A TRUSTED INFRASTRUCTURE FOR FACILITATING ACCESS CONTROL OF LOCATION INFORMATION

Yingying Chen, Jie Yang, Fangming He
{yingying.chen, jyang, fhe}@stevens.edu
Dept. of Electrical and Computer Engineering
Stevens Institute of Technology, Hoboken, NJ 07030

ABSTRACT

The location information is critical for many location-aware applications and services. Recent efforts have resulted in a plethora of techniques to localize wireless devices. However, with the availability of the location information, privacy concerns are raised: by whom and when should the location information be accessed? In order to provide trusted access to the location information, in this work, we propose an infrastructure to regulate the access of the location information by enforcing that the mobile devices are only able to access the location information in a manner that conforms to their privileges. We presented both a centralized architecture as well as a fully decentralized enforcement mechanism. For the decentralized infrastructure, we proposed an on-node trusted component and developed the Neighbor ObseRvation Mechanism (NORM) for position verification of wireless devices. Further, the feasibility of our proposed infrastructure is evaluated through both simulation as well as a prototyping effort with a real-time indoor localization system.

I. INTRODUCTION

Technology trends will continue to reduce the size, power consumption and cost of embedded networked sensors, which makes the wireless sensor networks a part of our inseparable social fabric. And the concept of a wireless node has been extended to anything with a radio. Further, the rapid development in positioning technologies such as GPS, GSM, WiFi (802.11), and RFID have enabled a host of new applications based on the positions of nodes. However, extensive deployment of location-aware applications without safeguards may endanger location privacy of mobile users due to significant vulnerabilities for abuse. For example, location information can be used to derive the traces of users and further to learn about users' social behaviors, medical conditions, or different lifestyles. Moreover, attackers may launch location-based attacks such as spamming users with unwanted advertisements [1] or stalking to perform mental or physical harm.

Such vulnerabilities are significantly threatening the success deployment of the location-based services. Therefore, while the location information provides abundant opportunities for developing new applications, it could also be dangerous if misused by adversaries. Thus, one of the main challenges in wireless localization is sharing the right information with the right party at the right time. In particular, it is desirable to develop mechanisms, which provide safeguard and only allow the location information to be accessed by the right party at the right time.

Instead of using the access control methods provided in the application layer [2], [3], in this work, we propose a trusted infrastructure for regulating the access of location information at the physical layer so that the right information can be obtained as early as possible for building upper-layer location-based applications. The concepts of communal policies are introduced to enforce the proper access of the location information.

The location information can be stored in two ways: centralized and distributed. In a centralized architecture, a central server performs localization and stores the results in a centralized database, while in a decentralized approach, the localization is performed at each mobile device and the results are only available to the mobile device itself. To build a trusted infrastructure, we present both a centralized architecture as well as a fully decentralized enforcement mechanism.

Our centralized approach prevents adversaries that masquerade as another device [4] from accessing the location information by performing verification using a central database that stores not only the node ID but also the node position information. For the case when the location information is distributed in each mobile device, it is essential that each node has methods to ensure the appropriate access of its location information. We proposed an on-node trusted component and developed the Neighbor ObseRvation Mechanism (NORM), which performs position verification and helps to enhance the node

978-1-4244-2677-5/08/$25.00©2008 IEEE

verification for secure access. Comparing to prior position verification techniques [5]–[7], NORM does not require special hardware, deployment knowledge, or a central verification center. By using a two-step approach, NORM could perform position verification of a mobile device depending on the spatial consistency relationship inherited between a mobile device and its neighbors in a fully distributed way.

Further, the feasibility of our proposed infrastructure is evaluated through both system simulation as well as a prototyping effort when integrating with a real-time indoor localization system in a real office building environment for both a 802.11 network as well as a 802.15.4 network. Our validation from simulation shows tht $NORM$ outperforms the existing centralized position verification methods developed in [6] with higher detection rate (above 95%) and lower false positive rate (below 10%). Thus, our results provide strong evidence that our proposed infrastructure can serve as a trusted building block of obtaining location information for high-level applications.

