Privacy of Community Pseudonyms in Wireless Peer-to-Peer Networks

Julien Freudiger[†], Murtuza Jadliwala[†], Jean-Pierre Hubaux[†],
Valtteri Niemi[‡], Philip Ginzboorg[‡] and Imad Aad[‡]

[†] EPFL, Switzerland
firstname.lastname@epfl.ch
[‡] Nokia Research Center
firstname.lastname@nokia.com

ABSTRACT

Wireless networks offer novel means to enhance social interactions. In particular, peer-to-peer wireless communications enable direct and real-time interaction with nearby devices and communities and could extend current online social networks by providing complementary services including real-time friend and community detection and localized data sharing without infrastructure requirement. After years of research, the deployment of such peer-to-peer wireless networks is finally being considered. A fundamental primitive is the ability to discover geographic proximity of specific *communities* of people (e.g, friends or neighbors). To do so, mobile devices must exchange some community identifiers or messages. We investigate privacy threats introduced by such communications, in particular, adversarial community detection. We use the general concept of *community pseudonyms* to abstract *anonymous* community identification mechanisms and define two distinct notions of community privacy by using a *challenge-response* methodology. An extensive cost analysis and simulation results throw further light on the feasibility of these mechanisms in the upcoming generation of wireless peer-to-peer networks.

1. INTRODUCTION

Communities are groups of interacting people sharing common interests, proximity or social relations [36]. They are fundamental to social structure as humans naturally organize their lives around communities they belong to. Sociological trends [41] show that, due to the pressures of time, money, mobility, and technology, people are increasingly withdrawing from communities and disconnecting from their local environment. Consequently, social structures (and social capital) are disintegrating. Technologies that better connect users with their communities and local environment may counter those trends.

Peer-to-peer wireless communications are part of the efforts that seek to create new structures to facilitate engagement, rekindle lost interactions and reconnect with local environments. In addition to infrastructure-based communications (e.g., cellular or WLAN), peer-to-peer communications (e.g., WiFi in ad hoc mode or Bluetooth) enable direct interactions with nearby users, and thus context-awareness: mobile devices can sense their environment. As wireless communications depend on the geographic proximity of mobile devices, peer-to-peer communications provide new ways to share information in real-time for local-area social networking [3, 4, 6, 7, 38], dating [1, 2, 30], gaming [5], or personal safety [39]. To meet the demand for such localized communications, corporations are developing wireless peer-to-peer technologies such as Nokia Instant Community [13] and Qualcomm FlashLinQ [31]. [1]

We study a communication primitive of peer-to-peer wireless technologies that enables users to discover the proximity of specific communities. Users subscribe to communities of interest and their devices then automatically detect wireless messages sent to their community by other members. For example, users can join the community of their local neighborhood (e.g., computer science lab) on Facebook. Using wireless peer-to-peer, they can then obtain *relevant* information in real-time about community events from other community members in proximity (e.g., upcoming lunch break). Wireless peer-to-peer complements infrastructure-based communications by encouraging participation, incentivizing community building and facilitating connection with the local environment.

Sharing information locally in a peer-to-peer fashion leaks data to wireless eavesdroppers, notably putting *data privacy* and *location privacy* at risk. In addition to the above two, the notion of communities brings forth a new set of privacy threat, namely *community privacy*: users from the same community could be linked, thus exposing their social relations [21]. For example, eavesdroppers may learn that two users are part of a controversial political party. Similarly, users' memberships to communities reveals their interests. For example, eavesdroppers may learn whether users passing by a city center belong to a community from a rich neighborhood. Community privacy also affects location and data privacy as it might be easier to infer information about an individual if it is known to belong to a specific community.

