LINK: Location verification through Immediate Neighbors Knowledge

Manoop Talasila, Reza Curtmola, and Cristian Borcea

Computer Science Department
New Jersey Institute of Technology
Newark, NJ, USA
{mt57,crix}@njit.edu,borcea@cs.njit.edu

Abstract. In many location-based services, the user location is determined on the mobile device and then shared with the service. For this type of interaction, a major problem is how to prevent service abuse by malicious users who lie about their location. This paper proposes LINK (Location verification through Immediate Neighbors Knowledge), a location authentication protocol in which users help verify each other's location claims. This protocol is independent of the wireless network carrier, and thus works for any third-party service. For each user's location claim, a centralized Location Certification Authority (LCA) receives a number of verification messages from neighbors contacted by the claimer using short-range wireless networking such as Bluetooth. The LCA decides whether the claim is authentic or not based on spatio-temporal correlation between the users, trust scores associated with each user, and historical trends of the trust scores. LINK thwarts attacks from individual malicious claimers or malicious verifiers. Over time, it also detects attacks involving groups of colluding users.

Key words: secure location authentication, trust, smart phones

1 Introduction

Recently, location-based services have started to be decoupled from the wireless network carriers, as illustrated by third-party services such as Loopt, Brightkite, and Google's Latitude. As such, the service providers must rely on the mobile devices to provide their location using GPS or other localization mechanisms. A major problem in this case is how to prevent service abuse by malicious users who tamper with the localization system on the mobile devices. For example, how can a store verify that only users in a 2-mile radius receive coupons? How can a cab company verify the location of a person who requested a cab? How can a news agency authenticate the claimed location of a geo-tagged photo uploaded by citizens located at an event of public interest?

Although a significant number of publications tackled the location authentication problem, all of them assumed support from the network infrastructure [1, 2] or from a deployed localization infrastructure using distance-bounding techniques [3, 4]. Typically, these solutions are based on signal measurements be-

tween the mobile devices and fixed beacons or base stations (e.g., cell towers, WiFi access points) with known locations [5]. The problem tackled in this paper is different as we aim for a solution that works without any support from the network/localization infrastructure. Such a solution is important because wireless carriers may refuse to authenticate user location for third-party services due to legal and commercial reasons: they may not be allowed by laws to share any type of user location data, and and they may not want to help their competition in the location-based services area.

This paper proposes LINK (Location verification through Immediate Neighbors Knowledge), a secure location authentication protocol in which users help verify each other's location claims. LINK associates trust scores to users, and mobile neighbors with high trust scores play similar roles with the trusted beacons/base stations in existing solutions. The main idea is to leverage the neighborhood information available through short-range wireless technologies, such as Bluetooth which is available on most cell phones, to verify if a user is in the vicinity of other users with high trust scores.

LINK employs a Location Certification Authority (LCA) that interacts with the location-based services and with the mobile users over the Internet. Before submitting a location authentication request to the LCA, the claimers must broadcast a message to their neighbors using short-range wireless ad hoc communication. In response to this message, the neighbors send verification messages to the LCA over the Internet. The LCA decides the claim's authenticity based on spatio-temporal correlation between users and the trust score associated with each user. The protocol leverages the centralized nature of the LCA to compute the trust scores based on past interactions and historical score trends. While it works best in dense networks that provide enough neighbors, LINK was designed to be resilient to situations when users are alone.

Extensive simulation results and security analysis show that LINK can thwart attacks from individual malicious claimers or malicious verifiers. Over time, it can also detect more complex attacks involving groups of colluding users.

The rest of the paper is organized as follows. Section 2 defines the assumptions and the adversarial model. Section 3 describes the LINK protocol, and Section 4 analyzes its security. Section 5 presents the simulation results. The related work is discussed in Section 6, and the paper concludes in Section 7.

2 Preliminaries

This section defines the interacting entities in our environment, the assumptions we make about the system, and the adversarial model.

