Scheduled Location Transparency for Privacy-Protected MANET Location Services

Michael Schäfer*, Christoph Reich* and Tim Leinmüller+
*Hochschule Furtwangen University, Faculty of Computer Science,
{schaefmi|rch}@hs-furtwangen.de
+DENSO AUTOMOTIVE Deutschland GmbH, Technical Research Department,
t.leinmueller@denso-auto.de

Abstract

Mobile ad-hoc networks (MANETs) that use geographic routing protocols may lead to a severe privacy breach since the current position of any node in the network can be resolved. By changing the node IDs frequently the location privacy of nodes can be achieved. It is disadvantageous that unicast communication between nodes is not possible any more. With the scheduled location transparency approach nodes know other nodes' IDs at a pre-arranged start and end time. During this time interval a unicast communication is possible again.

1 Introduction

With pervasiveness of mobile computing technology and wireless communication, mobile ad hoc networks (MANETs) could be a key networking technology of the future. By forming wireless grids the mobile users will be able to share digital resources between their different mobile computing devices, leverage computing power or other resources from remote nodes. Many MANETs use a geographic routing protocol [8], which utilizes the geographic position of a mobile node as network address and performs packet delivery via geographic forwarding algorithms. Every node sending unicast packets to another node, which is not in its direct radio range, has to address these packets to the geographic position of the desired receiving node, hence has to know the receiving nodes position. The location of a node can be resolved via the MANET location service (e.g. [13],[14]), a central or distributed service which can be queried for the position of any arbitrary node in the network. The location services itself and the corresponding network traffic may lead to a severe privacy breach, where network insiders and even outsiders are able to resolve and record the current geographic position of any node which is part of the network.

This paper is organized as follows. In the second chapter we present related work in the field of privacy support for MANETs. In third chapter we discuss privacy issues emphasizing the importance of the privacy support for MANET's location service. In the fourth chapter we present our proposed solution for a privacy protected location-service with scheduled location transparency, which then is evaluated in the following chapter. Chapter six outlines future work and finally the seventh chapter concludes this paper.

2 Related Work

The baseline terminology for anonymity, unobservability and pseudonymity is given by [15]. In Germany, the legal framework for the processing of personal data is provided by the German privacy protection law BDSG [5], the right to personal privacy is also specified in the Universal Declaration of Human Rights [1]. The study [6] extensively investigates location-privacy from a sociological perspective. It explores whether different types of users are willing to disclose their own location and to whom, depending under which circumstances, the disclosure takes place.

The paper [3] explores the problems pervasive computing brings for location privacy and proposes a possible solution, which is based on so-called mix-zones. The authors discovered several new and unresolved problems in the field of location-privacy and presented them for further research.

[7] depicts the privacy issues of MANETs from a car manufacturer perspective. The author concludes that neither full anonymity nor no privacy at all satisfies their specific requirements and that an architecture which manages the privacy related parts of the network is needed. Different mechanisms and technologies which are needed to build such an architecture are presented by means of an example.

[9] provides an in-depth analysis of location privacy in

ubiquitous systems, tracking attacks against known privacy protecting techniques and possible countermeasures.

In the article [12] the security and privacy issues for VANETs (Vehicular Ad Hoc Networks) are discussed. The author also explains how the VANET-technology could be introduced without being perceived as a big brother like monitoring tool.

[17] discusses the traceability of users of mobile communication networks. The authors isolated five common classes of untraceability requirements and provided new protocols which allow pseudonym authentification.

[2] analyzes the possibility of using random node IDs in MANETs to preserve node privacy, [18] explains the impact of pseudonym changes on the performance of geographic routing protocols.

[11] explains a method for the recognition of nodes only by their specific radio transmitter signature, which may undermine privacy-supporting technologies at upper layers.

[10] analyzes the anonymity of periodic location samples and presents tracker algorithms able to re-link movement paths by analyzing anonymous location data.

