicate with MS .

Similarly, the MS would have derived the same key k' from its collected key information. In Step (3) of Figure 4, after MS receives the three location keys $\{Y_1, Y_2, Y_3\}$ from the APs, the key k' is derived by

$$k' = Y_1^{X_{MS}} \equiv g^{X_1 \cdot X_{MS}} \pmod{p}.$$

Therefore, the MS and its corresponding AP can use the symmetric key k' for secure data communication later.

3.2.1 Stronger Location Authentication

Although the capability to gather all the location keys of a location group provides the evidence that an MS is located with the corresponding access-granted areas, it is still possible that the MS walks out of the access-granted area, but is still able to communicate with the associated AP. In this case, the key server has no means to detect the errand.

We propose two of many possible approaches to improve the strength of location-based authentication schemes:

- Make the MS to iteratively connect with each AP of the location group. This way, the connectivity between the MS and all the APs will be tested out in order to authenticate the true location of the MS. The iterative association with different APs can be achieved by the key server instructing the currently associated AP to explicitly disassociate with the MS, and disallow the MS to associate with the same AP again shortly after. Such policy forces the MS to connect with other APs available in the location group. However, if implemented without changing the IEEE 802.11 management functions, such physical location authentication may cause unnecessary control overhead, and could interrupt on-going communication sessions unexpectedly. Therefore, additional mechanisms are necessary to make such scheme work seamlessly, and we address this issue in the future work.
- The location authentication can be improved by renewing the location keys of the location groups periodically, thus forcing the MSs to re-authenticate themselves with the key server, and derive new shared keys for data communication purpose. We implement this approach in our simulations for LBAC evaluation purposes.

4 Evaluations

4.1 Efficiency Estimation

Our LBAC (Location-Based Access Control) protocol does not require special hardware such as GPS or ultrasonic devices for localizations, therefore providing low-cost efficient way to location-based access control operations.

As far as we can see, the overhead of operating LBAC protocol comes from the storage, computation and communication complexities. With regard to the storage complexity, we consider the three elements of a typical access controlled WLAN systems.

- The *key server* is required to store all the private keys of the APs in the WLAN system for the Diffie-Hellman algorithm computations, as well as the location group information for access control purposes. Using the private keys and the location group information, the key server can derive all other information, such as the location keys of APs and location claim keys. Therefore, the space requirement for the key server is linear to the numbers of APs and location groups in the WLAN system.
- Each *AP* is required to store *only one* public key, regardless of how many location groups the AP is in. Furthermore, each AP stores the shared key between the AP and their associated MSs. Therefore, the storage requirement for an AP is constant for its own public key, and linear for the number of MSs associated with the AP.
- The *MSs* are required to store the public keys of their respective location groups, and the shared keys between themselves and their associated APs. Therefore, the memory space requirement for each MS is linear to the size of the location group that the MS belongs to.

With regard to the computation complexity in LBAC, APs require no additional computations other than their normal 802.11 functions. However, the *key server* is required to generate the public keys for the APs, and authenticate each MS of the WLAN system, therefore its computation task increases linearly with

the number of APs and MSs in the system. The *MSs* has constant computation overhead in each location group.

With regard to the communication complexity, the complete LBAC protocol operations involve six steps as illustrated in Figure 4, in which case four messages went through the wireless interfaces, and five messages through the wired infrastructure network for authenticating the particular single MS. Therefore, the communication overhead is linear to the total number of MSs and APs in each round of the basic LBAC operations. However, because that the public keys for the APs can be broadcast piggybacked with 802.11 Beacon messages of the corresponding APs and that the MSs can send back their location claims piggybacked with 802.11 ReAssociation messages, the communication overhead is negligible over the air in WLAN systems.

Moreover, LBAC reduces the computation and communication overhead as compared with previous systems [19, 20, 7, 11] because LBAC adopts the Diffie-Hellman key distribution protocol, and does not require any pre-deployment phase for location authentication and key generation purposes.

4.2 Experimental Performance

In order to evaluate the effectiveness of LBAC in more practical scenarios, we implemented LBAC in NCTUns v4.0 [33], which was produced by SimReal Inc.. The reason what we chose NCTUns v4.0 was that it provides comprehensive and realistic simulations of IEEE 802.11 standards and other networking protocols. The essential network management functions complies with the standards to the fullest extent, such as WLAN channel estimation, probing, association, disassociation, re-association procedures, which were mostly not provided in other simulators, such as NS2[27]. In addition, NCTUns v4.0 directly integrates with existing command line programs, such as FTP, HTTP applications, so that the simulator evaluates real application performance in its simulated networks.

