Hide-and-Lie: Enhancing Application-level Privacy in Opportunistic Networks

László Dóra, and Tamás Holczer
Laboratory of Cryptography and Systems Security (CrySyS)
Budapest University of Economics and Technology
{dora, holczer}@crysys.hu

ABSTRACT

A delay-tolerant network is a mobile ad hoc network where the message dissemination is based on the store-carry-and-forward principle. This principle raises new aspects of the privacy problem. In particular, an attacker can build a user profile and trace the nodes based on this profile even if the message exchange protocol provides anonymity. In this paper, an attacker model is presented and some proposed attackers are implemented. We analyze the efficiency of both the attacks and the proposed defense mechanism, called Hide-and-Lie Strategy. We show that without any defense mechanism, the nodes are traceable, but with the Hide-and-Lie Strategy, the success probability of an attacker can be made equal to the success probability of the simple guessing. Furthermore, in some scenarios, the Hide-and-Lie Strategy increases the message delivery ratio. The number of downloaded messages and the maximal memory size required to apply the proposed privacy defense mechanism is also investigated.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Store and forward networks*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

General Terms

Security, Performance

Keywords

Opportunistic networks, Privacy in data forwarding

1. INTRODUCTION

A delay-tolerant network (DTN) is a mobile ad hoc network. In DTNs, the messages are disseminated according to the *store-carry-and-forward* principle. In particular, the messages are stored and carried by the mobile devices and forwarded to intermediate devices when they have connection (i.e., when they are in vicinity of each other). DTNs can be used to deliver information either in a target centric manner, or in a data centric manner. In the target centric manner, the recipient of the data is known, and the task is to deliver the data to that user. In the data centric manner, only the data is known, and the recipient can be anyone, who is interested in that particular data. The task here is to deliver the information to as many interested users as possible. In this paper, we consider the application of DTNs to inter-personal wireless communication. In these applications, local information needs to be distributed to a set of nearby destinations based on their interest in the information (data centric application).

As a motivating example, let us consider a small advertisement application. Using this application, a diligent student participating in each lecture can earn some money. He types an advertisement about helping other students. There is a group of students who participate in some lectures meeting the diligent student. Their handheld devices can download the advertisement, because they are interested in any messages related to the university. They go back to the dormitory and meet students who do not visit the lectures at all. Lazy student's handheld device downloads the advertisement and before the exam they can contact the diligent student.

Without privacy protection, no new technology should spread widely. The privacy of the users must be ensured in DTNs as well. Some of the problems can be mitigated by traditional technologies, but some new problems are introduced by the store-carry-and-forward manner of the DTNs.

The privacy problem can be clarified through the above described example. Let us consider a lecturer who wants to make a list of the students attending the lectures without the consent of the students. This list can be easily constructed if the ownership between the students and their handheld devices are disclosed once, and their handheld devices are traceable.

The privacy of a system and the anonymity of the users can be a problem on different levels of the communication stack. Using a rough partitioning, the problem can be related to the physical layer of the network, the network and transport layers, and the application layer. In the physical layer, the physical characteristics of the transceiver, in the application layer, the application data, and in any layers, especially in the network and transport layers, the identifiers must be hidden or anonymized.

The physical layer privacy problems are also referred to

MobiOpp '10, February 22-23, 2010, Pisa, Italy.

remote device fingerprinting or remote device identification. In [9], the authors can fingerprint a device remotely without any modification on the target machine, from any distance, measuring only the clock skew of the target machine. This technique can be very accurate, but needs very long interaction time. In [1, 13], very accurate hardware fingerprint based device identification methods are given which can identify the devices very precisely, but utilizes special expensive hardware equipments.

In the network and transport layer, the anonym emailing and anonym web browsing are well studied problems (for a comprehensive survey, see [3]). A more similar problem to the DTN's arises in Vehicular Ad-hoc Networks (VANET). In various aspects, the VANETs are similar to DTNs. They are mobile, the transaction times are short, and the devices belong to a well defined person. However, in VANETs, some infrastructure elements can be assumed in contrast to DTNs. Nevertheless, the solutions proposed for privacy issues can be a good inspiration for solving the privacy problems in DTNs. Many authors addressed the privacy problem in VANETs, (e.g., in [4, 12]). In VANETs, the messages contain state dependent information (e.g. speed, location, time) and contain no personal information. Thus, the most beneficial attack is the tracing of identities.

The problem of privacy in opportunistic networks is considered in some papers. In [10], the authors raise the problem of data privacy when a node sends sensitive information to another node and it does not want it to be available for intermediate nodes. The author of [6] proposes a pseudonym generating technique using public keys for supporting the anonymous communication. In [8], an anonymous communication solution is presented and a new anonymous authentication protocol is introduced for DTNs. The solution is based on identity-based cryptography and solves the problem of anonymous communication on the network level, but the application level problems are not handled.

A relevant, but somewhat different application level privacy problem can be found in [11], where a re-identification algorithm is given targeting anonymized social network graphs. The de-anonymization algorithm is based purely on the network topology. This mechanism can be hardly adapt into DTNs, because this algorithm only relies on the static social connections of the nodes.