3 Privacy Issue

In Germany the processing and storage of personal data is regulated by the German privacy protection law BDSG [5]. Every company or system which processes such data is bound to this legal framework. Similar frameworks exist in other countries, for the European Union. Therefore the location information available by the MANET's location services must be protected. A extensive discussion about location privacy can be found in Consolvo et. al [6]. To protect the location data of MANET nodes the changing or random node IDs approaches have been proposed ([7, 2, 4, 16]). Instead of using a fixed node ID, nodes use a set of pseudonyms. Nodes then change their node IDs frequently using these pseudonyms. The pseudonyms could be simple random IDs or created and signed in advance by a trusted third party or even cryptographically derived from existing IDs. The change frequency is dependent on application and privacy requirements. However, simply changing the node IDs without taking the current context into consideration is unlikely to lead to the desired level of location privacy. Much more advanced algorithms exist to link anonymous location samples by tracking movement paths and other advanced or application dependent properties, see [10, 9, 3].

The introduction of changing node IDs for the protection of location privacy results in a problem that is yet unresolved in literature: MANET location services can no longer work and therefore unicast communication is no longer possible in such a network. The privacy protection of the changing node IDs results from the fact that nobody can predict which node IDs are used by which node and what time, otherwise the protection would not work. If a node cannot determine the network ID which is currently used by the node it wants to communicate with, it cannot query the location service for the target node's current location. Without this location unicast communication is not possible in a network using position based routing.

4 Proposed Solution

As investigated in [6] users want to be in control about their location and movement data. Neither a system where location data can be easily gathered by others, nor a system where all location data is protected by changing node IDs appears to be a satisfying concept. The first violates privacy demands, the second prevents location services from working and thus promising MANET applications are broken. Therefore we developed a new approach that allows users of the MANET to limit the disclosure of their location information to specific other nodes or parties. Even though location data is protected by changing node IDs, nodes and parties that established a trust relationship between them are enabled by our mechanism to use the location service and communicate via unicast. Parties outside the MANET for which the user has established a trust relationship are able to query for the current location of the user's node and send unicast messages into the MANET. All nodes, parties, passive receivers and active attackers are unable to track the node or request its current position from the location service.

Our solution makes use of hash-functions and so-called hash-chains, which are explained in the following section.

4.1 Hash-functions and hash-chains

A hash-function $h = hash(m)$ is a deterministic function that takes a string or message m as input and produces the hash-value h as output. The length of h is always the same, independent of the length of the input message m. The hash-function is pre-image resistant, which means that given a hash-value h it is not possible to calculate any m to fulfill $h = hash(m)$, m can only be brute-forced. The hash-function is also second pre-image resistant, which means given an input m_1, it is not possible to find another input, m_2, so that $hash(m_1) = hash(m_2)$. Last but not least the hash-function is collision-resistant, which means it is not possible to create two messages m_1, m_2 which result in the same hash-value so that $hash(m_1) = hash(m_2)$. Commonly used cryptographic hash-functions are for example MD5, SHA1, SHA256 or RIPEMD-160.

A hash-chain is a set of hash-values created by repeatedly applying the hash-function on its own output (Figure 1). A hash-chain of the length l (Figure 1: $l = 5$)

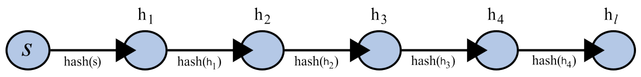

Figure 1. Hash chain $l = 5$

is created by using a nonce or other seed value s and a hash function $hash()$. Now we generate the first value of the chain by applying $hash()$ to the seed-value s yielding the hash-value $h_1 = hash(s)$. h_1 is saved in the chain and the hash-function is applied to its own output yielding the second hash-value $h_2 = hash(h_1)$. The function $h_n = hash(h_{n-1})$ is applied l-times while every hash value is stored in the chain. The chain can also be denoted as $hash^l(s)$.
Hash-chains have certain useful properties. Two parties which secretly share the relatively small seed s can compute hash-chains of arbitrary lengths. A chain of the length l produces l-hash-values. Everybody knowing the seed s can compute any h_n, with $1 < n < l$, either on the fly or by computing all hash-values of the chain in advance and storing them. If h_d is publicly disclosed anybody can calculate h_n with $d < n < l$, but still nobody can calculate h_n for $n < d$, except anyone knowing the seed s.