Interacting entities. The entities in the system are:

- *Claimer*: The mobile user who claims a certain location and subsequently has to prove the claim's authenticity.
- *Verifier*: A mobile user in the vicinity of the claimer (as defined by the transmission range of the wireless interface, which is Bluetooth in our implementation). This user receives a request from the claimer to certify the claimer's location and does so by sending a message to the LCA.

– *Location Certification Authority (LCA)*: A service provided in the Internet that can be contacted by location-based services to authenticate claimers' location. All mobile users who need to authenticate their location are registered with the LCA.
– *Location-based Service (LBS)*: The service that receives the location information from mobile users and provides responses as a function of this location.

System and Adversarial Model. We assume that each mobile device has means to determine its location. This location is considered to be approximate, within typical GPS or other localization systems limits. We assume the LCA is trusted and the communication between mobile users and LCA is secure. We also assume that each user has a pair of public/private keys and a digital certificate from a PKI. Similarly, we assume the LCA can retrieve and verify the certificate of any user. All communication happens over the Internet, except the short-range communication between claimers and verifiers.

We choose Bluetooth for short-range communication in LINK because of its pervasiveness in cell phones and its short transmission range (10m) which provides good accuracy for location verification. However, LINK can leverage WiFi during its initial deployment in order to increase the network density. This solution trades off location accuracy for number of verifiers.

LCA can be a bottleneck and single point of failure in the system. Currently, we do not address this issue, but standard distributed systems techniques can be used to improve the LCA's scalability and fault-tolerance. For example, an individual LCA server/cluster can be assigned to handle a specific geographic region, thus reducing the communication overhead significantly (i.e., communication between LCA servers is only required to access user's data when she travels away from the home region). Additionally, the geographic distribution of servers can improve response latency.

Any claimer or verifier may be malicious. When acting individually, malicious claimers may lie about their location. Malicious verifiers may refuse to cooperate when asked to certify the location of a claimer and may also lie about their own location in order to slander a legitimate claimer. Additionally, malicious users may perform stronger attacks by colluding with each other. A group of colluding malicious users may try to verify each other's false claims (we assume the attackers are able to communicate with each other using out-of-band channels).

We do not consider selfish attacks, in which users seek to reap the benefits of participating in the system without having to expend their own resources (e.g., battery). These attacks can be solved by leveraging the centralized nature of LCA, which can enforce a tit-for-tat mechanism (similar to those found in P2P protocols such as BitTorrent). For example, a user can be informed that she needs to perform a number of verifications for each submitted claim. Finally, we rely on that obtaining digital certificates is not cheap; this deters Sybil attacks [6].

3 Protocol Design

This section presents the basic LINK operation, describes the strategies used by LCA to decide whether to accept or reject a claim, and then details how trust

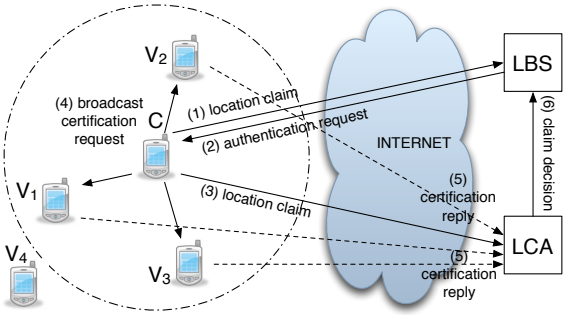

Fig. 1. Basic Protocol Operation (where C = claimer, V_i = verifiers, LBS = Location-Based Service, LCA = Location Certification Authority).

scores and verification history are used to detect strong attacks from malicious users who change their behavior over time or collude with each other.

3.1 Basic Protocol Operation

All mobile users who want to use LINK must register with the LCA. During registration, the LCA generates a userID based on the user's digital certificate. At the same time, the LCA assigns an initial trust score for the user (which can be set to a default value or assigned based on other criteria). Trust scores are maintained and used by the LCA to decide the validity of location claims. A user's trust score is additively increased when her claim is successfully authenticated and multiplicatively decreased otherwise in order to discourage malicious behavior. This policy of updating the scores is demonstrated to work well for the studied attacks, as shown in section 5. The values of all trust score increments, decrements, and thresholds are presented in the same section. A similar trust score updating policy has been shown to be effective in P2P networks as well [7].