Privacy preserving data mining is also a relevant topic. In [14] the categorization of different privacy preserving techniques is given. The problem of deriving private information from randomized data is analyzed in [7]. The problem is similar to ours, but in that paper, the attackers goal is to reveal the hidden data itself, while in our paper, the owner of the data is the hidden information. Another related problem in [5] is how to randomize some data, while keeping the statistics close to the original. The main difference is the same as the previous paper.

It is essential that the communication is anonymous. Anonymity (or at least pseudonymity) can be easily achieved by the usage of pseudonyms (i.e., temporal identifiers). A more serious and DTN specific application level privacy problem is that the nodes can be identified by their stored messages. If an attacker is able to build a user profile using the exchanged application data, the user becomes traceable even if the communication is completely anonymous. Therefore, new mechanism or adaptation of proposed mechanism is required in DTNs to ensure untraceability of the nodes, namely, to avoid building traceable user profiles.

The main contributions of our paper are the following: We are the first who raise the problem of application level privacy in Delay Tolerant Networks. We characterize and simulate efficient software based attackers that can link different appearances of the same node with high probability using only regular handheld devices. We suggest and evaluate an efficient defense strategy, too, which is useful against this attacker without jeopardizing the node's main goal, the message collection.

After reviewing the state-of-the-art and focusing on the problem of the application level privacy in DTNs, the remainder of the paper is organized as follows. In Section 2, we describe the system model. The attacker model is presented and four different attackers are defined in Section 3. In Section 4, we describe our proposed privacy enhancing technique, called Hide-and-Lie Strategy. The simulation environment is defined in Section 5. The efficiency of different attackers and the privacy enhancing technique is exhaustively analyzed by means of simulations in Section 6. Finally, we conclude our paper in Section 7.

2. SYSTEM MODEL

In our model, the users are placed on a field of arbitrary shape. They own devices which can communicate with each other within their radio range. The used wireless technology can be Bluetooth, Wi-fi, or any suitable wireless technique. The messages are generated and disseminated among the devices/users but each user is only interested in a subset of messages. The dissemination process is based on the store-carry-and-forward principle. A node includes a user (the owner of the device) and a device. We assume that the data dissemination has no impact on the user's movement.

The communication between the nodes is assumed to be anonymous. Hence, an attacker is not able to trace a node by e.g. simply tracing a network identifier.

The messages are generated by special nodes, called message generator nodes. In our system model, the time is slotted, and each message generator node generates a new message with a fixed average rate: ϱ messages per time step. Each of the message generator nodes stores only the most recently generated message. This message can be downloaded by any node that passes by the message generator node.

We assume that a mechanism can filter out the fake messages from the network. This assumption is necessary, otherwise, an attacker is able to create a special decoy message and trace a node by following it. The analysis of the effectiveness of this kind of attacks and countermeasures is out of the scope of this paper, but considered as future work.

For the sake of simplicity, it is assumed that there are C categories, and each message belongs to a single category. When a message generator node generates a message, it specifies which category the new message belongs to. Each message is classified into a category uniformly at random. Therefore, a new message belongs to a specific category with probability $\frac{1}{C}$.

An interest profile (IP) of a node, which is a part of the user profile, is a binary vector representing a list of categories the node is interested in. A message belonging to category k is called primary for a node if $IP[k] = 1$, the message is secondary otherwise. A node is interested in any given category (i.e. all the messages belonging to that category are primary for the node) with probability ε. As the participating nodes are interested in at least one category, the case when $\varepsilon = 0$ can be excluded, therefore, the cases when $0 < \varepsilon \leq 1$ are considered. For the sake of simplicity, ε is equal for each node in each considered scenarios.

Each message is assumed to have a unique identifier and a node can decide based on this identifier if a message M is stored in its memory before downloading from another node.

According to our assumptions, a message M has the following formula:

$$M = [ID|CAT|data] \tag{1}$$

where the ID is the unique identifier of the message, CAT is the identifier of the category which the message belongs to, and $data$ is the content of the message. The length of the $data$ may be some magnitude higher than the length of the ID and the CAT.

When two nodes get in the vicinity of each other, they start to exchange messages. Each node u wants to download those messages from the other participant that u does not store and fit to its interest profile.

The whole message exchange may be not completed because the nodes are mobile and they may leave the radio range of the other participant before exchanging all the required messages. Therefore, according to the system model, the nodes are not able to obtain as many messages as they want but at maximum one for each participant per time step. It is assumed that a message is downloaded without interruption in a time step.

In our imagined scenario, the batteries of the handheld devices can be easily recharged day by day, hence, the cost related to the battery consumption due to communications is negligible. However, the average number of downloaded messages per node is investigated in the simulations to get an insight of the message exchange rate.

The storage cost has two aspects: 1) The messages need storage space and storage constraints may limit the number of stored messages. No explicit limitation for the storage space is defined, however, the maximum quantity of the stored messages in the devices is investigated. 2) The time needed to determine which messages the nodes want to download increases polynomially with the number of messages stored by the other participant. Therefore, the increasing number of messages is controlled by deleting the messages of the system. Each message is deleted D time steps after its generation. D should be sufficiently large, as the network itself is delay tolerant.