4.2 Location Transparency

Our solution is centered around the idea of putting the users back into control about their location data. Nodes still use changing node IDs but they do not change them randomly or sequentially but according to an algorithm that makes use of reverse hash-chains. By using a hash-chain as source of the node IDs nodes that share a common secret are able to predict the node IDs of the other node at any time while others can't.
In the initial situation, where no trust relationships between nodes have been established, every node changes its ID randomly, so the node IDs cannot be predicted and nodes cannot easily be tracked. Now we assume that node A wants to share its location data with node B. For this A has to establish a trust relationship with B via a secure channel. This can be realized if both nodes are in close proximity at the time of trust establishment or via a secure side-channel, for example via GSM/GPRS, GSM/SMS or Bluetooth. Node A defines four values: the nonce s, the change frequency f, the length l and the start time t_{start}. The nonce s is derived from a random number generator and later used as seed for a hash-chain. The frequency f determines the interval by which A will change its node ID and hereby the amount of tracking protection offered. The length l describes how many node ID change iterations the trust relationship should initially last. So the lifespan of a trust relationship established via these four values can be expressed as $\triangle t = l * f$. Finally the start time t_{start} defines at which exact point in

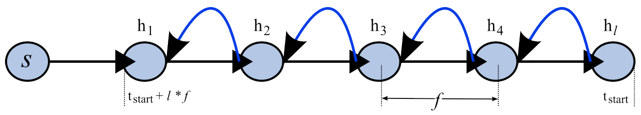

Figure 2. Reverse usage of hash-chain

time the trust relationship starts, so t_{end} can be defined as $t_{start} + l * f$. We assume that all nodes will have a synchronized time, e.g. via a time signal derived from GPS or Galileo.
To bootstrap the trust relationship node A sends a message to node B containing all four values, see message 1:

$$message : A \longrightarrow B : (s, f, l, t_{start}) \tag{1}$$

Both nodes now build up a hash-chain with the formerly exchanged seed value s and the length l. They either build it once and store it or build corresponding hash-values of the chain each time they need access to the chain.

Normally node A changes its node ID randomly with frequency f. Beginning at start time t_{start} the node A stops using random node IDs and starts using the hash-chain reversely to determine its next node ID. At time point t_{start} node A uses a node ID derived from the hash-value at the end of the chain, h_l. Until the next ID change it uses this ID to identify itself in the network and against the location service. When it is time for the next ID change node A uses an ID derived from the second last hash-value of the chain, h_{l-1}, and so on. Figure 2 depicts the usage of the previously calculated chain from Figure 1. This picture also explains the name reverse hash-chain, because the chain is used from the back. This process is repeated in frequency f until the lower end of the chain, h_1 is reached. After this the lifespan of the trust relationship is over and the node either has to refresh the trust relationship or starts being untraceable to everyone, including node B, by using random IDs again.

If now node B wants to query the location service for the current location of node A during the lifespan of the trust relationship it can calculate the current ID used by node A with the help of the values from message 1. First node B determines the $step_{now}$ in which the reverse-chain is currently used by node A, via equation 2. B calculates the difference between the current time t_{now} and the start time of the trust relationship t_{start} and then divides it by the frequency f.

$$step_{now} = \frac{t_{now} - t_{start}}{f} \tag{2}$$

Secondly B calculates the hash-value $h_{step_{now}}$ of the hash-chain and can derive node A's current ID from it. Via this ID node B can now query node A's position from the location service and use it to establish a unicast communication via position based routing.

While node B can now send geo-unicast messages to node A at any time during the lifespan of the trust relationship, node A can still not contact node B or query its position. This one-way location query possibility is still very useful. For example node B maybe an infrastructure connected, fixed node that acts as a gateway from a fixed network to the MANET. Services running in the fixed network want to contact node A in the MANET and therefore have to query for its current position. Since the position of node B is in this case fixed anyway a unidirectional trust relationship is sufficient for this application scenario.

