International Journal For Technological Research In Engineering
Volume 1, Issue 11, July-2014 *ISSN (Online): 2347 - 4718*

TOWARD SECRECY CONSERVING AND CONNIVANCE RESISTANCE IN A LOCATION PROOF UPDATING SYSTEM

Shilpa S.M[1], Dr. Suresh M.B[2]
[1]M-Tech, Department of Computer Network Engineering
[2]Prof & Head, Department of Information Science
East West Institution of Technology, Bangalore, India,

Abstract: Today's position -sensible service relies on user's mobile device to ascertain the current location. This grants enormous users to access a curbed resource or furnish bogus alibis by betraying on their positions. To address this issue, we propose A Privacy- saving position proof Updating System (APPLAUS) in which collocated Bluetooth enabled mobile devices mutually bring forth location proofs and send informs to a location proof server. Periodically changed pseudonyms are used by the mobile devices to protect source position privacy from each other, and from the untrusted location proof server. We also develop user-centric location privacy model in which mortal users evaluate their location privacy levels and decide whether and when to accept the location proof calls for. In order to defend against colluding attacks, we also present beur regarding-based and correlation clustering-based accesses for occupant detection. APPLAUS can be implemented with existing network substructure, and can be easily distributed in Bluetooth enabled mobile devices with little computation or power cost. Covering experimental results show that APPLAUS can efficaciously provide location proofs, importantly preserve the source position privacy, and effectively detect colluding flacks.

I. INTRODUCTION

The position established services take reward of user position entropy and provide mobile users with various imaging sand services. Nowadays, more and more location- based applications and services require users to provide location proofs at a particular time. For example, "Google Latitude" and "Loopt" are two services that enable users to track their friends' locations in real time. These applications are location-sensitive since position proof plays a critical role in enabling these applications. There are many kinds of location-sensitive applications. One category is location-based access control. For example, a hospital may allow patient information access only when doctors or nurses can prove that they are in a particular room of the hospital. Another class of location-sensitive applications require users to provide past location proofs , such as auto policy estimate in which auto policy companies offer rebates to drivers who can prove that they take safe routes during their daily commutes, police investigations in which police detective are interested in finding out if a person was at a murder scene at some time, and location-based social communicating in which a user can ask for a location proof from the service requester and accepts the request only if the sender is able to present a valid location proof. The common theme across these location sore applications is that they offer a payoff or benefit to users In implement, the opponent can thus be a rogue individual, a set of malicious mobile nodes, or eavesdropping devices in the network. In the worst case, it is possible that the untrusted location proof server may be compromised by the adversary and the location information can then be easily inferred by analyzing the records of location proofs, e.g., the adversary could apply statistical testing such as K-S test to identify a user while no real identity is included. Therefore, we need to fittingly design and arrange the position proof records in the untrusted node and protecting the location privacy of peer nodes from each other, from the adversary, Oren from the untrusted location proof server to prevent other parties from finding out a node's past and current location entropy. Position privacy from each other, and from the untrusted location proof server. We develop a user-centric location privacy model in which individual users evaluate their position privacy levels in real time and decide whether and when to accept a position proof request. In order to defend against colluding attacks, we also present betweenness ranking-based and correlation clustering-based approaches for outlier detection. Extensive experimental and simulation results based on multiple data sets show that APPLAUS can effectively provide location proofs, significantly preserve the source location privacy, and effectively detect conspiring attacks. The rest of the paper is organized as follows: We first introduce preliminaries of our scheme in Section 2, and then present our location proof updating scheme in Section 3.Section 4 presents the source location privacy analysis.

II. PRELIMINARIES

In this paper, we focus on mobile networks where mobile devices such as cellular phones communicate with each other through Bluetooth. In our implementation, mobile devices periodically initiate position proof requests to all contiguous devices through Bluetooth. After receiving a request, a mobile node decides whether to exchange location proof, based on its own location proof updating requirement and its own privacy consideration. Given its appropriate range (about 10 m) and low power use, Bluetooth is a natural choice for mutual encounters and location proof exchange.