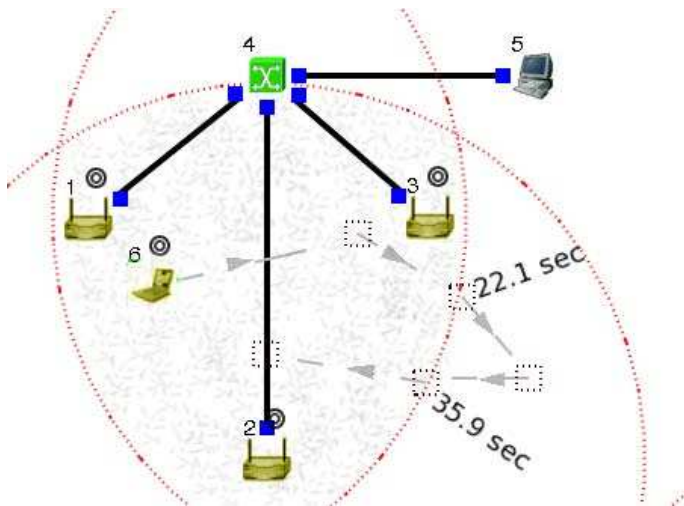


Figure 5: Simulation Setup

Figure 5 illustrates our simulated scenario, in which three APs, nodes 1, 2, 3, form a location group $\{AP_1, AP_2, AP_3\}$, and the access-granted area is defined by the triangular area filled with a random pattern. The key server module runs on the switch, node 4, in Figure 5. All wireless links operate at 11 Mbps data rate.

The LBAC protocol starts with the key server identifying the APs, which form the location group $\{AP_1, AP_2, AP_3\}$. Afterward, the key server at the switch periodically generates and distributes individual keys $\{Y_1, Y_2, Y_3\}$ to the corresponding APs, which in turn broadcast the keys in their beacon messages. In IEEE 802.11, the beacon period is 100 ms by default, which guarantees prompt delivery of the keys to the MS in the access-granted areas. We set the location key refresh period to 5 seconds on the key server. The location claims are carried in 802.11 ReAssociation messages sent by the MSs every second periodically.

In order to test the effects of LBAC mechanisms to data traffic, we simulated two traffic patterns — one TCP stream and one CBR stream, respectively, from the MS (node 6) to the fixed host (node 5) in Figure 5, and created a mobility pattern such that there are two critical points in the simulations — the MS moves out of the access-granted area at 22.1 second, and comes back into the area at 35.9 second. And the simulations were carried out with and without LBAC mechanism enforced, respectively, for comparison purposes.

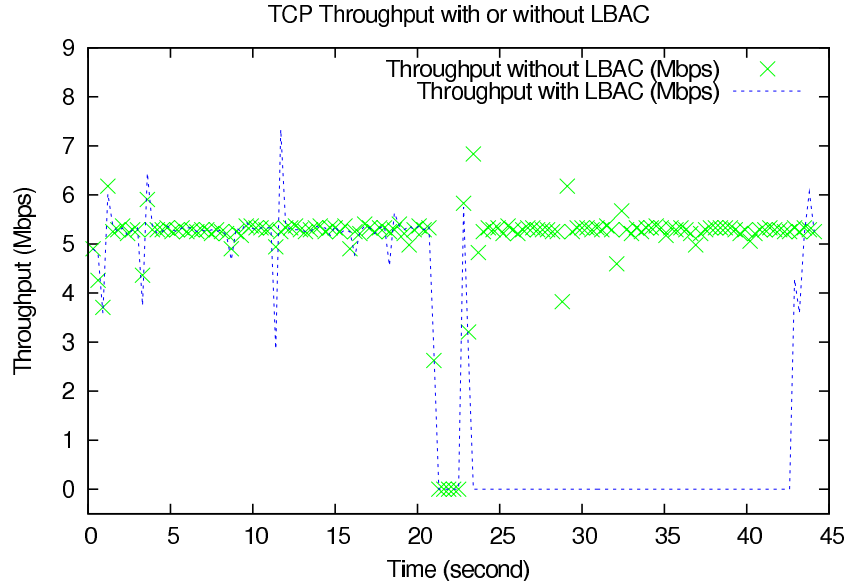


Figure 6: TCP Throughput with or without LBAC.