In order to investigate the effect of the defense mechanism on the message delivery ratio, a formula is defined for the gain of the nodes. Until time step t, a node u obtains $O_u(t)$ number of primary messages while in the system, there have been $A_u(t)$ number of messages generated which is primary for u. The gain $(G_u(t))$ of node u is the ratio of these values: $G_u(t) = \frac{O_u(t)}{A_u(t)}$. The fact that the gain reaches its steady state value is proven in [2]. Therefore, in what follows, we approximate the steady state value of the gain by considering the gain after sufficiently large simulation time and we denote it by G_u.

From the privacy point of view, an extreme approach consists in denying the participation in the network. We assume that the nodes want to obtain messages while they want to preserve their privacy, too. A beneficial and in the same time selfish behavior would be to download any kind of messages but not forwarding them. To prevent this kind of selfish behavior, we assume that a mechanism encourages the message dissemination among the nodes [2]. Therefore, the nodes themselves want to increase the number of offered messages.

3. ATTACKER MODEL

The attacker wants to track the target node to breach its privacy. To do so, it tries to link the profiles acquired in different times together. If the profiles can be linked correctly, the attack against the privacy is successful.

In this section, we describe what information an attacker can get from the nodes, how he can obtain this information and how he can link the nodes.

3.1 Leaking information

The communication between the nodes can leak some information about the interests of the participants. In this paper, attacks based on these leaked information are considered. We assume that the attacker can estimate the following user profile (UP) from a node u at time t:

$$UP_u(t) = (EIP_u(t), CHM_u(t), IDL_u(t)) \tag{2}$$

The UP consists of the following triple: Estimated Interest Profile (EIP) is a binary vector. The value of the vector at the k^{th} position equals to 1 if category k seems to be interesting for node u. Category Histogram of offered Messages (CHM) shows, for each category, how many messages in the ID list belong to that category. IDL is the ID list of offered messages.

In this paper, we abstract away what message exchange protocol is used, we only assume that an attacker can obtain the $UP_u(t)$ triplets for each node u in time step t.

3.2 Attacker behavior

The attacker, in our model, behaves according to the following attacker model:

1) The attacker identifies its target node (u_T) from N nodes.

2) The attacker reads the current user profile of the target: $UP_{u_T}(t_0)$. The time step when this happens is considered as a reference time, i.e. t_0.

3) τ time later ($t_1 = t_0 + \tau$), the attacker reads $UP_{u_i}(t_1), i \in [1..N]$ of each node and calculates a metric how similar is u_i to u_T is. τ is also referred as the attacker delay. To mislead the attacker, the nodes can slightly modify their UPs. The UP perturbation is defined in Section 4.

4) The attacker chooses the node most similar to the target node. If more than one have the maximal similarity value, it chooses randomly between them. If the chosen node is u_T, the attacker is successful.

We have chosen for the analysis the success probability of the attacker as the privacy metric, because it is widely used and tells the most about the expected outcome of the attack. In the cryptographic literature, a widely used metric is the indistinguishability of the target from a randomly chosen node This metric differs from ours slightly as the attacker wants to distinguish the target from every other node. Our extended metric can be imagined as the conventional metric used N times one after the other. More precisely, if the attacker can recognize its target from two nodes with probability p, then it can recognize it from N

nodes with probability p^{N-1}, if the nodes are independent. The conventional model is more sensitive for p close to 0.5. In contrast, the extended model is more informative for p close to 1. As the results show, p can be close to 1 when no defense mechanism is used, so the extended model is used.

To fully define the attacker, a similarity metric must be defined. Some possible and useful similarity functions are defined in the next section.

3.3 Attacker functions

The attacker can define the similarity of the target and a suspected node based on the $UP_u(t)$. Using the user profiles of the nodes, the attacker can calculate the similarity using an attacker function $\mathcal{A}$.

More formally the input of $\mathcal{A}$ are $N+1$ user profiles, and the output is an ID of a node:

$$\mathcal{A} : (UP_{u_T}(t_0), UP_{u_i}(t_1), i \in [1..N]) \rightarrow j, j \in [1..N] \quad (3)$$

The attack is successful if and only if $j = T$.

It is clear that any attacker can reach a minimal value of the success probability $\frac{1}{N}$ by simple guessing. Higher values can also be achieved using more sophisticated attacker functions. In the following, four different simple attacker functions are defined.

Prefiltered ID Based attacker function assumes that nodes show their real interest profiles. The attacker can filter out every suspect who has different EIPs, considering only the nodes whose $EIP_u(t_1)$ equals to $EIP_{u_T}(t_0)$. From the remaining set, it selects the one whose $IDL_u(t_1)$ is the most similar to $IDL_{u_T}(t_0)$. Under similarity, the cardinality of the intersection of the target ID list and the suspect's ID list is meant. If the remaining set is empty, the attacker selects the target by pure guessing. The intuition behind this attacker is that after some time the target can get some new messages and delete some old ones, but mainly its ID list is unchanged. This attacker can be very efficient if the nodes show their real IPs which means that EIPs are not changed over time, but can be very inefficient if the EIPs are changed.

Unfiltered ID Based attacker function is a simplified version of the previous function, as it uses only the cardinality of the intersection of $IDL_{u_T}(t_0)$ and $IDL_u(t_1)$, but it does not prefilter the nodes by their EIP. This attacker is not so efficient in case of time invariant EIPs, but less sensitive for changing EIPs.