If both, node A and node B, are moving MANET nodes and the application scenario requires that both can initiate unicast connection to the other node the bootstrapping process of the trust relationship has to be extended. After receiving node A's initial message (see message 1) node B also creates a nonce s_b and sends it via the secure channel to node A so that the entire process looks like in sequence 3.

$$\begin{aligned} A &\longrightarrow B : (s_a, f, l, t_{start}) \\ B &\longrightarrow A : (s_b) \end{aligned} \tag{3}$$

Starting at t_{start} both nodes now use node IDs derived from the corresponding reverse hash-chains seeded with s_a for node A and s_b for node B. Since both nodes know the four values s, f, l, t_{start} of the reverse hash-chain which the other node is using, both can calculate the current ID of the other and use it to query the location service.

At any time a node can decide to stop disclosing his location data to his trust partner by stopping to use the reverse hash-chain derived node IDs and switching back to using random IDs. At the end of the trust relationship's lifespan ($t_{start} + l * f$) the node can decide if it wants to refresh the trust relationship or not. If not it simply reverts to using random node IDs. If a node wants to sustain a trust relationship beyond the lifespan of the current hash-chain, it can use the unicast communication channel to the other node to build up a secure channel (for example using a Diffie-Hellmann key exchange and the nonce of the chain) and securely exchange the four, or, in the case of bidirectional trust relations, five new parameters for the initialization of the next hash-chains.

If a node wants to establish trust relationships with more than one other node there are two possible implementation variants:

One chain for all trust relationships: The assumption is that the lifespan of trust has to be long enough to enclose the trust relationships with all nodes. If a node is unavailable during the extension message exchange the trust relationship to this node is destroyed. With this variant a node can only stop disclosing its location data by destroying the trust relationship with all nodes, de-trusting single nodes is not possible. The advantage is that the node has to calculate and store one hash-chain and poses a low load to the location service.

A unique chain for every trust relation: This approach allows the extension and destruction of trust relations with other nodes selectively but requires the node to exchange, calculate and store hash-chains for every trust relationship. Additionally it is required that the node uses multiple node IDs at the same time, one for every trust relationship. This puts a high load on the network and the location service and may lead to an information leak allowing to identify the node in the network. The selection of the variant is dependent on the application scenario and corresponding requirements.

Summarized the presented concept allows the users of the MANET to disclose their location information in a self-controlled way and only to specific nodes. Uni-cast communication can be used with trusted parties while at the same time location privacy is preserved. The trust relationship to other nodes can be revoked at any time.

4.3 Retrospective Linking Attack

While our concept protects against untrusted nodes and attackers determining the current node ID and querying the location service under special circumstances there's still one attack vector. If an attacker is in the position to passively monitor and record the traffic of the *whole* MANET for a sufficient time he could be able to retrospectively identify a hash-chain in the recordings. Afterwards the privacy protection would be partially broken. An attacker capable of recording the traffic of the whole MANET has to be very powerful and presumably has other possibilities for location tracking at his disposal. An attacker recording the traffic of the whole MANET, takes one node ID, id_1, out of the recording, applies the hash function $hash()$ and then searches his recording for a node ID id_2 for which $id_2 = hash(id_1)$ is fulfilled. Repeated several times the attacker can isolate a single hash-chain from the recordings and calculates the upcoming hash-values of the chain by applying $hash()$ to the last recorded node ID of the isolated chain.

To mitigate such an attack we propose an extended version of our concept: When bootstrapping a trust relationship nodes not only create one nonce but two, s_1 and s_2, both are exchanged with the other node:

$$\begin{aligned} A &\longrightarrow B : (s_{a_1}, s_{a_2}, f, l, t_{start}) \\ B &\longrightarrow A : (s_{b_1}, s_{b_2}) \end{aligned} \tag{4}$$

Every node calculates two hash-chains, $h1$ and $h2$, one seeded with s_1, the other seeded with s_2. When deriving a node ID from the chains the node calculates the current hash-values of both chains, $h1_n$ and $h1_n$ and then applies a binary exclusive-or to them. The node ID is then derived from the result of the XOR-function:

$$h_n = h1_{step_{now}} \bigoplus h2_{step_{now}} \qquad (5)$$

Since the values of the hash-chain are now influenced by two parameters unknown to the attacker the effort to isolate a hash-chain is higher compared to a single hash-chain. Instead of just searching for a $id_2 = hash(id_1)$ event the attacker now has to pre-calculate all possible combinations of chains. This leads to a high number of possible transitions between one node ID and the next one, which all have to be compared against the recorded traffic. Additionally the number of detected transitions has the be much higher to reliably isolate a chain.