The rest of the paper is organized as follows. We present an overview of our proposed trusted infrastructure in Section II. The on-node enforcement is discussed for the decentralized approach. We then describe NORM, the position verification mechanism in Section III. The authentication process and the policy formalism are presented in Section IV. Section V discusses our experimental evaluation. Finally, we conclude the paper and discuss further directions we are currently investigating in Section VI.

II. ARCHITECTURE OVERVIEW

In this section, we present an overview of our trusted infrastructure, which targets for any wireless networks. We first present a centralized architecture for location information access control. We then turn our focus to a decentralized policy enforcement approach. In our model, the location information access control involves two phases, *verification* and *authorization*. The verification phase performs authentication of the client, i.e., the mobile device requests the location information. The access of the location information will then be authorized based on communal policies. We will discuss the details of the *authorization* phase in Section IV.

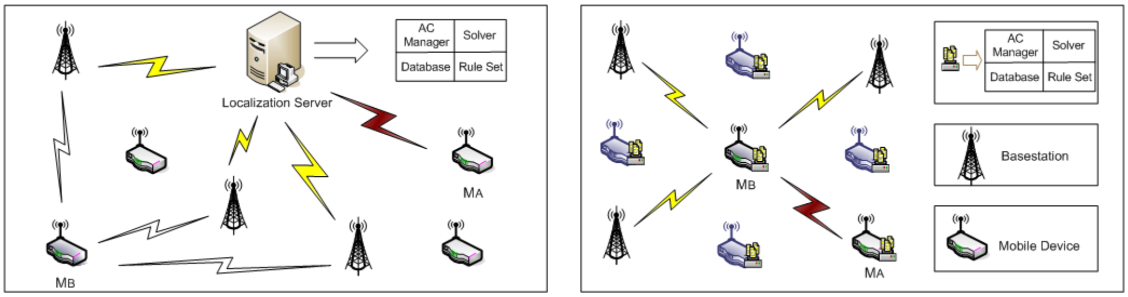
We note that the location information can be represented with multi-resolution spatial accuracy (e.g., point, room, or building) together with multi-period temporal accuracy (e.g., current or past). One major advantage of our proposed approach is that it can adjust the resolution of the returning location information to protect the location privacy. We use the terminology mobile devices to refer to any wireless devices that can be localized.

A. Centralized Architecture

In a centralized approach, the wireless localization is performed in a central server. The localization process is conducted continuously and the results are stored in the database as depicted in Figure 1 (a). In the area of interest, the base stations will report the signal readings of a mobile device back to a localization server. The localization server contains *solver* that has the data processing and analysis capabilities to estimate the positions of mobile devices. A management entity, namely the *Access Control Manager (AC Manager)*, performs verification and authorization before accessing the location information stored in the database. The *AC Manager* can reside within the localization server as shown in Figure 1 (a) or operate separately in a centralized manner but can access the database remotely. A set of access control rules will be disseminated and stored in the *AC Manager*.

As illustrated in Figure 1 (a) when a mobile device M_A wants to obtain the location information of another mobile device M_B, first it sends a request message to the *AC Manager* with its ID and current position. As evidenced by the numerous possible security threats due to node ID compromise or identity-based spoofing attacks [4], we note that it is not enough to verify a mobile device just based on its node ID. However, the position information is relatively harder to falsify without being detected. The advantage of the centralized architecture is that it can easily prevent identity-based attackers from accessing the location information by comparing to the complete position information stored in the central database. If a match is found, then the mobile device M_A is authenticated. Next, based on the verification status, the *AC Manager* consults with the access control policies stored in the $rule\ set$ and decides whether to send the exact location information as requested (e.g., the real coordinates of the position) or adjust the resolution of the location information (e.g., the room or floor level location resolution is returned.).

One drawback of the centralized approach utilizing *AC Manager* is that the server contains all the location information and inherently introduces an issue related to the user's privacy. Consequently, mobile devices may be tracked by the central server. Next,



(a) Centralized architecture

(b) Decentralized enforcement

Fig. 1. Architecture overview of the proposed trusted architecture.

we present our decentralized policy enforcement for location access control, which achieves user location privacy by not requiring interaction between a mobile device and a central server.