A. Pseudonym

As commonly used in many networks, we consider an online Certification Authority (CA) run by independent trusted third

party which can install certification for the mobile devices. Similar to many pseudonym approaches, to protect location privacy, every mobile node i registers with the CA by preloading a set of M public/private key pairs K Pubic before entering the network. The public key K Pubic is used to serve as the pseudonym of node i. The private key KPrvenables node i to digitally sign messagesso that the receiver can validate the signature authenticity.Due to the broadcast nature of wireless communication, probes are used for mobile nodes to discover the neighbors. When a node i receives a probe from another node, it checks the certificate of the public key of the sender and the physical identity, e.g., Bluetooth MAC address. After that, i verify the signature of the probe message. Later on, if privacy is required, a security tie is established (e.g., with Diffie-Hellman).

B. Threat Model

We assume that an opponent aims to track the position of a mobile node. An opponent can have the like certificate as a mobile node and is fitted to eaves drop communications. We assume that the opponent is inner, passive, and global. By internal, we mean that the adversary is able to via media or control individual mobile device and then communicate with others to explore private data, or single devices may collude with each other to give false proofs. We assume that the number of colluders is small compared with that of valid devices. In the worst case, the adversary could via media the location proof server to get the stored location proof records. However, it is not able to take control of the server to work as a colluder, since once compromised, the attack will be detected promptly and the location proof server will be replaced by a back-up server. The same premise applies to the CA. By passive, we assume the adversary cannot perform active channel jamming, mobile worm attacks or other denial-of service attacks, since these attacks are not related to location privacy. By global, we assume the adversary can monitor, eavesdrop, and analyze all the traffic in its neighboring area, or even monitor all the traffic around the server. In practice, the adversary can thus be a rogue individual, a set of malicious mobile nodes, or eavesdropping devices in the network.

In the worst case, it is possible that the untrusted location proof server may be compromised by the adversary and the location information can then be easily inferred by examining the records of location proofs, e.g., the adversary could apply statistical testing such as K-S test to identify a user although no real identity is included. Therefore, we need to appropriately design and arrange the location proof records in the untrusted server so that no private information related to individual users will be revealed even after it is compromised. Hence, the problem we address in this paper consists of collecting a set of location proofs for each peer node and protecting the location privacy of peer nodes from each other, from the adversary, or even from the untrusted location proof server to prevent other parties from learning a node's past and current position information.

Fig. 1. Location proof updating architecture and message flow.

C. Location Privacy Level

In this paper, we use many pseudonyms to protect location privacy; i.e., mobile nodes change at the regular interval pseudonym used to contract contents, thus lessen their long term link ability. To head off special correlation of their location, mobile nodes in closeness coordinate anonym changes by using silent fuse zonas[16], [17], or areas where the opponent has no reporting [4]. Without loss of generalization, we assume each node alters its anonyms from time to time according to its private necessity. If this node changes its anonym at least once during a time period of time (mix zone), a mix of its identity and location occurs, and the mix zone goes a confusion point for the adversary. Consider a mobile network composed of N mobile nodes and each node has M anonyms.

III. THE LOCATION PROOF UPDATING SYSTEM

In this part, we introduce the location proof updating architecture, the communications protocol, and how mobile knobs docket their location proof updating to achieve location to protect in APPLAUS.

A. Architecture

In applaus mobile nodes communicate with neighboring node through Bluetooth, and communicate with the untrusted server through the cellular network based on the roles they play in the process of location proof updating, they are categorized as prover, witness, location proof server, certificate authority or verifier .The message flow of applaus is shown in the architecture diagram fig 1 Prover: will collect the information from surrounding nearby nodes when a location proof needs time t ,then prover will send a location proof request to its nearby nodes if no positive result is received the prover will generate junk value and submit. Location proof server: Retrieve the location proof information and stores the history and records of the location proof server is needed to store the history of location proofs. Also helps in exchanging the information without out anything intervening with the prover nodes who submit the location proof it is impossible for the attackers to reveal the

real source of the location proof. Certificate authority: It works as a bridge between the verifier and the location proof server. It is able to retrieve location proof server and send it to verifier CA is most commonly used in many networks here we consider an online CA and preloads a set of public/private key pairs before entering into the network only CA will know the mapping between the real identity and pseudonyms. Verifier: unknown user or an application who is authorized to verify a provers location within a specific time period. The verifier is usually associated with the prover. Eg: friends or colleagues, to be trusted enough to gain authorization.