While providing protection against this special, powerful attack the extended concept puts more load on the bootstrapping process of trust-relations and demands more calculations and storage at the nodes. The load put on network and location service is unchanged. The decision if the extended concept should be applied depends on the specific application scenario and the corresponding security threat and risk model.

5 Evaluation

5.1 Requirements & Performance Impact

For the concept to be applicable to a MANET the participating nodes have to possess the corresponding computing power to regularly compute hash-functions. Storage for the initialization parameters has to be provided. Depending on the costs of processing power compared to the costs of storage capacity nodes could either pre-calculate the hash-chains and store them (less processing power but more storage needed) or calculate them on the fly (more processing power needed for timely calculation, less storage required). The node ID scheme of the MANET must allow for hash-value derived node IDs. Since the output-length can be adapted and most MANETs will use relatively large IDs (e.g. IPv6 addresses, 128 bits) this should normally be not a problem.

The impact of frequently changing node IDs on geographic routing has already been researched in [18]. Figure 3 shows the impact of ID changes with frequencies between 5 and 30 seconds. [18] further states that ID change frequencies over 60 seconds have little to zero impact on the performance of geographic routing. Change frequencies of 60 seconds and more still provide a very strong level of location privacy.

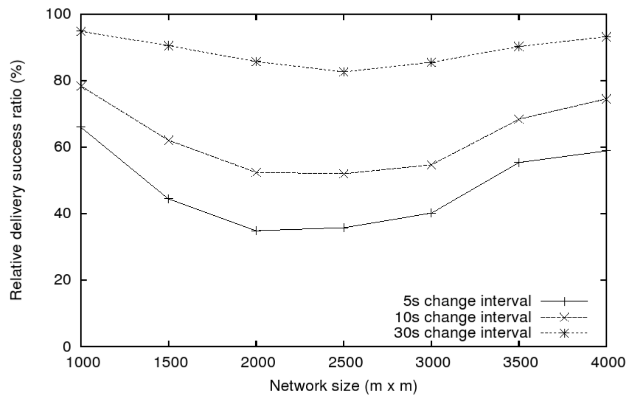

Figure 3. Impact of changing node IDs on delivery ratio [18]

The impact of the frequently changing node IDs on the performance of the Grid Location Service cannot be estimated easily. Research in [19], which analyzes GLS performance with different simulated mobility models may provide first pointers for further research. A reliable analysis should only be possible through network and GLS simulation.

5.2 Remaining Attacks

There are two remaining attack vectors. If an attacker, who is able to perform the retrospective linking attack against the unextended concept, possesses a very big amount of processing power he could still be able to re-link certain hash-chains. However such a powerful attacker likely has other possibilities to attack the location privacy of the users. Additionally it is always possible to add further strength to the hash-chains by enlarging the length of the hash-function output and the seed or by adding more combined hash-chains, of course at the cost of higher processing and storage requirements.
The second attack vector is not unique to our solution but also effects nodes using purely random, changing node IDs. Path tracking algorithms, so-called trackers, like presented and analyzed in [9, 10, 3] are able to re-link anonymous location samples retrospectively and isolate single nodes and movement paths out of datasets under specific conditions.

6 Future Work

Future work should analyze the impact of frequent changing node IDs on the performance of the Grid Location Service. Additionally several techniques could strengthen the process of deriving the IDs from the hash-chains against the retrospective relinking attack. In the field of the aforementioned tracking algorithms several ideas and methods have been developed to protect against retrospective tracking in anonymous location datasets. It should be analyzed if these methods can be transformed to our concept, especially how our concept can be combined with ID changing algorithms that do not change IDs at fixed frequencies but change IDs based on their context.