The upcoming generation of mobile computing thus requires mechanisms to anonymously identify communities. Existing work suggests the use of *anonymous credentials* [18] and *group/ring signatures* [19, 42]. Although these techniques protect authenticators' privacy, they leak community membership to eavesdroppers. Mechanisms such as *Key-private encryption* [11], *affiliation-hiding envelopes* [29], *hidden credentials* [15] and *oblivious signature-based envelopes* (OSBEs) [33] can privately share information with

[1]Note as well that as cellular networks approach their theoretical communication limits, peer-to-peer wireless offer an alternative to further extend wireless throughput between local devices.

members of a group. However, anyone is able to send messages to a given group when only community members are authorized to communicate with other members. With *Private set intersection* (PSI) [23], several parties can input their community membership as a list of community identifiers, execute the PSI protocol and obtain common communities. But PSI computes the pair-wise intersection of the entire input sets of the communicating parties. It does not scale well for a large number of users and communities.

Affiliation-hiding authentication schemes (secret handshakes) [10] and their extension (Affiliation-Hiding Authenticated Key Exchange (AH-AKE) [26]) enable anonymous authentication of group members with linear complexity for communities discovery [34]. Secret handshakes require exchange of messages and execution of cryptographic operations to authenticate other group members. For example, state-of-the-art AH-AKE [26] requires three communication rounds and two (multi) exponentiations per party and community. Secret handshakes work well for Internet-based scenarios. But in peer-to-peer wireless networks, users will encounter a large number of devices for a *short period of time*, e.g., dozens of encounters per minute in a city center, thus *frequently* invoking group-identification mechanisms. In particular, as interactions are short-lived, these group-identification mechanisms have to operate fast. In addition, battery constraints hinder the use of *computation* and *communication* intensive operations.

Previous work on secret handshakes suggests the use of one-time pseudonyms to achieve unlinkability. However, single-use pseudonyms may require too much storage. Some solutions provide cryptographic unlinkability using zero-knowledge proofs [28] or Key-Private Group Key Management Schemes [27], thus entailing a high cost (communication- and computation-wise). Other solutions include heuristics where users achieve unlinkability by rotating through a small set of pseudonyms, by setting strict time limits on the use of each pseudonym, by using k-anonymous techniques [48], or by associating different pseudonyms with different locations [26]. Yet, the achievable privacy is unclear.

In view of the extensive literature on anonymous group communications and their shortcomings for use in peer-to-peer wireless networks, we aim make to make the following novel contributions:

1. We formally define and quantify the notion of *community privacy* in such networks.

2. By means of a *community pseudonym*-based framework, we study four categories of community identification schemes.

3. We evaluate, both analytically and using simulations, the privacy and cost of these schemes, including existing secret handshake-based schemes.

2. PRELIMINARIES

We introduce next the assumptions made throughout the paper.

2.1 Network Model

We consider a network composed of personal mobile devices that can communicate with an infrastructure such as cellular or WLAN, and in a peer-to-peer fashion upon coming in radio range (e.g., WiFi or Bluetooth). We define $\mathcal{U} = \{U_1, ..., U_n\}$ as the set of users in the network, where n is the total number of users. Mobile devices have a link-layer identifier (i.e., MAC address), network-layer identifier (i.e., IP address) and may have application-layer identifiers (i.e., usernames or cookies). Let $ID_k(t)$ refer to all identifiers of user U_k at time t; we call $ID_k(t)$ a *user pseudonym*. A trusted Central Authority (CA) (in practice, a service provider such as Verisign, Facebook or cellular operator) manages authentication and message confidentiality. Users periodically obtain appropriate authentication credentials (i.e., asymmetric keys) from the CA. The CA may not always be available because of factors such as access costs or limited network availability.

2.2 Community Model

In this work, we define a community as a *group of interacting users* (in the network model described above) sharing common interests. All users in the network can be represented in a graph with a vertex for each user and an edge between two users that are related or share an interest. A community can be then be seen as a complete subgraph or a union of several complete subgraphs that share many of their vertices [36]. In this work, we assume that communities can be centrally formed, for example, in an online fashion such as in Facebook groups or created in a distributed and localized fashion by users using wireless peer-to-peer communications.