Category Histogram Based attacker function selects the node u whose $CHM_u(t_1)$ is the most similar to the $CHM_{u_T}(t_0)$. The similarity of two histograms is calculated using the χ^2–test. The intuition behind this attacker function is that a node can show a modified EIP but the histogram represents its real interest profile if the node collects messages according to its real interests.

Significant Category Based attacker function is the most complex function analyzed in this paper. It assumes that the interested categories are overrepresented in the ID list and the uninterested categories are underrepresented. This categorization only depends on the real IP of the target, and is hard to influence without totally changing the IP. To find the interested categories, the C categories must be classified into two clusters: the significant categories, and the remaining categories. This task can be easily done using the k-means clustering algorithm on the CHMs. The result of the clustering is a binary vector of length C with ones at the significant categories. The similarity of two binary vectors is defined as the Hamming distance of the vectors.

The properties and efficiency of the different attacker functions are analyzed in Section 6.

4. DEFENSE MECHANISM

In order to preserve a node's privacy, the User Profile (UP) should be obfuscated to distract the attacker. Against an eavesdropping attacker, another solution is to design the message exchange protocol in a way that it ensures that no sensitive information can leak during the communication. We are focusing on developing an obfuscation based mechanism because that can be used if the attacker actively takes part in the communication.

The more $UP_u(t_1)$ is different from $UP_u(t_0)$, the less likely the attacker can link the two profiles. The continuously changing profile hardens the task of the attacker, however, it may thwart the node from collecting primary messages. This is thoroughly analyzed in Section 6.

Two simple methods can be used to modify the UP through modifying the Interest Profile (IP) of the node. The first one is to *hide* some interesting categories, and claim them as uninteresting. The second one is to *lie* about some uninteresting categories, and claim them as interesting. These techniques can be used at the same time, this is what we call *Hide-and-Lie Strategy* (HLS). The temporarily obfuscated IP is the Temporal Interest Profile or the EIP from the attacker point of view. The EIP can be transient, which means that a new EIP can be generated by every node in every time step.

Obviously, the required and offered messages during the message exchange must be synchronized with the EIP: 1) messages relating to hidden categories must be hidden as well, and 2) when a node lies about being interested in a given category, it collects and offers messages belonging to that uninteresting category.

Many different HLSs can be envisioned. Different strategies can hide or lie about different categories in different situations. In the following, a simple but rather general solution is given: every node generates its EIP from its IP by inverting every category in the IP with a given probability λ. Inverting means indicating an uninteresting category as interesting or vice versa. This parameter λ is the Hide-and-Lie strategy value.

As λ is a probability, it is between 0 and 1. The nodes which do not use any obfuscation techniques can be modeled as nodes using HLS with $\lambda = 0$ as the node never modifies its EIP.

The other interesting value of λ is 0.5. It totally randomizes the IP, making the EIP a uniformly distributed random binary vector. It is the best strategy for the privacy sensitive users. For demonstrating this, let us assume two nodes, u_1 and u_2 at time τ of the attack. They show temporal interests in every category with probability 0.5, thus, their interest profiles are independent from their real interest profiles. If τ is greater than the D message expiration time, then none of them has any messages from the reference time of the attack (t_0). On average, every user shows interest in a given category in every second round (as $\lambda = 0.5$), so every message is collected by every node with the same probability. The CHMs of the nodes are close to the uniform distribution considering those categories where the EIP shows that the node is interested in (other categories are represented by

0 messages). The reason is that every message is generated and collected with the same probability. Therefore, u_1 and u_2 show user profiles, which are independent from the user and statistically the same.

The used HLS transformation of the users' profiles generates every possible UP with the same probability, thus, no statistical test can distinguish between u_1's and u_2's UP.

Values of λ greater than 0.5 are useless for the nodes, as they make the EIP as traceable as the inverse EIP with $\lambda' = 1-\lambda$, but the nodes collect more uninteresting messages than interesting ones. Consequently, in the following, only $0 \leq \lambda \leq 0.5$ are considered.

5. SIMULATION

In the simulations implemented in C++, the fixed-number of mobile nodes move in discrete time steps according to one of the two mobility models: the random walk (RW) and restricted random waypoint (RRW) model.

In the RW model, 300 nodes move on a grid of size 15×15. In each time step, a node can move to one of the four neighboring grid points (in what follows, these are called meeting points), or stay at the current place. The probability of each of these actions is 0.2. In each time step, the nodes that happen to be at the same meeting point are paired randomly and each pair executes the message exchange protocol. These pairs are able to download one message from each other as described in Section 2.

In the RRW model, 300 nodes (initially placed uniformly at random) move on a field of size 20×20 unit. On the field, there are some special points chosen at random; these are called meeting points. Each node selects a meeting point randomly, and moves towards this meeting point along a straight line with a fixed speed. When the meeting point is reached, the mobile node stops and stays there for randomly chosen time (10 time steps on average). Then, it chooses another meeting point and begins to move again. The nodes that happen to be at the same meeting point in the same time step are paired randomly and these pairs are able to download one message from each other as described in Section 2.

In the case of RW, 30 message generator nodes are placed on the subset of all meeting points uniformly at random. In the case of RRW, one message generator node is placed on each meeting point, and the number of meeting points is 30. All the 30 message generator nodes together generate one new message per time step on average both in case of RW and RRW mobility model.