Figure 6 shows the throughput changes of the TCP connection between nodes 6 and 5 when the network operates with and without LBAC enforcement, respectively. As shown in Figure 6, both TCP streams with and without LBAC enforcement were disrupted at time 22.1 second because the MS lost the WiFi connection with the originally associated AP (node 1). However, the TCP stream without LBAC came back quickly to the normal throughput after associating with another AP (node 3), whereas the TCP stream with LBAC enforcement was not able to recover its connection until the MS returned to the access-granted area at time 35.9 second. Note that the TCP connection with LBAC had a longer time to recover its data stream because of TCP timing mechanisms.

Interestingly, there is a TCP throughput spike at 23 second for the simulation with LBAC enforcement in Figure 6. The reason was that the LBAC key server refreshes the public keys of the APs at 5 second intervals, and the MS still held the valid keys for the location group during 22-25 second. Therefore, the MS was able to continue communicating through the newly associated AP before the old keys expired. Once the new keys were released at time 25 second, the MS was no longer able to access the network.

Secondly, we simulated a CBR traffic between the MS and the fixed host in the same setup as shown in Figure 5. The CBR traffic was generated at 2 Mbps data rate. Figure 7 shows the throughput of the

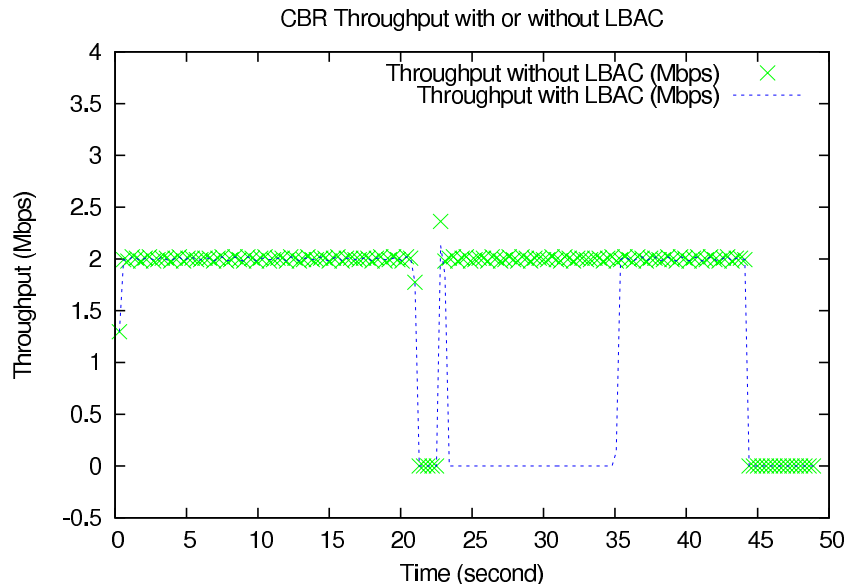


Figure 7: CBR Throughput with or without LBAC.

CBR connection with and without LBAC enforcement. We can see that the CBR came back up almost immediately to 2 Mbps once the MS got back to the access-granted area at time 35.9 second. The same throughput spike as the first simulation of TCP stream shows up at time 23 second because of the same reason.

4.3 Security Analysis

Man-in-the-middle attack is possible in Diffie-Hellman key exchange protocol, which was addressed extensively in the literature. Therefore, we analyze LBAC from different perspectives — the wormhole attack and the Sybil attack.

Wormhole attack The simplest wormhole attack is usually staged by two colluding attackers, one of which intercepts the traffic on one side of the network, and tunnels the packets to the other attacker for replaying on another side of the network. The wormhole attack is very difficult to detect, since it can be launched without compromising any host, or intruding the integrity and authenticity of the protocols [16, 13, 14].

In the LBAC system, wormhole attackers can forge legitimate location claims by gathering the location keys from different points of the network, even though none of the locations is authorized to access the network. For instance in Figure 8, two mobile stations MS_1 and MS_2 can collect the two location keys of a location group $G_1 = \{AP_1, AP_2\}$ by exchanging the missing keys of each other, and successful access the network through the corresponding AP visible to each MS.