Let us define $\mathcal{C} = \{C_1, C_2, ..., C_m\}$ as the set of communities where m is the number of communities in the network. Each community C_i is composed of a set of users $C_i = \{U_k\}$ and has private credentials SK_i (i.e., a *secret*) known only to all community members. At time t, a community is identified by one or multiple *community pseudonyms* $P_{C_i}(t) = \{p_{i,j}\}$ where j is the jth pseudonym. We focus for simplicity on a time period during which community pseudonyms do not change and write $P_{C_i}(t) = P_{C_i}$. The set of all community pseudonyms in the system is then $\mathcal{P} = \bigcup_i P_{C_i}$. Community pseudonyms are generated using a *community pseudonym scheme* (discussed in Section 4). We consider that each user belongs to a fixed number of communities n_c and has knowledge of all valid pseudonyms of that community.

2.3 Communication Model

As in this work we focus on the privacy of community members interacting in a localized and close-range setting, we assume that users communicate with each other only using wireless peer-to-peer communications. Users use infrastructure-based communications to access CA and third-party services containing community membership information. In order to automatically detect the presence of other users in radio range, mobile devices periodically broadcast proximity beacons of the form $U_k \rightarrow * : ID_k(t) \mid t$ where $ID_k(t)$ is the user pseudonym of U_k at time t. We consider an energy-efficient contention-based beaconing mode, similar to the one used in IEEE 802.11 ad hoc mode which distributes the beaconing task uniformly in the user neighborhood. A mobile user may reply directly to the sender with a unicast message.

In addition to unicast (user-user) interactions, mobile users can exchange information with user groups or communities. A *community packet* is a message sent to a community of users which contains a *community pseudonym* that serves to identify the community. Community packets broadcasted by user U_k to community C_i at time t have the following form, $U_k \rightarrow C_i : ID_k(t) \mid p_{i,j} \mid msg$ where $p_{i,j}$ is the j-th community pseudonym of C_i (i.e., packet destination) and msg is the message. Receivers who belong to C_i have knowledge of all pseudonyms of C_i and rely on $p_{i,j}$ to detect messages sent to this community. Unicast messages of the following form $U_j \rightarrow U_k : ID_j(t) \mid ID_k(t) \mid msg$ can then be used to establish communication channel with members of the same community. In order to identify members of the same or a particular community in the neighborhood, users advertise all communities they belong to using such community packets. Moreover, all communications, i.e., both unicast and community packets, are *multi-hop* and standard *energy-efficient routing* algorithms [16] are used.

2.4 Threat Model

An adversary $\mathcal{A}$ can jeopardize user privacy by extracting messages content (data privacy), obtaining users' locations (location privacy) or detecting community members (community privacy).

In order to avoid user pseudonyms' traceability, users can change their pseudonyms $ID_k(t)$ over time [12, 24]. As a pseudonym changed by an isolated node can be trivially guessed by an external party, pseudonym changes should be coordinated in regions called *mix zones* [12, 17, 25, 32]. In this work, we consider that mobile users coordinate pseudonym changes as defined in [32] and achieve location privacy at a low cost.

In some cases, users may broadcast private messages to other community members, and in others they may share their messages with everyone. Several mechanisms can be used to encrypt *msg*. Without loss of generality, we consider that the CA sets up shared secrets: upon registration to a community C_i, a user is given a symmetric key SK_i. Community packets can thus be encrypted. Other distributed solutions [8, 9, 20, 22, 44] could also be used. For simplicity, we consider centrally generated symmetric keys and focus on the problem of *loss of community privacy* due to community pseudonyms. Revocation challenges are discussed in Section 6.