B. Protocol

When a prover needs to collect location proof at time t it executes protocol in fig 2 to get hold of location proofs from the nearby nodes within Bluetooth communication range….each and every node uses its M pseudonyms PMi¼1 as its identity throughout the communication.

1 The prover broadcasts a location proof request to its nearby nodes through Bluetooth based on its update scheduling. The request contains the prover current pseudonyms Pprov, and a random number Rprov.

2 The witness decides whether to accept the location once agreed,it will to produce a location proof for both prover ,the location proof includes the prover pseudonym Pprov, prover's random number Rprov, witness's current time stamp Twit, witness's pseudonym Pwitt, and their shared location Laths proof is signed and hashed by the witness unable to deny this proof. It also encrypted by servers public key to prevent from traffic monitoring or listening secretly.

3 Once the location proof is received, theprover is responsible for to yield this proof to the location proof server. The text also includes provers pseudonym pseudonym Pprov and random number Rprov, or its own location for confirmation purpose.

4 An empowered voucher can question the CA for location proofs of specific prover. This question holds a true identity to its similar association pseudonyms during that time interval and retrives their location proofs from the server. In order not to disclose correlation amongst pseudonyms to the location server, CA will always collect sufficient questions from k are sent out.

5 the location proof server only brings back hashed location instead the literal location to the CA who then forwards to the voucher. The voucher compares the hashed location with the claimed location acquired from the prover to decide if claimed location is authentic. In order to keep the CA from knowing location of a literal identity. The location proof server computes the hash of each location and only sends the hashed location to the CA in step 5.

C. Scheduling location proof updates

As discussed earlier, the opponent or enemy may obtain complete coverage and track nodes throughout the entire network, by flexible the location proof server and obtain all history location proof server and obtain all history location proof updating schedules so no origin location data related to individual user is disclosed even if server is flexible Say a mobile node i has a band of pseudonyms P1; P2; . . . ; PM which alter sporadically, and distinct parameters for each pseudonym are preset If each pseudonym P_j updates its location proofs such that the interrupt date interval follows Poisson distribution with parameter p_j, then the entire interrupt date intervals for node i follow Poisson distribution with a parameter it has the attributes of n pseudonym unclipped and statistically firm source location unperceivable. The elaborated scheduling protocol for the prover is shown in Algorithm 1. The predefine dup dating parameter finds out how often location proofs are updated. In some cases, no location proof is generated when the location proof updating time comes. To ensure that location proof updating follows the Scheduled Poisson distribution, a junk proof is generated and submitted. The junk proof has the same format as the literal location proof and cannot be separated by the attackers.

D. Source location privacy Analysis

In this part we discuss the location privacy menace in our system, as well as our countermeasures. We first await at how an opponent may disclose location data by examining the location proof account .say the attackers has enough resource First, the aggressor may merely monitor and analyze the content of a record that comprise of the user's identity and location. Second, even if the user's ID is encrypted or pseudonym zed, it is easy for the opponent to draw endorses all the location actions associated to the same ID once its pseudonym is detected. Third, even though the user's anonyms change sporadically, it is still possible for the opponent to deduce this user's other pseudonyms from one pseudonym if these pseudonyms alter at like time or locations. Moreover, the aggressor may do more advanced traffic analysis including rate monitoring and location correlation. In a rate supervising attack, the attacker tries to monitor and correlate location proof updating rates from unlike pseudonyms. In a location correlation attack, the attacker may observe the correlation in the updated location between a node and its neighbors.