Figure 1 illustrates the basic LINK operation. In step 1, a user (the claimer) wants to use the LBS and submits her location. The LBS then asks the claimer to authenticate her location (step 2). In response, the claimer will send a signed message to LCA (step 3) containing *(userID, location, seq-no, serviceID)*. The sequence number is used to protect against replay attacks (to be discussed in Section 4). The LCA timestamps and stores each newly received claim.

The claimer then starts the verification process by broadcasting to its neighbors a location certification request over the short-range wireless interface (step 4). This message is signed and contains *(userID, seq-no)*, with the same sequence number as the claim in step 3. The neighbors who receive the message, acting as verifiers for the claimer, will send a signed certification reply message to LCA (step 5). This message includes *(userID, location, certify-request)*, where the userID and location are those of the verifier and certify-request is the certification request broadcasted by the claimer. The certification request is included to allow the LCA to match the claim and its certification messages. Additionally, it proves that indeed the certification reply is in response to the claimer's request.

The LCA waits for the certification reply messages for a short period of time and then starts the decision process (described next in section 3.2). Finally, the

LCA informs the LBS about its decision (step 6), causing the LBS to provide or deny service to the claimer.

3.2 LCA Decision Process

In the following, we describe the LCA decision process. For the sake of clarity, this description skips most of the details regarding the use of historical data when making decisions, which are presented in Section 3.3.

Spatio-temporal correlation. The LCA checks the claimed location of the claimer with respect to the claimer's previously recorded claim. If it is not physically possible to move between these locations in the time period between the two claims, the new claim is rejected.

Contradictory verifications. If the claimer's location satisfies the spatio-temporal correlation, the LCA selects only the "good" verifiers who responded to the certification request. These verifiers must have trust scores above a certain threshold. We only use "good" verifiers because verifiers with low scores may be malicious. Nevertheless, the low score verifiers respond to certification requests in order to be allowed to submit their own certification claims (i.e., tit-for-tat mechanism) and, thus, potentially improve their trust scores.

After selecting the "good" verifiers, the LCA checks if they are colluding with the claimer to provide false verifications, and it rejects the claim if that is the case. This *collusion check* is described in detail in the next section. If the claimer and verifiers are not colluding, the LCA accepts or rejects the claim based on the difference between the sums of the trust scores of the two sets of verifiers, those who agree with the claimer and those who do not.

Low difference between the two sets of verifiers. If the difference between the trust score sums of two sets of verifiers is low, the LCA does not make a decision yet. It continues by checking the trust score trend of the claimer: if this trend is poor, with a pattern of frequent score increases and decreases, the claimer is deemed malicious and the request rejected. Otherwise, the LCA checks the score trends and potentially the location of the verifiers who disagree with the claimer. If these verifiers are deemed malicious, the claim is accepted. Otherwise, the claim is ignored, which forces the claimer to try another authentication later.

No verifiers. Finally, the LCA deals with the case when no "good" verifiers are found to certify the claim (this includes no verifiers at all). If the claimer's trust score trend is good and her trust score is higher than a certain threshold, the claim is accepted. In this situation, the claimer's trust score is decreased by a small value to protect against malicious claimers who do not broadcast a certification request to their neighbors when they make a claim. Over time, a user must submit claims that are verified by other users; otherwise, all her claims will be rejected.

3.3 Use of Historical Data in LCA Decision

The LCA maintains for each user the following historical data: (1) all values of the user's trust score collected over time, and (2) a list of all users who provided verifications for this user together with a verification count for each of

them. These data are used to detect and prevent attacks from malicious users who change their behavior over time or who collude with each other.

Trust score trend verification. The goal of this verification is to analyze the historical trust values for a user and find malicious patterns. This happens typically when there are no good verifiers around a claimer or when the verifiers contradict each other with no clear majority saying to accept or reject the claim.