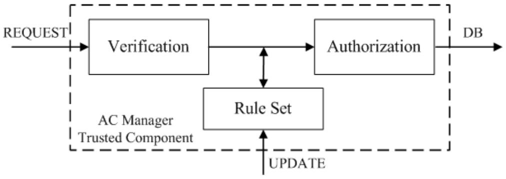


Fig. 2. Decentralized enforcement via AC Manager's trusted computing module.

B. Decentralized Enforcement

In order to build a decentralized infrastructure, mobile devices are equipped with the localization capability. Various localization methods can be applied in the *solver*. One simple approach is to use the multilateration strategy. When a mobile device collects signals from three or more base stations, it can position itself by applying the multilateration calculation [8]. The location information will then be stored in the database within the mobile device.

As shown in Figure 1 (b) the functionalities of location information access control will be distributed to each mobile device, which forms a decentralized trusted computing base. The access control policies will be disseminated to each mobile device and examined by the *AC Manager* that resides in each mobile device. Although we use the same name *AC Manager*, we replace the central entity of *AC Manager* with a distributed set of $ACManagers$. Structurely, all these $ACManagers$ are generic, support the same set of communal policies, and all must be trusted to interpret correctly any rules they might operate under.

On-Node Trusted Component: The access control policies need to be supported by enforcement mechanisms local to the mobile devices. It is therefore necessary to develop an on-node trusted computing base in each mobile device that enforces the policies. As depicted in Figure 2, the *AC Manager* implements the trusted component in each mobile device. It contains several logical components including $verification$, $authorization$, and $rule\ set$. Conceptually, the *AC Manager* can be viewed as a safeguard when the location request first coming in.

When a mobile device receives a request for location information, the *AC Manager* on the mobile device performs verification, which is the same as in the centralized approach. However, in the decentralized enforcement, the *AC Manager* does not have a central database that can be used to verify the client's ID and position. Instead of introducing the traditional cryptographic authentication methods on the mobile device [9], [10], we focus the node verification based on its location and propose a node location verification mechanism, Neighbor ObseRvation Mechanism ($NORM$), which utilizes observations from neighboring nodes, to enhance the identity-based authentication methods. The algorithmic details of $NORM$ is presented in Section III. Next, the *AC Manager* evaluates the request along with the verification results and checks it against the access control policies stored in the $rule\ set$. If the client's credentials don't permit the privilege level of the request, then the *AC Manager* will either try to find a permissible modification of the request that adapts to the access control policy and authorize the access of the granted location information, or reject the request if such a modification is not feasible. We will discuss the details of the adaptation functionality of the *AC Manager* in Section IV.

III. Neighbor ObseRvation Mechanism (NORM)

NORM is a decentralized location verification method for mobile devices that is used by the *AC Manager* to perform client node verification in our trusted decentralized enforcement. NORM consists of two steps: *neighbor examination* and *neighbor verification*.

Neighbor Examination Step: When the mobile device M_A requests the location information of the mobile device M_B, it sends its node ID, position P_A, and also its neighbor list, M_i, $i = 1, 2, ...N$ (with N is the total number of neighbors). Then the *AC Manager* of M_B issues a special verification request to the *AC Manager* of each device M_i in the neighbor list, and asks whether it has M_A in its neighbor list. Note that the special verification request is only communicated between the trusted *AC Managers* and will not be exposed to other components on the device. If the device M_i receives the request, its *AC Manager* confirms and reports its current position P_i back. We define the neighbor examination probability P_{ex} as

$$P_{ex} = \frac{\sum_{i=1}^{K} Mi}{N}. \tag{1}$$

where K is the total number of neighbors that responds to the request of the *AC Manager*. If $P_{ex} > \alpha$ where α is the confidence level, *AC Manager* determines that M_A passes the neighbor examination. A dishonest wireless device, who lies about its neighbor list, will thus result in $P_{ex} < \alpha$ and fails the neighbor examination. In reality the communication between two wireless devices is not symmetric. We found that useful values of α can have a wide dynamic range. Even smaller values of α can result in high detection rate in $NORM$, which is shown in Section V.