Another possible scenario is that an MS quickly moves between the coverage areas of the APs in a location group, thereby gathering all the location keys for location authenticating purposes.

With regard to such attacks, we can ensure that each access-granted MS can in fact communication with all the APs of the location group by physically and forcefully switching the associated APs of an MS, as mentioned in Section 3.2.1. In addition, by periodically changing the location key of each AP, we can also

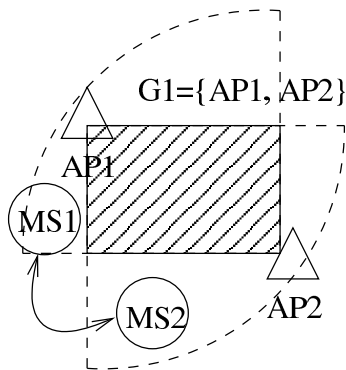


Figure 8: Wormhole Attack to LBAC.

ensure that the location keys are renewed in time before the collusive MSs catch up with the old location keys.

Sybil Attack In the Sybil attack [9, 24], an adversary impersonates multiple network entities by assuming their identities. Unlike the wormhole attack, the attacker is able to compromise communications, gain access to the cryptographic quantities, obtaining multiple node identities and injecting bogus data into the network [15].

In the LBAC system, Sybil attack can be staged by a malicious MS impersonating an AP (Rogue AP) of a location group by broadcasting bogus location keys as if it were the legitimate access point, henceforth preventing MSs to acquire legitimate keys and access the networks. Sybil attack to the MSs is usually prevented by MS authenticating the APs. For simplicity in LBAC, we have not provided such authentication mechanisms. However, LBAC can be easily extended to resist the Sybil attacks by requiring the APs to include a PKI certificate in the public key broadcast messages. Assuming that the mobile stations have sufficient computational and communication capacities, the public keys of a location group can be easily verified by the MSs.

5 Summary

We have described and evaluated LBAC, a secure Location-Based network Access Control based on the new location group and location key concepts. LBAC extensively uses the Diffie-Hellman algorithm, and does not depend on expensive location-detection hardware or infrastructure for location tracking and authentication purposes, therefore improving the protocol efficiency. Compared to previous systems, LBAC provides the minimum communication and computational overhead, thus is a promising technique for network access control based on coarse-grain location information.

References

- [1] IEEE Std 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Technical report, IEEE, Jul. 1997.
- [2] IEEE Std 802.11i/D3.0. Specification for Enhanced Security. Technical report, IEEE, Nov. 2002.

- [3] IEEE Std 802.1x. Port Based Network Access Control. Technical report, IEEE, Jun. 2001.
- [4] M. Balazinska, H. Balakrishnan, and D. Karger. INS/Twine: A Scalable Peer-to-Peer Architecture for Intentional Resource Discovery. In *Proc. of the First International Conference on Pervasive Computing*, pages 32–43, Aug. 2002.
- [5] P. S. L. M. Barreto, H. Kim, B. Bynn, and M. Scott. Efficient Algorithms for Pairing-Based Cryptosystems. *Advances in Cryptology - Crypto 2002*, Lecture Notes on Computer Science 2442:354–368, 2002.
- [6] D. Boneh and M. Franklin. Identify-based encryption from the weil-pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003.
- [7] S. Capkun, L. Buttyan, and J. Hubaux. SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks. In *Proc. of SASN 2003*, Virginia, Oct. 2003.
- [8] W. Diffie and M. Hellman. New Directions in Cryptography. *IEEE Transaction on Information Theory*, 22:644–654, Nov. 1976.
- [9] J. R. Douceur. The Sybil attack. In *First International Workshop on Peer-to-Peer Systems (IPTPS)*, Mar. 2002.
- [10] J. Edney and W.A. Arbaugh. *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*. Addison-Wesley, 2003.
- [11] U. Hengartner and P. Steenkiste. Implementing access control to people location information. In *Proceedings of 9th ACM Symposium on Access Control Models and Technologies (SACMAT 2004)*, pages 11–20, Yorktown Heights, NY, Jun. 2004.
- [12] J. Hightower and G. Borriello. Location Systems for Ubiquitous Computing. *IEEE Computer Magazine*, Aug. 2001.
- [13] Y. Hu, A. Perrig, and D. Johnson. Packet Leashes: A Defense Against Wormhole Attacks in Wireless Ad Hoc Networks. In *Proc. of INFOCOM*, San Francisco, CA, USA, Apr. 2003.
- [14] Y. Hu, A. Perrig, and D. Johnson. Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols. In *Proc. of ACM Workshop on Wireless Security (WISE 2003)*, Oct. 2003.
- [15] C. Karlof and D. Wagner. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. In *Proc. of IEEE International Workshop on Wireless Sensor Network Protocols and Applications*, 2004.
- [16] I. Khalil, S. Bagchi, and N. B. Shroff. LITEWORP: A Lightweight Countermeasure for the Wormhole Attack in Multihop Wireless Network. In *the International Conference on Dependable Systems and Networks (DSN)*, Yokohama, Japan, 2005.
- [17] J. Kohl and B. C. Neuman. RFC 1510 - The Kerberos Network Authentication Service (Version 5). Technical report, IETF, September 1993.
- [18] D. Kotz and K.H. Baek. A Survey of WPA and 802.11i RSN Authentication Protocols. Technical report, Dartmouth College Computer Science, 2004.