7 Conclusion

We presented a new approach to protect the location privacy of nodes in a MANET, while simultaneously still enable a working location service and therefore geo-unicast communication. Our mechanism allows the user fine-grained control of the location data he discloses to which other parties, expressed by individual trust relationships. The user can also control when and for which time span the information is disclosed. Additionally the user can at anytime revert past decisions about disclosing his location information. Compared to using random IDs our approach does not pose new loads onto the network or location service and only require little new requirements for nodes computing and storage capacities.
Summarized our approach provides a valuable extension to location privacy protecting MANETs, while re-enabling certain functions lost through previous, simple protection approaches. Our approach effectively combines technical and sociological requirements for location privacy aware mobile ad hoc networks.

References

[1] United Nations, Universal Declaration of Human Rights, General Assembly Resolution 217 A (III), 1948.

[2] Random ids for preserving location privacy. In *SecureComm 2005*, 2005.

[3] A. Beresford and F. Stajano. Mix zones: User privacy in location-aware services. In *IEEE International Workshop on Pervasive Computing and CommunicationSecurity (PerSec)*, March 2004.

[4] B. Bochow, F. Doetzer, A. Festag, M. Gerlach, T. Leinmueller, and R. Kroh. Attacks on inter vehicle communication systems - an analysis. In *International Workshop on Intelligent Transportation*, 2006.

[5] Bundesministerium des Innern. Bundesdatenschutzgesetz, 1990.

[6] S. Consolvo, I. E. Smith, T. Matthews, A. LaMarca, J. Tabert, and P. Powledge. Location disclosure to social relations: why, when, & what people want to share. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 81–90, New York, NY, USA, 2005. ACM Press.

[7] F. Doetzer. Privacy issues in vehicular ad hoc networks. In *Workshop on Privacy Enhancing Technologies*, Cavtat, Croatia, May 2005.

[8] S. Giordano, I. Stojmenovic, and L. Blazevie. Position based routing algorithms for ad hoc networks: a taxonomy. Juli 2001.

[9] M. Goettle. Location Privacy in ubiquitaeren Systemen. Master's thesis, University of Ulm, 2006.

[10] M. Gruteser and B. Hoh. On the anonymity of periodic location samples. In *SPC*, pages 179–192, 2005.

[11] J. Hall, M. Barbeau, and E. Kranakis. Enhancing intrusion detection in wireless networks using radio frequency fingerprinting.

[12] J.-P. Hubaux, S. Capkun, and L. Jun. The Security and Privacy of Smart Vehicles. *IEEE Security and Privacy*, 4:49–55, 2004.

[13] M. Kaesemann, H. Fuessler, H. Hartenstein, and M. Mauve. A Reactive Location Service for Mobile Ad Hoc Networks. Technical Report TR-14-2002, Department of Computer Science, University of Mannheim, Mannheim, Germany, 2002.

[14] W. Kiess, H. Fuessler, J. Widmer, and M. Mauve. Hierarchical Location Service for Mobile Ad-Hoc Networks. In *ACM SIGMOBILE Mobile Computing and Communications Review*, pages 47–58, 2004.

[15] A. Pfitzmann and M. Koehntopp. Anonymity, unobservability, and pseudonymity - a proposal for terminology. In *Proceedings of Workshop on Design Issues in Anonymity and Unobservability*, page 1?9. Springer, 2001.

[16] M. Raya and J.-P. Hubaux. The security of vehicular ad hoc networks. In *SASN '05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, pages 11–21, New York, NY, USA, 2005. ACM Press.

[17] D. Samfat, R. Molva, and N. Asokan. Untraceability in mobile networks. In *MobiCom '95: Proceedings of the 1st annual international conference on Mobile computing and networking*, pages 26–36, New York, NY, USA, 1995. ACM Press.

[18] E. Schoch, F. Kargl, S. Schlott, T. Leinmüller, and P. Papadimitratos. Impact of pseudonym changes on geographic routing in vanets. In *Third European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS 2006)*, Hamburg, Germany, Sept. 2006.

[19] C. Shete, S. Sawhney, S. Hervvadkar, V. Mehandru, and A. Helmy. Analysis of the effects of mobility on the grid location service in ad hoc networks. In *Communications, 2004 IEEE International Conference on*, volume 7, pages 4341–4345 Vol.7, 2004.