For example, a malicious user can submit a number of truthful claims to improve her trust score and then submit a malicious claim without broadcasting a certification request to her neighbors. Practically, the user claims to have no neighbors. This type of attack is impossible to detect without verifying the historical trust scores. To prevent such an attack, the LCA counts how many times has a user's trust score been decreased over time. If this number is larger than a certain percentage of the total number of claims issued by that user, the trend is considered malicious.

Colluding users verification. Groups of users may use out-of-band communication to coordinate attacks: For example, they can send location certifying messages to LCA on behalf of each other with agreed-upon locations. To mitigate such attacks, the LCA maintains an NxN matrix M that tracks users certifying each other's claims (N is the total number of users in the system). M[i][c] counts how many times user i has acted as verifier for user c.

For each claim, the LCA uses weighted trust scores for verifiers. The weighted trust score of a verifier v is $W_v = T_v/\log_2(M[i][c])$, where T_v is the actual trust score of v. The more a user certifies another user's claims, the less its certifying information will contribute in the LCA decision. We choose a log function to induce a slower decrease of the trust score as the count increases. Nevertheless, a small group of colluding users can quickly end up with all their weighted scores falling below the threshold for "good" users, thus stopping the attack.

If the group of colluding users is larger, the weighted scores will be above this threshold for a longer time, improving the attack's effectiveness. To protect against this attack, LINK rejects a claim if the following conditions are satisfied for the claimer: (1) the number of claims verified by each potentially colluding user is greater than a significant fraction of the total number of claims issued by the claimer, and (2) the number of potentially colluding users who satisfy the first condition is greater than a significant fraction of the total number of verifiers for the claimer.

Eventually, repeated verifications from the same group of colluding verifiers will be ignored. However, it is possible that repeated verifications are from legitimate verifiers (e.g., close family or a few colleagues at work). If the number of repeated verifiers is small compared to the total number of verifiers for a given claimer, LINK will reset the weights of these verifiers to allow them to have a greater contribution in future verifications for the claimer. The detailed algorithm is presented in the companion technical report [8].

4 Security Analysis

The decision made by the LCA to accept or reject a claim relies on the trust scores of the users involved in this claim (i.e., claimer and verifiers). Thus, from a security perspective, the protocol's goal is to ensure that *over time* the trust score of malicious users will decrease, whereas the score of legitimate users will increase. LINK uses an additive increase and multiplicative decrease scheme to manage trust scores in order to discourage malicious behavior.

There are certain limits to the amount of adversarial presence that LINK can tolerate. For example, LINK cannot deal with an arbitrarily large number of malicious colluding verifiers supporting a malicious claimer because it becomes very difficult to identify the set of colluding users. Similarly, LINK cannot protect against users who accumulate high scores and very rarely issue false claims while pretending to have no neighbors (i.e., the user does not broadcast a certification request). An example of such situation is a "hit and run" attack, when the user does not return to the system after issuing a false claim. Thus, we do not focus on preventing such attacks. Instead, we focus on preventing users that *systematically* exhibit malicious behavior. Up to a certain amount of adversarial presence, our experimental evaluation in Section 5 shows that the protocol is able to decrease over time the scores of users that exhibit malicious behavior consistently and to increase the scores of legitimate users.

All certification requests and replies are digitally signed, thus the attacker cannot forge them, nor can she deny messages signed under her private key. Attackers may attempt simple attacks such as causing the LCA to use the wrong certification replies to verify a location claim. LINK prevents this attack by requiring verifiers to embed the certification request in the certification reply sent to the LCA. This also prevents attackers from arbitrarily creating certification replies that do not correspond to any certification request, as they will be discarded by the LCA.

Another class of attacks claims a location too far from the previously claimed location. In LINK, the LCA prevents these attacks by detecting it is not feasible to travel such a large distance in the amount of time between the claims.

Attackers may try to slander other nodes by intercepting their certification requests and then replaying them at a later time in a different location. However, the LCA is able to detect that it has already processed a certification request (extracted from a certification reply) because each such request contains a sequence number and the LCA maintains a record of the latest sequence number for each user.