Neighbor Verification Step: A more advanced adversary may masquerade as another device [4] in order to access the location information with higher privileges, but send its correct neighbor list. Thus, passing the neighbor examination test is not enough to defeat a sophisticated adversary. Based on the reported positions of the responded neighbors, *AC Manager* could further conduct the neighbor verification. Given that the neighboring devices of M_A must be within the communication range R_A of M_A, the distance between the estimated locations of M_A and its neighbor $||P_A - P_i||$ should be within R_A for an honest wireless device. The *AC Manager* could complete the position verification of M_A if $||P_A - P_i|| < R_A$ for all $i = 1...K$.

However, since there are localization errors from the location estimation process [11], we define

$$||P_A - P_i|| < R_A + r, \tag{2}$$

where r is a random variable introduced by localization errors. We may assume localization errors are Gaussian [11]. Under this assumption, r also follows a Gaussian distribution with mean μ and variance σ. Thus the probability that P_A and P_i are neighbors is given by

$$Pr(P_i)=Pr(r>(||P_A-P_i||-R_A))=1-F(||P_A-P_i||-R_A), \tag{3}$$

with $F(r)$ as the Cumulative Distribution Function of r. Further, we define the neighbor verification probability P_{ve}, which is the joint probability that all P_i, $i = 1...K$, are the neighbor of P_A as:

$$P_{ve} = \prod_{i=1}^{K} Pr(P_i). \tag{4}$$

Moreover, we set a confidence level β (e.g., 75%) such that if $P_{ve} > \beta$, we declare that M_A passes the neighbor verification.

By using the two-step approach, NORM could perform position verification of a mobile device depending on the spatial consistency relationship inherited between a mobile device and its neighbors. Comparing to prior position verification techniques [5]–[7], NORM is a fully decentralized mechanism with no requirement of specialized hardware, deployment knowledge, or a central verification server, and is thus suitable for device position verification in our proposed decentralized trusted infrastructure.

IV. AUTHORIZATION AND POLICY FORMALISM

In our proposed trusted framework, wireless devices must adhere to the communal policies when requesting location information. In this section, we present the main functionality of the *AC Manager* to perform authorization, which evaluates the policy and authorizes the access of the location information. We also describe the policy formalism for accessing the location information.

The *AC Manager* implements three main functionalities, namely *Matching*, *Adaptation*, and *Application*. First, *AC Manager* matches the location request to the rule set. A location request may satisfy one or more policies in the rule set. The *AC Manager* will then return the information based on the matching that provides the finest granularity of the location information that is permitted according to the client's credentials.

However, if the client's credentials don't permit the privilege level of the location request. For example, if M_A requests for point-level location information of M_B, but its credentials only allows it to access the room-level location information of M_B based on current policies. The *AC Manager* modifies the request to adapt to the access control policy and authorize M_A to access the room-level location information of M_B. On the other hand, the available location information may be in a finer granularity than the location request. One example is that M_A only needs to know at which floor that M_B is located. The *AC Manager* could then reduce the spatial accuracy, protect the location privacy, and meet the requirement of the location request. Finally, if an adaptation is not feasible, the location request is rejected at the authorization phase.

Moreover, the application function is used to impose the usage of the location information returned to the client. Two important aspects are: *retransmission* and *retention*. The application of *retransmission* defines whether the client is permitted to share the obtained location information with other mobile devices. *retransmission* aims to prevent unauthorized usage of the location information. Whereas the application of *retention* defines the duration that the returned location information is valid. Further, in order to prevent frequent location requests from the same client, which may be used to derive the moving track of a mobile device, the *AC Manager* keeps a list of the clients and records their request time as part of the application function.

Turning to examine the policy formalism, the mobile devices should be able to interpret the policies and update them as needed. Hence, policies should be expressed in an easy to understand manner and can facilitate rule integration, consistency checking and conflict resolution. The following are two examples of rules to access the location information specified in plain English. The pseudo code implementation of $R1$ is presented in Figure 3.