IV. COLLUDING ATTACKS AND COUNTERMEASURES

The joint issues of location proof and location secrecy, but the menace of colluding attack is still a clear issue. This threat exists when two nodes collude with each other to give fake location proofs. For example, when a dishonest node C1 from San Francisco needs to prove herself in New York City (NYC), she can have another colluding node C2 to generate bogus location proofs for her, with location tag of New York City. Generally speaking, such attacks can be identified by looking into the location draws and analyzing the interactions amongst colluders as well as the time and location content along the moving trajectory. We first consider statistical threshold based solution in which the system requires the prover to obtain a number of spectator nodes, no matter what their real identities are.

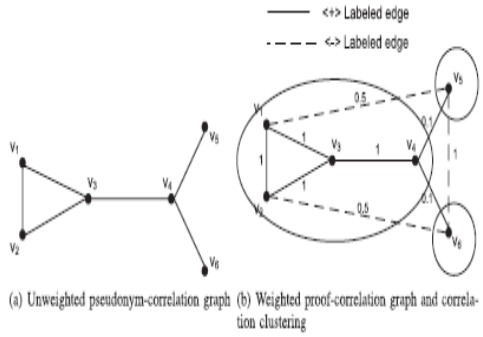

Fig 2 Illustrate a colluding case

As we know, the location proof server has information about the number of anonyms at a particular time and location. This data can be used to guess whether the prover lies down about not finding enough peers or always finding the same peer based on some statistical techniques. More specifically, the server-level detection is did higher than preselected threshold, PA is considered as a good node. Fig. 2 illustrates a colluding case where node PA tries to claim her location in New York city with her colluders although she is not at New York City. Although PA finds a colluder PB who is located in New York City to generate a bogus location proof for her, it can't use other provers since she is not at NYC. The location proof server looks through the location proof records to check if there are other provers in PA's communication range. In this example, several other nodes (e.g., PE, PF, and PG) exist. It is easy to calculate the trust level of PA's location proof is which is below the threshold. Thus, PA is listed as suspicious to be untrusted. Calculating the trust level of a location proof involves the examination of its surrounding location proofs for both prover and spectator, as well as large amount of redundant calculations between individual location proofs. To overcome this problem, we develop techniques that can perform verifications on a set of location proofs which are relevant in time and space, rather than individual proofs. We present two approaches to detect suspicious location proofs and pseudonyms: betweenness grading and correlation bunch. The betweenness grading approach calculates the rating of each pseudonym in a graph and then ranks these pseudonyms based on their grading value. The pseudonyms with low ranking are considered as untrusting nodes. The correlation bunch access takes into account the time lag between two nearby nodes location proofs, and uses an altered correlation clustering algorithm on a secular - angled graph to rule out occupant clusters, which are considered as fishy location proofs. Both approaches use undirected graph to shine the relationship

V. CONCLUSION

In this paper, we proposed a privacy-preserving location proof updating system called APPLAUS, where collocated Bluetooth enabled mobile devices mutually generate location proofs and upload to the location proof server. We use statistically changed pseudonyms for each device to protect source location privacy from each other, and from the untrusted location proof server. We also develop a user centric location privacy model in which individual users evaluate their location privacy levels in real time and decide whether and when to accept a location proof exchange request based on their location privacy levels. To the best of our knowledge, this is the first work to address the joint problem of location proof and location privacy. To deal with colluding attacks, we proposed between ness ranking based and correlation clustering-based approaches for outlier detection. Extensive experimental and simulation results show that APPLAUS can provide real-time location proofs effectively. Moreover, it preserves source location privacy and it is collusion resistant.

REFERENCES

[1] A.R. Beresford and F. Stajano, "Location Privacy in Pervasive Computing," IEEE Security and Privacy, 2003.

[2] U. Brandes, "A Faster Algorithm for Betweenness Centrality,"J. Math. Sociology, vol. 25, no. 2, pp. 163-177, 2001. ZHU AND CAO: TOWARD PRIVACY PRESERVING AND COLLUSION RESISTANCE IN A LOCATION PROOF UPDATING SYSTEM 63

[3] S. Brands and D. Chaum, "Distance-Bounding Protocols," Proc.Workshop Theory and Application of Cryptographic Techniques on Advances in Cryptology (EUROCRYPT '93), 1994.