For the community privacy threat, we consider both a *passive* and an *active* adversary. In order to abstract the adversary's strengths and the possible attacks on community privacy, we model the adversary's capabilities as a set of *oracles* as follows. A passive adversary $\mathcal{A}$ will collect broadcast messages and obtains communication *traces*. By using well-known *traffic analysis* techniques on the collected traces, $\mathcal{A}$ infers the relation between community pseudonyms and communities. We call s the number of packets collected by $\mathcal{A}$. At best, $\mathcal{A}$ is *global* and collects all packets in the network. Let $O_P = \mathsf{TrafficAnalysis}(s)$ be the passive oracle that captures such a traffic analysis attack. $\mathcal{A}$ inputs s messages to O_P that outputs a partial mapping between the pseudonyms and communities.

In addition to passive capabilities, an active $\mathcal{A}$ can compromise a subset D of mobile devices. An active Oracle O_A, on input D, outputs community memberships of D and all corresponding pseudonyms. We model active attacks with four oracles $O_A \subseteq \{\mathsf{Q}, \mathsf{R}, \mathsf{J}, \mathsf{C}\}$. $\mathsf{Q}=\mathsf{Query}(D, C)$: Sending forged/replayed community packets of community C to D and waiting for D's response. A unicast reply from D reveals membership to a community C. $\mathsf{R}=\mathsf{Reveal}(D, C)$: Revealing D's membership to community C by direct access to the devices D, e.g, hardware hacking. $\mathsf{J}=\mathsf{Join}(D, C)$: Joining community C of D, for example, using social engineering attacks. $\mathsf{C}=\mathsf{Create}(D, C)$: Creating social link with D by creating fake community C and inviting D to that community. This can be used to identify pseudonyms of other communities of D.

3. COMMUNITY PRIVACY PROBLEM

The goal of this paper is to study the community privacy problem in the system model described above by formalizing the various notions of community privacy. But before formalizing privacy leakage, let us first understand the type of attacks on community privacy that are possible considering the adversarial capabilities as discussed in Section 2.4.

3.1 Probabilistic Attacks

In *probabilistic attacks*, the adversary constructs a probabilistic mapping of community pseudonyms $\mathcal{P}$ to communities $\mathcal{C}$. Such a mapping between every pseudonym and its corresponding community(ies) can be represented by a weighted bipartite graph $\mathbf{G}$ as shown in Fig. 1 (a). Every edge connects a vertex in $\mathcal{P}$ to one in $\mathcal{C}$ and is weighted by the probability (estimated by the adversary $\mathcal{A}$) of linking a pseudonym to a community. Such probabilistic attacks are generally carried out by a passive adversary who collects

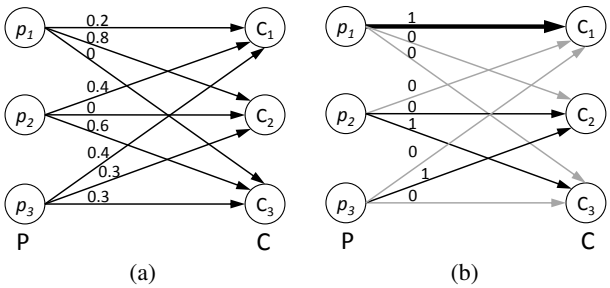


Figure 1: Illustration of probabilistic and deterministic attacks. (a) Weighted bipartite graph G. (b) Weighted bipartite graph $\mathbf{G}'$ resulting from the combination of G and $\mathbf{G_A}$. The bold edge between p_1 and C_1 belongs to $\mathbf{G_A}$. Assuming each community uses a single pseudonym, the adversary can update weights of edges in G using $\mathbf{G_A}$. Here, $\mathcal{A}$ learns all mappings between community pseudonyms and communities.

s packets and obtains a mapping of community pseudonyms $\mathcal{P}$ to communities $\mathcal{C}$ (or in other words $\mathbf{G}$) by using the traffic analysis oracle O_P. The number of collected messages s indicates the strength of $\mathcal{A}$. We discuss multiple ways to obtain such graphs in Section 5.