- *Rule 1:* (1) allow access to both the current as well as the past 30 minutes location information, (2) the location accuracy is at room-level, (3) the location information is forbidden to be shared with other devices once obtained by a client device, (4) the location information is valid for 60 minutes, and (5) the access frequency of the location information is 30 minutes.
- *Rule 2:* (1) allow access to the current location trace and the duration of the trace is 1 hour, (2) the location accuracy is at point-level, (3) the location information is allowed to be shared with other devices once obtained by a client device, (4) the location information is valid forever, and (5) the access frequency of the location information is 2 hours.

```
Bool Matching () {
    if (Request(location) == ROOM)
        return TRUE;
}
Permission () {
    Multi_Time () {FALSE};
    One_Time () {TRUE};
}
One_Time () {
    current_location = TRUE;
    past_location = TRUE;
    past_duration = 30 MIN;

    location_resolution = ROOM_LEVEL;
}
Bool Adaptation () {
    if (Matching ()) {
        // if the Request(request) is a subset
        // of the Permission, then no need to
        // perform adaptation.
        if (Request(request) <= Permission)
                Authorization = Request(request);
        else    Authorization = Modified_request();
        return TRUE;
    }
}
Modified_Request() {
    //Adapting the location resolution from
    //POINT_LEVEL to ROOM_LEVEL
    Request(request(resolution)) = ROOM_LEVEL;
}
Bool Application () {
    if (Adaptation()) {
        do Authorization;
        do Usage ();
        frequency = 30 MIN;
    }
}
Usage () {
    retransmission = FORBIDDEN;
    retention = 60 MIN;
}
```

Fig. 3. Pseudo code implementation of *Rule 1*.

Figure 3 illustrates how the *AC Manager* performs authentication in terms of matching, adaptation, and application. To enforce *Rule 1*, $Matching()$ is used to apply the room-level resolution, and $Adaptation()$ adapts the location information from the point level to the room level. Finally, $Application()$ enforces the usage of the location information with *retransmission* and *retention* setting to $FORBIDDEN$ and 60 minutes respectively.

V. EXPERIMENTAL EVALUATION

In this section, we evaluate the effectiveness of $NORM$ in the decentralized enforcement and describe our prototype in building the trusted infrastructure.

To evaluate the feasibility of our centralized architecture for policy enforcement, we integrated the $ACManager$ into a real-time indoor localization system [12]. The system components for the prototype is shown in Figure 4. During the localization process, a mobile device sends packets. Some number of Landmarks (i.e., traffic observers or base stations) observe the packets and record the RSS (Received Signal Strength) readings. Each landmark forwards the observed RSS from the mobile device to the

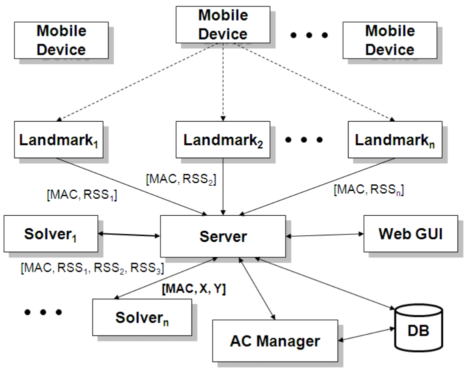


Fig. 4. Prototype: system components in the centralized approach.

Server. The Server collects the complete RSS vector for the mobile device and sends the information to a Solver instance for location estimation. The Solver instance performs the localization and returns the location estimate of the wireless device back to the Server. The Server stores the location estimate to the database and displays it in GUI.

When a mobile device sends a request to the Server to access the location information of another mobile device, the $ACManager$ performs *verification* and *authentication* before granting the access to the location information as described in Section II. Once the $ACManager$ grants the access, the requested location information is fetched from the database and sent back to the client device. We prototyped the centralized approach in both a 802.11 (WiFi) network as well as a 802.15.4 (ZigBee) network in a real office building environment as depicted in Figure 5 where the stars are the deployed landmarks.