We now consider individual malicious claimers that claim a false location. If the claimer follows the protocol and broadcasts the certification request, the LCA will reject the claim because the claimer's neighbors provide the correct location and prevail over the claimer. However, the claimer may choose not to broadcast the certification request and only contact the LCA. If the attacker has a good trust score, she will get away with a few false claims. The impact of this attack is limited because the attacker trust score is decreased by a small

decrement for each such claim, and she will soon end up with a low trust score; consequently, all future claims without verifiers will be rejected.

An individual malicious verifier may slander a legitimate user who claims a correct location. However, in general, the legitimate user has a higher trust score than the malicious user. Moreover, the other (if any) neighbors of the legitimate user will support the claim. The LCA will thus accept the claim.

A group of colluding attackers may try to verify each other's false locations using out-of-band channels to coordinate with each other. LINK deals with this attack by recording the history of verifiers for each claimer and gradually decreasing the contribution of verifiers that repeatedly certify for the same claimer (see Section 3.3). Even if this attack may be successful initially, repeated certifications from the same group of colluding verifiers will eventually be ignored.

Limitations and future work. The thresholds in the protocol are set based on our expectations of normal user behavior. However, they can be modified or even adapted dynamically in the future.

LINK was designed under the assumption that users are not alone very often when sending the location authentication requests. As such, it can lead to significant false positive rates for this type of scenario. Thus, LINK is best applicable to environments in which user density is relatively high.

A potential attack is when a group of colluding verifiers may try to slander a legitimate claimer. As long as at least one malicious verifier is near the legitimate claimer, it can use out-of-band communication to forward the claimer's certification requests and coordinate with the other malicious verifiers to slander the claimer. However, in order to target a specific claimer, the attackers would need to have a physical presence near the claimer. Since it is unlikely that the attackers would have a physical presence near an arbitrarily chosen claimer, we do not consider this attack in the paper.

We implicitly assume that all mobile devices have the same nominal wireless transmission range. One can imagine ways to break this assumption, such as using non-standard wireless interfaces that can listen or transmit at higher distances such as the BlueSniper rifle from DEFCON '04. In this way, a claimer may be able to convince verifiers that she is indeed nearby, while being significantly farther away. Such attacks can be prevented by using a "traditional" secure localization protocol that bounds the distance between a prover and a verifier based on the signal's time of flight [9].

Location privacy could be an issue for verifiers. Potential solutions may include rate limitations (e.g., number of verifications per hour or day), place limitations (e.g., do not participate in verifications in certain places), or even turning LINK off when not needed for claims. However, the tit-for-tat mechanism requires the verifiers to submit verifications in order to be allowed to submit claims. To protect verifier privacy against other mobile users in proximity, the verification messages could be encrypted as well.

Table 1. Simulation setup for the LINK protocol

Parameter	**Value**
Simulation area	100m x 120m
Number of nodes	200
% of malicious users	1, 2, 5, 10, 15
Colluding user group size	4, 6, 8, 10, 12
Bluetooth transmission range	10m
Simulation time	300min
User walking speed	1m/sec
Claim generation rate (uniform)	1/min, 1/2min, 1/4min, 1/8min
Trust score range	0.0 to 1.0
Initial user trust score	0.5
"Good" user trust score threshold	0.3
Low trust score difference threshold	0.2
Trust score increment	0.1
Trust score decrement - common case	0.5
Trust score decrement - no neighbors	0.1

5 Performance Analysis

This section presents the evaluation of LINK using the ns-2 simulator. The two main goals of the evaluation are: (1) Measuring the false negative rate (i.e., percentage of accepted malicious claims) and false positive rate (i.e., percentage of denied truthful claims) under various scenarios, and (2) Verifying whether LINK's performance improves over time as expected.

5.1 Simulation Setup

The simulation setup parameters are presented in Table 1. The average number of neighbors per user considering these parameters is slightly higher than 5. Since we are interested to measure LINK's security performance, not its network overhead, we made the following simplifying changes in the simulations. Bluetooth is emulated by WiFi with a transmission range of 10m. This results in faster transmissions as it does not account for Bluetooth discovery and Piconet formation. However, the impact on security is minimal due to the low, walking speeds considered in these experiments. The second simplification is that the communication between the LCA and the users does not have any delay; the same applies for the out-of band communication between colluding users. Finally, a few packets can be lost due to wireless contention because we did not employ reliable communication in our simulation. However, given the low claim rate, their impact is minimal.