[4] L. Buttya´n, T. Holczer, and I. Vajda, "On the Effectiveness of Changing Pseudonyms to Provide Location Privacy in VANETs," Proc. Fourth European Conf. Security and Privacy in Ad-Hoc and Sensor Networks, 2007.

[5] S. Capkun and J.-P. Hubaux, "Secure Positioning of Wireless Devices with Application to Sensor Networks," Proc. IEEE INFOCOM, 2005.

[6] L.P. Cox, A. Dalton, and V. Marupadi, "SmokeScreen: Flexible Privacy Controls for Presence-Sharing," Proc. ACM MobiSys, 2007.

[7] E.D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica, "Correlation Clustering in General Weighted Graphs," Theoretical Computer Science, vol. 361, nos. 2/3, pp. 172-187, 2006.

[8] N. Eagle and A. Pentland, "CRAWDAD Data Set mit/reality(v.2005-07-01)," http://crawdad.cs.dartmouh.edu/mit/reality, July 2005. Proc. 16th ACM Conf. Computer and Comm. Security (CCS), 2009

[9] J. Freudiger, M.H. Manshaei, J.P. Hubaux, and D.C. Parkes, "On Non-Cooperative Location Privacy: A Game-Theoretic Analysis," Proc. 16th ACM Conf. Computer and Comm. Security (CCS), 2009.

[10] B. Gedik and L. Liu, "A Customizable K-Anonymity Model for Protecting Location Privacy," Proc. IEEE Int'l Conf. Distributed Computing Systems (ICDCS), 2005.

[11] M. Gruteser and D. Grunwald, "Anonymous Usage of Location- Based Services through Spatial and Temporal Cloaking," Proc. ACM MobiSys, 2003.

[12] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.C. Herrera, A.M. Bayen, M. Annavaram, and Q. Jacobson, "Virtual Trip Lines for Distributed Privacy-Preserving Traffic Monitoring," Proc. ACM MobiSys, 2008.
[13] T. Jiang, H.J. Wang, and Y.-C. Hu, "Location Privacy in Wireless Networks," Proc. ACM MobiSys, 2007.
[14] V. Kostakos, "Experiences with Urban Deployment of Bluetooth," presentation given at the Univ. of California, San Diego, Mar. 2007.
[15] V. Lenders, E. Koukoumidis, P. Zhang, and M. Martonosi, "Location-Based Trust for Mobile User-Generated Content: Applications Challenges and Implementations," Proc. Ninth Workshop Mobile Computing Systems and Applications, 2008.
[16] M. Li, R. Poovendran, K. Sampigethaya, and L. Huang, "Caravan: Providing Location Privacy for VANET," Proc. Embedded Security in Cars (ESCAR) Workshop, 2005.
[17] M. Li, K. Sampigethaya, L. Huang, and R. Poovendran, "Swing & Swap: User-Centric Approaches Towards Maximizing Location Privacy," Proc. Fifth ACM Workshop Privacy in Electronic Soc., 2006.
[18] Y. Li and J. Ren, "Source-Location Privacy Through Dynamic Routing in Wireless Sensor Networks," Proc. IEEE INFOCOM, 2010.
[19] W. Luo and U. Hengartner, "Proving Your Location Without Giving Up Your Privacy," Proc. ACM 11th Workshop Mobile Computing Systems and Applications (HotMobile '10), 2010.
[20] J. Manweiler, R. Scudellari, Z. Cancio, and L.P. Cox, "We Saw Each Other on the Subway: Secure Anonymous Proximity-Based Missed Connections," Proc. ACM 10th Workshop Mobile Computing Systems and Applications (HotMobile '09), 2009.
[21] J. Manweiler, R. Scudellari, and L.P. Cox, "SMILE: Encounter- Based Trust for Mobile Social Services," Proc. ACM Conf. Computer and Comm. Security (CCS), 2009.
[22] F.J. Massey Jr., "The Kolmogorov-Smirnov Test for Goodness of Fit," J. Am. Statistical Assoc., vol. 46, no. 253, pp.68-78, 1951.