The parameters of the mobility models were determined such that the number of message exchanges are equal on average.

The length (number of time steps) of the simulation was determined in an empirical way by taking into account that the gain has to reach its steady-state value. In the beginning of the simulation, the nodes do not store any messages. Therefore, their gains are volatile in the first time steps. When the simulations were run for 3000 time steps, the average gain has not changed considerably for upcoming 1000 time steps in the analyzed simulations. Therefore, the attacker started its attack after 3000 time step long bootstrap ($t_0 = 3000$) and the simulator was run for additional 1000 upcoming time steps to investigate the effectiveness of the attacker for different τ values.

Some of the parameters that describe our envisioned system were fixed in order to reduce the number of simulation scenarios and other ones which have the highest effect on the success probability of the attacker were varied. The fixed simulation parameters are summarized including the mobility model specific ones in Table 1.

Table 1: Fixed simulation parameters

Parameter	RRW	RW
Simulation length in time steps	4000	
Number of nodes (N)	300	
Number of message generator nodes	30	
Message generation rate (ϱ)	0.0333	
Simulation area (unit)	20×20	15×15
Number of meeting points	30	225
Probability of leaving a meeting point	0.1	0.8
Velocity	1 unit/time step	
Lifetime of messages (D)	500	

The simulation parameters that are related to the interest profile were varied: number of message categories (C) and probability of being interested in a category (ε) as these parameters affect most the success probability of the attack.

Parameter C should be higher than 1, otherwise all the nodes have the same IP. Therefore, the chosen lowest value is 2. We think that 50 categories is high enough, because a higher value would not affect the simulation results considerably (the results for 30 and 50 are similar). The considered values are 2, 5, 10, 30, 50.

To reduce the complexity of the simulations, we selected some $0 < \varepsilon \leq 1$ values in a way that instead of $\varepsilon = 0$ we included $\varepsilon = 0.05$, and we also investigated a special case, $\varepsilon = 0.5$. The considered values are 0.05, 0.2, 0.4, 0.5, 0.6, 0.8, 1. Note that with probability $(1-\varepsilon)^C$, the simulator generates such an interest profile that the node is not interested in any category. In that case, a new IP is generated.

Recall that a node can choose a Hide-and-Lie strategy value from the interval $0 \leq \lambda \leq 0.5$. For the sake of simplicity, only those cases are considered when each node chooses the same λ value from the following set: $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

We also investigate how the time elapsed between t_0 and t_1 (i.e., τ) affects the success probability of the attack. We considered the following values of τ: $\{1, 50, 250, 500, 1000\}$. In the special case when $\tau = 1$, the ID list of stored messages does not change considerably, however, the Hide-and-Lie Strategy affects the UP. Recall that the messages are deleted from the system after 500 time steps. Therefore, when $\tau > 500$, no message will match to the target node's ID list in t_0.

Table 2: Varied simulation parameters

Parameter	Possible values
Number of categories (C)	2, 5, 10, 30, 50
Probability that a node is interested in a category (ε)	0.05, 0.2, 0.4, 0.5, 0.6, 0.8, 1
Hide-and-Lie strategy value (λ)	0, 0.1, 0.2, 0.3, 0.4, 0.5
Attacker delay (τ)	1, 50, 250, 500, 1000

The main objectives are to investigate the success probability of the attacks and the efficiency of the HLS. The analysis was performed in each combination of the parameter values summed up in Table 2. In one parameter set, the success probability of every attacker function was calculated using the following method: A different simulation run for each parameter set was executed. In each execution, the attacker selects each node as a target node one-by-one at time t_0 and performs the attacker function with the target node and all the nodes as the input of the function at each $t_1 = t_0 + \tau$ time. The success probability is the ratio of the successful attacks.

6. RESULTS

In this section, two representative scenarios (see Table 3) are exhaustively analyzed. In particular, the efficiency of different attacker functions presented in Section 3.3 and the efficiency of the defense mechanism presented in Section 4 are investigated. Beyond the analysis of two emphasized scenarios, we show the differences compared to the other simulated scenarios. In the two considered scenarios, we investigate the effect of the Hide-and-Lie Strategy on the reached gain and the number of downloaded primary and secondary messages and the maximum memory required to follow the proposed Hide-and-Lie strategy.

As our experience showed that the chosen mobility model does not affect the results considerably, we have selected the random walk mobility model in the analyzed scenarios. Because of the space limits, we emphasize rather the effect of the probability of being interested in a category instead of the number of categories. Therefore, in the presented simulation results, the number of categories is fixed to 30, which can be a realistic value for a lot of applications. In the two investigated scenarios, the probability of being interested in a category takes the values 0.05 and 0.4. The former value refers to those scenarios where the nodes are interested in a small subset of messages, like in the example scenario presented in Section 1, while in the latter scenario, the nodes are interested in a large subset of messages. In Table 3, we summarize the parameter values of the scenarios beyond the already fixed parameters introduced in Table 1.

Table 3: Parameter values of investigated scenarios

	Mobility model	**C**	ε
Scenario 1	RW	30	0.05
Scenario 2			0.4

The success probability of the attacker functions is plotted against different Hide-and-Lie strategy values (λ) and different attacker delay (τ) values of Scenario 1 and 2 in Figure 1(a) and 1(b), respectively. For the sake of better understanding, the plots are separated by different attacker delay values.