5.2 Simulation Results

Always malicious individual claimers. In this set of experiments, a certain number of non-colluding malicious users sends only malicious claims; however, they verify correctly for other claims.

If malicious claimers broadcast certifying requests, the false negative rate is always observed to be 0. These claimers are punished and, because of low trust

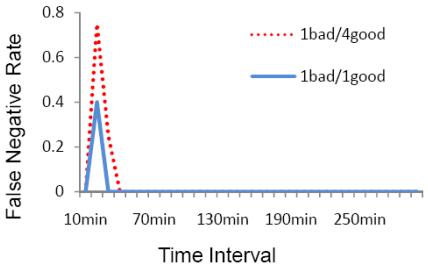

Fig. 2. False negative rate over time for individual malicious claimers with mixed behavior. The claim generation rate is 1 per minute, 15% of the users are malicious, and average speed is 1m/s.

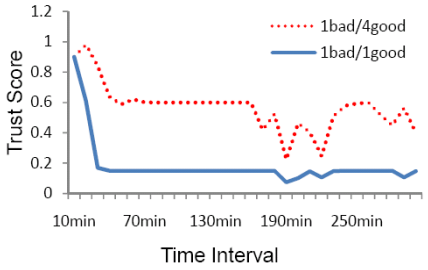

Fig. 3. Trust score of malicious users with mixed behavior over time. The claim generation rate is 1 per minute, 15% of the users are malicious, and average speed is 1m/s.

scores, they will not participate in future verifications. For higher numbers of malicious claimers, the observed false positive rate is very low (under 0.1%), but not 0. The reason is that a small number of good users remain without neighbors for several claims and, consequently, their trust score is decreased; similarly, their trust score trend may seem malicious. Thus, their truthful claims are rejected if they have no neighbors. The users can overcome this rare issue if they are made aware that the protocol works best when they have neighbors.

If malicious claimers do not broadcast certifying requests, a few of their claims are accepted initially because it appears that they have no neighbors. If a claimer continues to send this type of claim, her trust score falls below the "good" user threshold and all her future claims without verifiers are rejected. Thus, the false negative rate will become almost 0 over time. The false positive rate remains very low in this case.

Sometimes malicious individual claimers. In this set of experiments, a malicious user attempts to "game" the system by sending not only malicious claims but also truthful claims to improve her trust score. We have evaluated two scenarios: (1) Malicious users sending one truthful claim, followed by one false claim throughout the simulation, (2) Malicious users sending one false claim for every four truthful claims. For the first 10 minutes of the simulation, they send only truthful claims to increase their trust score. Furthermore, these users do not broadcast certifying requests to avoid being proved wrong by others.

Figure 2 shows that LINK quickly detects these malicious users. Initially, the false claims are accepted because the users claim to have no neighbors and have good trust scores. After a few such claims are accepted, LINK detects the attacks based on the analysis of the trust score trends and punishes the attackers.

Figure 3 illustrates how the average trust score of the malicious users varies over time. For the first type of malicious users, the multiplicative decrease followed by an additive increase cannot bring the score above the "good" user threshold; hence, their claims are rejected even without the trust score trend analysis. However, for the second type of malicious users, the average trust score

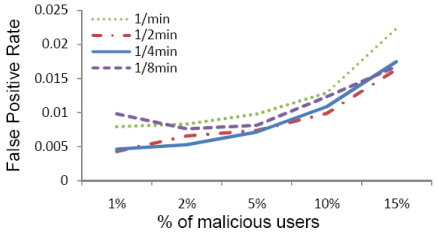

Fig. 4. False positive rate as a function of the percentage of malicious verifiers for different claim generation rates. The average speed is 1m/s.