On the other hand, in the decentralized enforcement, we evaluated the effectiveness of $NORM$. We conducted simulation of both sparse as well as dense wireless networks by deploying 250 sensors and 500 sensors randomly in a $250m$ X $250m$ square field respectively. Since in reality the communication between two wireless devices is not symmetric, we set sensors' communication range follow a Gaussian distribution with mean at $R = 20m$ and standard deviation as $3m$. There are average 4 neighbors per node in our sparse network and 9 neighbors per node in our dense network. We then conducted various experiments based on our simulation setup.

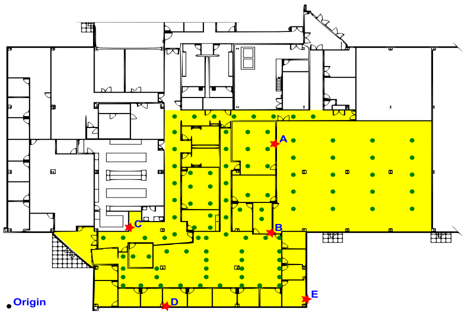
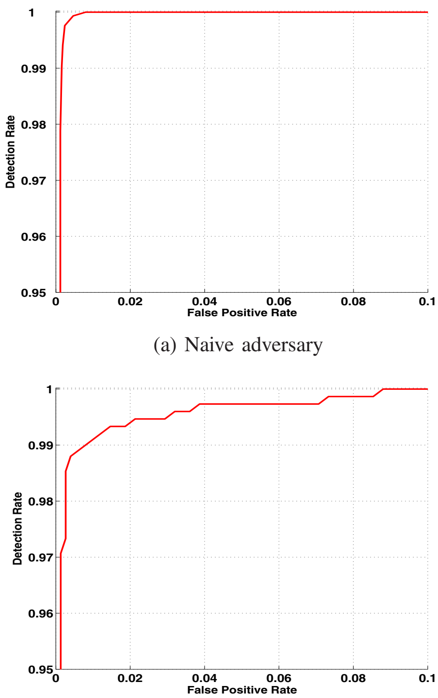


Fig. 5. Prototype: experimental floor plan.



(a) Naive adversary

(b) Sophisticated adversary

Fig. 6. Receiving Operating Characteristic (ROC) curve using NORM in a dense network with about 9 neighbors per node.

Table I presents the detection rate and false positive rate when using $NORM$ for the naive adversary case and the sophisticated adversary case. We found that NORM can achieve high detection rate (above 90%) and low false positive rate (below 10%) even under small confidence levels. We observed that the detection rate in the dense network outperformed the detection rate in the sparse network. Since $NORM$ is based on the observation of neighbor nodes, the more neighbors in the dense network help to improve the detection rate and reduce the false positive rate.

Further, Figure 6 shows the ROC curves in a dense network with about 9 neighbors per node for the naive adversary case and the sophisticated adversary case respectively. We observed that for false positive rates less than 10%, the detection rates are above 95% and reaching 100% when the false positive rate increases to 1% for the naive adversary case and 9% for the sophisticated adversary case.

The work of position verification methods that is most closely related to ours is [6], which needs a centralized verification server. We thus compare the performance of $NORM$ to the algorithms proposed in [6]. In addition to its fully distributed nature, we found that $NORM$ outperformed the TI and GFM algorithms developed by [6] in both the high density

Confidence	Detection Rate	False Positive Rate
Naive Adversary Model, Number of neighbors: 9		
α = 0.13	0.982	0.002
α = 0.30	0.999	0.005
α = 0.40	1	0.008
Naive Adversary Model, Number of neighbors: 4		
α = 0.18	0.971	0.029
α = 0.33	0.997	0.043
α = 0.47	1	0.072
Sophisticated Adversary Model, Number of neighbors: 9		
β = 0.25	0.973	0.003
β = 0.68	0.997	0.039
β = 0.76	1	0.088
Sophisticated Adversary Model, Number of neighbors: 4		
β = 0.35	0.916	0.028
β = 0.61	0.948	0.060
β = 0.72	0.952	0.104