The Prefiltered ID Based attacker function assumes that the nodes do not apply any privacy enhancing technique. According to this, it is the most efficient attacker function when $\lambda = 0$, but in any other cases, the attacker function can not distinguish the target node from the others, because even one entry changing in the EIP misleads the attacker.

A more robust solution can be obtained by omitting the prefiltering which results in the Unfiltered ID Based attacker function. The success probability of this function decreases when $\lambda = 0$ compared to the prefiltered function but considerably increases in other cases. The reason is that the number of all the combinations of the messages give enough variety to the attacker to identify the nodes with higher probability even if they hide a small subset of the messages when they meet other nodes. As the nodes increase the λ value, they collect messages from larger sets and they can hide more messages. Hence, the nodes are able to deceive the attacker with high probability. Therefore, the success probability of the attacker function decreases with the increasing λ value. If $\lambda = 0.5$, the attacker function is as inefficient as a naïve attacker.

The Unfiltered ID Based attacker function is very sensitive for the attacker delay. As τ increases the nodes delete more and more messages making the attack less and less efficient. Finally, the nodes delete all the messages that could match the $IDL_{u_T}(t_0)$ after D time steps and this attack becomes inefficient in cases where $\tau = 500$ or $\tau = 1000$. Recall that $D = 500$ in the considered scenarios.

The Category Histogram Based attacker function is less sensitive to the τ value, but it is less efficient when τ is lower than the ID Based attacker function. The inefficiency of this attacker function comes from the fact that the Hide-and-Lie Strategy causes intolerable differences for the χ^2–test when all the messages appear or disappear belonging to a category when EIP changes.

The attack that is least sensitive to τ is the Significant Category Based attacker function. The advantageous characteristic comes from the fact that this function tries to reveal the real interest profile. However, it still does not work when the nodes hide their identity with $\lambda = 0.5$ strategy, because there are no over- and underrepresented categories in that case.

The Significant Category Based attacker function is the most efficient attacker function in Scenario 2, but it is less efficient in Scenario 1.

Taking all the considered attacker functions into consideration, we can conclude that the efficiency of the attacker functions changes according to the parameters of the model. However, a common tendency is that if the nodes apply the Hide-and-Lie Strategy with high value of λ, none of the attackers is able to distinguish them better, independently of the value of τ, than a naïve attacker which picks up one of the nodes by random.

Even if an attacker can distinguish two nodes if their IPs are different (we call this attacker ideal IP based attacker $\mathcal{A}_{IP\ \mathrm{ideal}}$), the probability that two nodes have the same IP is not negligible. The success probability of an ideal IP based attacker can be viewed as an upperbound for any other IP based attacker, such as, e.g. the Significant Category Based attacker function. This value can be determined analytically. Through this analysis, we show how different C and ε values affect the success probability of the attackers.

The success probability of the ideal IP based attacker is determined by the number of equal IPs. To compute the success probability, first the probability p of two IPs being equal is computed as follows:

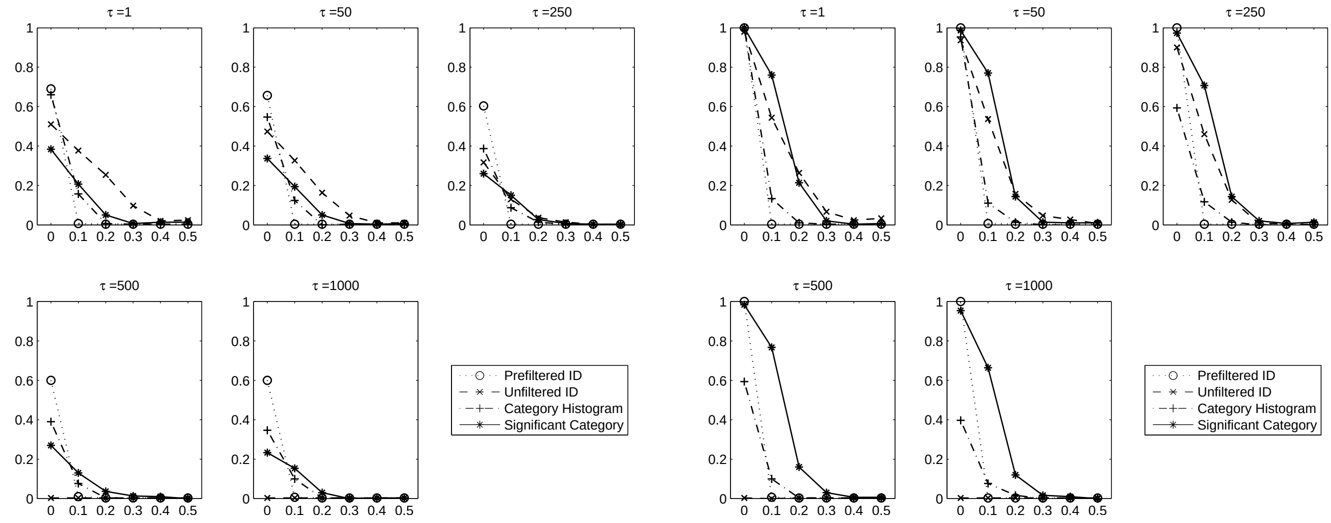
$$\begin{aligned} p &= \frac{\sum_{w=1}^{C}\binom{C}{w}\left(\varepsilon^2\right)^w\left((1-\varepsilon)^2\right)^{C-w}}{\left(1-(1-\varepsilon)^C\right)^2} \\ &= \frac{\left(\varepsilon^2+(1-\varepsilon)^2\right)^C-(1-\varepsilon)^{2C}}{\left(1-(1-\varepsilon)^C\right)^2} \end{aligned} \tag{4}$$

where w is the weight of the IP varying between 1 and C (at least every node is interested in one category).