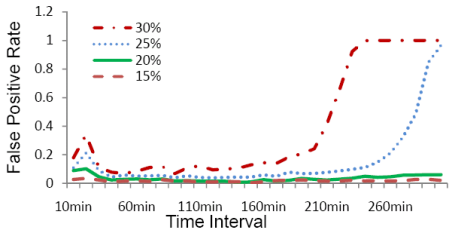

Fig. 5. False positive rate over time for different percentages of malicious verifiers. The claim generation rate is 1 per minute and the average speed is 1m/s.

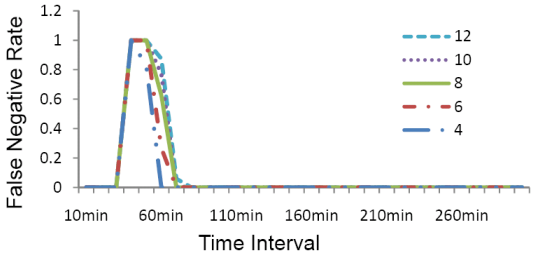

Fig. 6. False negative rate over time for colluding users. Each curve is for a different colluding group size. Only 50% of the colluding users participate in each verification, thus maximizing their chances to remain undetected.

is typically greater than the "good" user threshold. Nevertheless, they are detected based on the trust score trend analysis.

Always malicious individual verifiers. The goal of this set of experiments is to evaluate LINK's performance when individual malicious verifiers try to slander good claimers. In these experiments, there are only good claimers, but a certain percentage of users will always provide malicious verifications.

From Figure 4, we observe that LINK performs well even for a relatively high number of malicious verifiers, with a false positive rate of at most 2%. The 2% rate happens when a claimer has just one or two neighbors and those neighbors are malicious. However, a claimer can easily address this attack by re-sending a claim from a more populated area to increase the number of verifiers.

Of course, as the number of malicious verifiers increases, LINK can be defeated. Figure 5 shows that once the percentage of malicious users goes above 20%, the false positive rate increases dramatically. This is because the trust score of the slandered users decreases below the threshold and they cannot participate in verifications, which compounds the effect of slandering.

Colluding malicious claimers. This set of experiments evaluates the strongest attack against LINK. Groups of malicious users collude, using out-of-band communication, to verify for each other. Furthermore, colluding users can form arbitrary verification subgroups; in this way, their collusion is more difficult to detect. To achieve high trust score for the colluding users, we con-

sider that they submit truthful claims for the first 30 minutes of the simulation. Then, they submit only malicious claims.

Figure 6 shows that LINK's dynamic mechanism for collusion detection works well for these group sizes (up to 6% of the total nodes collude with each other). After a short period of high false negative rates, the rates decrease sharply and subsequently no false claims are accepted.

6 Related Work

Location authentication for mobile users has been studied extensively so far. To the best of our knowledge, all existing solutions employ trusted network/localization infrastructure [3, 4, 10, 11, 12, 13, 14] to detect malicious users claiming false locations. Most of these solutions use distance bounding techniques, in which a beacon acting as verifier challenges the mobile device and measures the elapsed time until the receipt of its response.

None of these solutions, however, can be directly applicable to scenarios that involve interaction between mobile users and third-party services (i.e., services that do not have direct access to the network/localization infrastructure). The main novelty of LINK comes from employing mobile users (more exactly their mobile devices) to certify the location claimed by other users.

Similar to our work, SMILE [15] and Ensemble [16] use information collected by mobile devices (keys from nearby users or received signal strength – RSS – values) to provide mutual co-location verification for mobile users. However, they do not provide location verification. RSS signatures in conjunction with RSS fingerprinting could be used for location verification, but such solutions do not scale due to the very dense fingerprinting required to achieve good accuracy.

As it is based on trust scores, LINK shares a number of similarities with work on reputation systems for P2P and mobile ad hoc networks. For example, CONFIDANT is a protocol [17] that avoids node misbehavior by establishing trust relationships between nodes based on direct and indirect observations reported by other nodes. The CORE protocol [18] takes a similar approach and uses reputation to enforce node cooperation. In contrast with CONFIDANT, CORE requires reputation values received from indirect observations, thus preventing malicious nodes from wrongfully accusing legitimate nodes.