TABLE I

EFFECTIVENESS OF NEIGHBOR OBSERVATION MECHANISM (NORM).

network as well as the sparse network. Specifically, in the high density network, the detection rate of $NORM$ can achieve 100% when false positive rate is less than 10%, whereas the detection rates are 88% for TI and 82% for GFM respectively. In the sparse network, the detection rate of $NORM$ also reaches 100% when the false positive rate is 10%, however, the detection rates are only 85% and 69% for TI and GFM algorithms respectively. Thus, our simulation results provide strong evidence that NORM is effective in providing location verification in a decentralized manner.

Finally, the decentralized enforcement is prototyped in a 802.15.4 network using Tmote Sky motes. The $ACManager$ and the rule set are implemented at each mote together with NORM for position verification. Our results show that it is feasible to build a trusted infrastructure to regulate the access to the location information. Future work is to quantify the overhead in terms of the performance and the memory that the decentralized enforcement approach has introduced to wireless devices.

VI. CONCLUSION

In this work, we proposed a trusted infrastructure for facilitating the access control of the location information by enforcing that the mobile devices are only able to access the location information in a manner that conforms to their privileges. We presented both a centralized architecture as well as a fully decentralized enforcement mechanism. For the decentralized infrastructure, we proposed an on-node trusted component and developed Neighbor ObseRvation Mechanism ($NORM$) for position verification. We applied the concept of communal policies to access the location information and discussed the policy formalism. Further, our simulation results show that $NORM$ is highly effective to perform position verification for both the naive adversary case and the sophisticated adversary case. By integrating with a real-time indoor localization system, we prototyped our trusted infrastructure in both a 802.11 network as well as a 802.15.4 network in a real office building environment. Our results provide strong evidence of the feasibility of the trusted infrastructure for location information access control.

Our future work in this direction falls into two areas: (1) develop the context-based access control of the location information and (2) build hierarchical policies to interoperate across different wireless networks.

REFERENCES

[1] B. Schilit, J. Hong, and M. Gruteser, “Wireless location privacy protection,” *IEEE Computer 36 (12)*, December 2003.

[2] E. Snekkenes, “Concepts for personal location privacy policies,” in *Proceedings of the ACM Conference on Electronic Commerce (EC)*, Oct. 2001.

[3] H. Tschofenig, H. Schulzrinne, A. Newton, J. Peterson, and A. Mankin, “The IETF geopriv and presence architecture focusing on location privacy,” in *Proceedings of the W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven enforcement*, Oct. 2006.

[4] Y. Chen, W. Trappe, and R. P. Martin, “Detecting and localizing wirelss spoofing attacks,” in *Proceedings of the Fourth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, May 2007.

[5] S. Capkun and J. P. Hubaux, “Secure positioning of wireless devices with application to sensor networks,” in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2005, pp. 1917–1928.

[6] Y. Wei, Z. Yu, and Y. Guan, “Location verification algorithms for wireless sensor networks,” in *Proceedings of the 27th IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2007.

[7] W. Du, L. Fang, and P. Ning, “Lad: Localization anomaly detection for wireless sensor networks,” in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS 05)*, April 2005.

[8] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, “Locating the nodes,” *IEEE Signal Processing Magazine*, July 2005.

[9] B. Wu, J. Wu, E. Fernandez, and S. Magliveras, “Secure and efficient key management in mobile ad hoc networks,” in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2005.

[10] T. Aura, “Cryptographically generated addresses (cga),” *RFC 3972, IETF*, 2005.

[11] A. Krishnakumar and P. Krishnan, “On the accuracy of signal strength-based location estimation techniques,” in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, March 2005.

[12] Y. Chen, G. Chandrasekaran, E. Elnahrawy, J.-A. Francisco, K. Kleisouris, X. Li, R. P. Martin, R. S. Moore, and B. Turgut, “Grail: General real time adaptable indoor localization,” *Sensor Review*, 2008.