The success probability of $\mathcal{A}_{IP\ \mathrm{ideal}}$ is the reciprocal of the average number of nodes with the same IP:

$$\Pr(\mathcal{A}_{IP\ \mathrm{ideal}}(UP_{u_T}(t_0), UP_{u_1}(t_1), \ldots, UP_{u_N}(t_1)) = u_T)$$

$$\simeq \frac{1}{1+p\,(N-1)} \tag{5}$$



(a) Scenario 1

(b) Scenario 2

Figure 1: Success probability of $\mathcal{A}$ as a function of the Hide-and-Lie strategy values (λ)

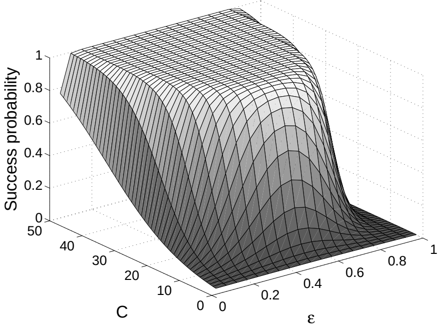


Figure 2: Analytically determined success probability of an ideal IP based attacker functions when 300 nodes are present in the network

The ideal values according to Eq. (5) are 0.341 and 1 for Scenario 1 and 2, respectively. These values are valid only for $\lambda = 0$, and confirmed by Figure 1. These values are shown in Figure 2, too, where Eq. (5) is plotted against different C and ε values.

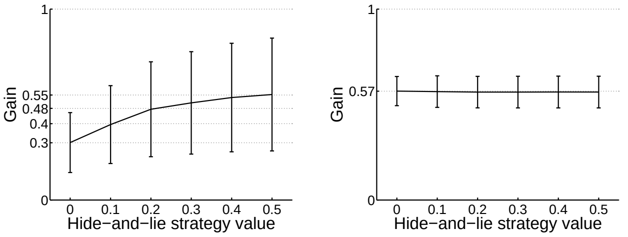
The characteristic of the success probability of the attacker in the case of the two emphasized scenarios are similar to each other as Figures 1(a) and 1(b) show and these are similar to the other scenarios which are simulated but not presented here. However, as Figure 2 shows, the success probability of the ideal IP based attacker depends on the parameter value of the number of categories and the probability of a node being interested in a category. As one can read from the figure, when there are large number of categories in the system, the success probability of an ideal attacker is high. On the other hand, when the number of the categories is low, the success probability highly depends on the value of ε. As the value ε gets closer to 0.5, the success probability increases. The reason is that an attacker can distinguish nodes when the probability that the IPs of two nodes are equal is low. All these statements are confirmed by the simulation results that are not presented here, and these effects can be observed even in cases when $\lambda > 0$.

In Figure 3, we show the average gain of all the nodes as a function of the Hide-and-Lie strategy in the two scenarios and its empirical standard deviation. We have to stress that these two figures do not represent all the appeared characteristic of the figures, however, Figure 3(a) shows an interesting property of the Hide-and-Lie Strategy. Namely, increasing λ does not degrade but increases the data delivery ratio in some scenarios.

The Hide-and-Lie Strategy has two contradictory effects: On the one hand, when the nodes happen to hide what they are interested in, they may miss some primary messages to download. On the other hand, when the nodes happen to lie being interested in some category, they store-carry-and-forward secondary messages, which increases the data delivery ratio in general [2]. The cumulative effect depends on the system parameters. E.g. in a case when nodes are interested only in a small subset of categories and they do not carry secondary messages, they can exchange messages only with small probability. Therefore, the Hide-and-Lie Strategy in some cases can be viewed as a motivation to store-carry-and-forward secondary messages as it can be seen in Figure 3(a). On the other hand, when the nodes have many possibilities to get primary messages, the latter effect has no considerable benefit while the former effect degrades the gain. Surprisingly, the two effects are balanced in Scenario 2 as one can see in Figure 3(b).

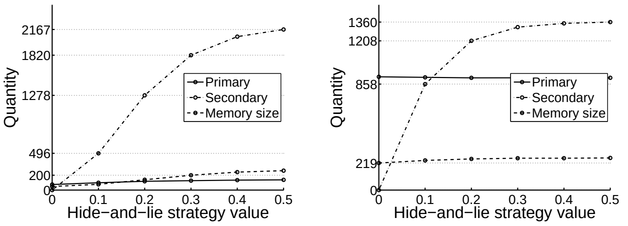
Even though we did not take into consideration the energy consumption and the memory costs of the communication when we calculated the gain, we collected related information during the simulation. We plotted the average number of primary and secondary messages downloaded by one node and maximum memory usage as a function of the Hide-and-Lie strategy in the two considered scenarios in Figure 4.