There are two main differences between this type of solution and LINK. First, LINK cannot monitor indirectly additional user actions (such as packet forwarding or file sharing) to assess the trust. Second, LINK employs the centralized LCA to have a global view of the the entire system. As such, it is able to detect malicious trust score trends and collusion attacks.

7 Conclusions

This paper presented LINK, a protocol for location authentication based on certification among mobile users. LINK can be successfully employed to provide location authentication for location-based services without requiring cooperation from the network/localization infrastructure. The simulation results have demon-

strated that several types of attacks, including strong collusion-based attacks, can be quickly detected while maintaining a very low rate of false positives.

Acknowledgment. This research was supported by the National Science Foundation under Grant No. CNS 0831753. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. T. Kindberg, L. Zhang, and N. Shankar, "Context authentication using constrained channels," in *Proc. of WMCSA '02*, 2002, pp. 14–21.
2. C. Wullems, O. Pozzobon, and K. Kubik, "Trust your receiver? Enhancing location security," *GPS World*, vol. 1, pp. 23–30, Oct 2004.
3. S.Brands and D. Chaum, "Distance-bounding protocols," in *Proc. of EUROCRYPT '93*, May 1993, pp. 344–359.
4. K. Rasmussen and S. Čapkun, "Location privacy of distance bounding protocols," in *Proc. of the 15th ACM Conference on Computer and Communications Security (CCS'08)*, Oct 2008, pp. 149–160.
5. A. Vora and M. Nesterenko, "Secure location verification using radio broadcast," *IEEE Trans. Dependable Secur. Comput.*, vol. 3, no. 4, pp. 377–385, 2006.
6. J. Douceur, "The Sybil Attack," in *Proc. of IPTPS '01*, 2002, pp. 251–260.
7. X. Chu, X. Chen, K. Zhao, and J. Liu, "Reputation and trust management in heterogeneous peer-to-peer networks," *Springer Telecommunication Systems*, vol. 44, no. 3-4, pp. 191–203, Aug 2010.
8. M. Talasila, R. Curtmola, and C. Borcea, "LINK: Location-verification through immediate neighbors knowledge," Department of Computer Science, NJIT, Tech. Rep., 2010.
9. D. Singelee and B. Preneel, "Location verification using secure distance bounding protocols," in *Proc. of the 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'05)*, Nov 2005, pp. 834–840.
10. N. Tippenhauer and S. Čapkun, "Id-based secure distance bounding and localization," in *Proc. of ESORICS '09*, 2009, pp. 621–636.
11. J. T. Chiang, J. Haas, and Y.-C. Hu, "Secure and precise location verification using distance bounding and simultaneous multilateration," in *Proc. of the 2nd ACM Conference on Wireless Network Security (WiSec'09)*, 2009, pp. 181–192.
12. N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *Proc. of the 2nd ACM Workshop on Wireless Security (Wise'03)*, 2003, pp. 1–10.
13. N. Chandran, V. Goyal, R. Moriarty, and R. Ostrovsky, "Position based cryptography," in *Proc. of CRYPTO '09*, 2009, pp. 391–407.
14. V. Shmatikov and M.-H. Wang, "Secure verification of location claims with simultaneous distance modification," in *Proc. of the 12th Annual Asian Computing Science Conference (Asian'07)*, Dec. 2007, pp. 181–195.
15. J. Manweiler, R. Scudellari, and L. Cox, "SMILE: Encounter-based trust for mobile social services," in *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, 2009, pp. 246–255.
16. A. Kalamandeen, A. Scannell, E. de Lara, A. Sheth, and A. LaMarca, "Ensemble: cooperative proximity-based authentication," in *Proc. of MobiSys '10*. ACM, 2010, pp. 331–344.
17. S. Buchegger and J.-Y. L. Boudec, "Performance Analysis of the CONFIDANT Protocol," in *Proc. of MobiHoc '02*, 2002, pp. 226–236.
18. P. Michiardi and R. Molva, "CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Proc. of the IFIP TC6/TC11 6th Joint Working Conference on Communications and Multimedia Security*, 2002, pp. 107–121.