- [19] L. Lazos and R. Poovendran. SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks. In *2004 ACM workshop on Wireless security (ACM WiSe 2004)*, Philadelphia, PA, 2004.
- [20] N. Malhotra, M. Krasniewski, C. Yang, S. Bagchi, and W. Chappell. Location Estimation in Ad-Hoc Networks with Directional Antennas. In *25th International Conference on Distributed Computing Systems (ICDCS 2005)*, 2005.
- [21] R. C. Merkle. Secure Communication over an Insecure Channel. *Communication of ACM*, 21:294C99, Apr. 1978.
- [22] N. Michalakis. PAC: Location Aware Access Control for Pervasive Computing Environments. Technical report, MIT Laboratory of Computer Science, 200 Technology Square, Cambridge MA, 02139 USA, 2002.
- [23] R. M. Needham and M. D. Schroeder. Using Encryption for Authentication in Large Networks of Computers. In *Communication of the ACM*, pages 993–999, Dec. 1978.
- [24] J. Newsome, E. Shi, D. Song, and A. Perrig. The Sybil Attack in Sensor Networks: Analysis and Defenses. In *Proc. of IPSN 2004*, Berkeley, CA, Apr. 2004.
- [25] D. M. Pozar and D. H. Schaubert. *Microstrip Antennas: The Analysis and Design of Microstrip Antennas and Arrays*. Wiley-IEEE Press, May 1995.
- [26] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. In *6th ACM International Conference, Mobile Computing and Networking (MOBICOM)*, Aug. 2000.
- [27] VINT Project. The Network Simulator - ns-2. <http://www.isi.edu/nsnam/ns/>.
- [28] N. Sastry, U. Shankar, and D. Wagner. Secure Verification of Location Claims. In *Proc. of ACM Workshop on Wireless Security (WISE 2003)*, 2003.
- [29] RSA Security. RSA SecureID, Jun. 2003.
- [30] R.E. Smith. *Authentication: From Passwords to Public Keys*. Addison Wesley, 2002.
- [31] M. Steiner, G. Tsudik, and M. Waidner. Diffie-Hellman Key Distribution Extended to Group Communication. In *Proc. of the 3rd ACM conference on Computer and communications security*, pages 31–37, New Delhi, India, 1996.
- [32] A. S. Tanenbaum. *Modern Operating Systems, Second Edition*. Prentice Hall, 2001.
- [33] S.Y. Wang, C.L. Chou, Y.H. Chiu, Y.S. Tseng, M.S. Hsu, Y.W. Cheng, W.L. Liu, and T.W. Ho. NCTUns 4.0: An Integrated Simulation Platform for Vehicular Traffic, Communication, and Network Researches. In *1st IEEE International Symposium on Wireless Vehicular Communications*, Baltimore, MD, USA, Oct. 1 2007.
- [34] B. Waters and E. Felten. Proving the Location of Tamper Resistant Devices. Technical report, Princeton University, 2000.
- [35] Y. Zhang, W. Liu, W. Lou, and Y. Fang. Securing Sensor Networks with Location-Based Keys. In *IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, LA, Mar. 2005.