As one can expect, the number of the downloaded secondary messages increases with increasing λ value. The number of the downloaded primary messages changes as the gain changes because the gain is a normalized value of the



(a) Scenario 1 (b) Scenario 2

Figure 3: Average gain with the empirical standard deviation



(a) Scenario 1 (b) Scenario 2

Figure 4: Costs (Average number of primary and secondary messages downloaded by a node and the maximum memory usage)

number of obtained primary messages.

Even though the gains are comparable in the two scenarios as Figure 3 shows, there is almost one order of magnitude difference in the number of downloaded primary messages. The reason is that in Scenario 1, the nodes are interested in 5% of the messages and in Scenario 2, the nodes are interested in 40% of the messages while the number of the generated messages does not change considerably in the two scenarios. Due to the same reason, the number of the secondary messages for a node is less in Scenario 2 than in Scenario 1. The ratio of the number of downloaded primary and the number of secondary messages is $\varepsilon(1-\lambda) : (1-\varepsilon)\lambda$.

Note that even if the nodes download more and more secondary messages as λ increases, the maximal memory usage does not increase at the same order. Thus, the nodes do not need to maintain much larger memories when they want to protect their privacy.

7. CONCLUSION

In this paper, the application level privacy in Delay Tolerant Networks has been investigated. In particular, an attacker can build a user profile of a node based on what messages the node stores and what messages it wants to download. After profiling, the attacker can trace the node based on the user profile even if the node communicates with the other nodes through anonymous links. A system and an attacker model was built and some attacker functions were proposed. A defense mechanism called Hide-and-Lie Strategy against such attacks was proposed, too. This mechanism has a free parameter with which the system can be tuned between high privacy level and low data-forwarding overload. In our model, we analyzed the efficiency both of the attacks at different parameter values and the proposed defense mechanism. We showed that without any defense mechanism, the nodes are traceable, but with the proposed Hide-and-Lie Strategy, the success probability of an attacker can be decreased substantially. The message delivery ratio and the costs at different Hide-and-Lie parameter values are also investigated. We found that in some scenarios, the Hide-and-Lie Strategy can be viewed as a motivation for other nodes to carry messages that they are not interested in. Therefore, as a positive side effect, the message delivery ratio is also increased.

As a future work, the possible effects of a more realistic mobility model can be analyzed, and the case of dependent categories can be examined. As mentioned in Sec. 2, the problem of decoy messages can also be investigated.

Acknowledgment. This work was supported by the European Commission in the context of the 6th Framework Programme through the BIONETS Project (www.bionets.eu). The authors are thankful to Levente Buttyán, László Czap, Péter Schaffer, and Daniel Schreckling for their useful comments.

8. REFERENCES

[1] V. Brik, S. Banerjee, M. Gruteser, and S. Oh. Wireless device identification with radiometric signatures. In *Proc. of the ACM MobiCom*, pages 116–127, 2008.

[2] L. Buttyán, L. Dóra, M. Félegyházi, and I. Vajda. Barter trade improves message delivery in opportunistic networks. *Elsevier Ad Hoc Networks*, 2009.

[3] G. Danezis and C. Diaz. A survey of anonymous communication channels. Technical Report MSR-TR-2008-35, Microsoft Research, January 2008.

[4] F. Dötzer. Privacy Issues in Vehicular Ad Hoc Networks. In *Proc. of the PET*, pages 197–209, 2005.

[5] W. Du and Z. Zhan. Using randomized response techniques for privacy-preserving data mining. In *Proc. of the 9th ACM SIGKDD*, pages 505–510, 2003.

[6] A. Heinemann. *Collaboration in Opportunistic Network*. PhD thesis, Fachbereich Informatik der Technischen Universität Darmstadt, 2007.

[7] Z. Huang, W. Du, and B. Chen. Deriving private information from randomized data. In *Proc. of the 2005 ACM SIGMOD*, pages 37–48. ACM, 2005.

[8] A. Kate, G. Zaverucha, and U. Hengartner. Anonymity and security in delay tolerant networks. In *Proc. of the 3rd SecureComm*, pages 504–513, 2007.

[9] T. Kohno, A. Broido, and K. Claffy. Remote physical device fingerprinting. *IEEE Transactions on Dependable and Secure Computing*, 2(2):93–108, 2005.

[10] L. Lilien, Z. Kamal, V. Bhuse, and A. Gupta. Opportunistic Networks: The Concept and Research Challenges in Privacy and Security. In *Proc. of the WSPWN*, pages 134–147, 2006.

[11] A. Narayanan and V. Shmatikov. De-anonymizing Social Networks. In *Proceedings of the 30th IEEE Symposium on Security and Privacy*, 2009.

[12] M. Raya and J.-P. Hubaux. Securing vehicular ad hoc networks. *Journal of Computer Security*, 15(1), 2007.

[13] S. Čapkun, R. Rengaswamy, I. Tsigkogiannis, and M. Srivastava. Implications of Radio Fingerprinting on the Security of Sensor Networks. In *Proc. of the 3rd SecureComm*, 2007.

[14] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis. State-of-the-art in privacy preserving data mining. *ACM SIGMOD Record*, 3(1):50–57, March 2004.