3.2 Deterministic Attacks

In *deterministic attacks*, the adversary constructs a fixed mapping of community pseudonyms $\mathcal{P}$ to communities $\mathcal{C}$. The result of such an attack can be represented by a weighted bipartite graph $\mathbf{G_A}$ as shown in Fig. 1 (b). If there is an edge between a vertex in $\mathcal{P}$ to one in $\mathcal{C}$ in $\mathbf{G_A}$, then its weight is either 1 or 0 meaning that the pseudonym either belongs to the community or not. Such deterministic attacks are generally carried out by an active adversary who learns about communities and their pseudonyms by interacting with the oracle O_A.

Let us represent the result of applying oracle O_A to device D by $W(D)$: $W(D)$ is a mapping between communities and community pseudonyms. Then, a deterministic attack is a set of calls to O_A by an active adversary $\mathcal{A}$: $\{W_1(D_1), ..., W_\ell(D_\ell)\}$ where ℓ is the number of interactions and a system parameter indicating the strength of $\mathcal{A}$. At each interaction, $\mathcal{A}$ changes the device input to O_A. Note that two interactions $W_i(D_1), W_j(D_2), i \neq j$ may produce identical mappings. Similarly, an interaction may be unsuccessful and output an empty result, $W_i(D) = \emptyset$. The set of community pseudonyms $P^{\mathcal{A}}_{C_i}$ of C_i known to the adversary increases depending on ℓ. In general, $\mathbf{G_A}$ depends on the number of different devices input to the Oracle (i.e., the number of devices under attack) and the larger $\mathbf{G_A}$ (in terms of the number of edges) is, the more successful is the attack. An active adversary can combine the information from the probabilistic graph $\mathbf{G}$ to produce a more accurate version of $\mathbf{G_A}$ and is represented as $\mathbf{G}'$ (Fig 1 (b)).

3.3 Community Privacy Properties

Based on the graph $\mathbf{G}$ or $\mathbf{G}'$, an adversary can threaten two privacy properties, namely, *Community Anonymity* (CAN) and *Community Unlinkability* (CUN). In order to preserve community privacy, any community pseudonym scheme must satisfy both CAN and CUN. Below we formalize each of this property.

3.3.1 Community Anonymity or CAN

Community anonymity or CAN guarantees that users cannot be linked by third parties or non-community users to the communities they belong to, i.e., community pseudonyms do not affect the anonymity of the community members. Informally, there is com-

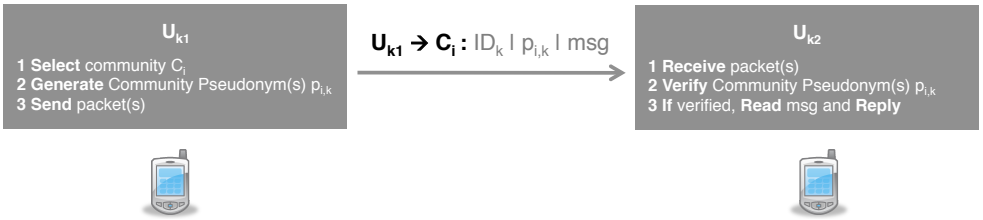


Figure 2: Overview of community pseudonym schemes. For each community packet, users need to generate a community pseudonym. Each community pseudonym scheme thus implements *generate* and *verify* functions.

munity anonymity at any time t if and only if for all communities C_i and for all pseudonyms $p_{i,j}$ of C_i (at that time), only members of C_i are able to deterministically verify that $p_{i,j}$ is a valid pseudonym of C_i. We formalize the notion of community anonymity by means of a *challenge-response* game between the adversary (who has access to the oracles) and an unbiased challenger as follows. For convenience, we consider the more stronger adversary model, i.e., an active adversary.

1. $\mathcal{A}$ collects s messages, interacts with O_P and obtains $\mathbf{G}$. She interacts ℓ times with oracle O_A and obtains $\mathbf{G_A}$. She combines $\mathbf{G}$ and $\mathbf{G_A}$ to obtain $\mathbf{G}'$.
2. $\mathcal{A}$ randomly queries challenger one community $C_i \notin \mathbf{G}'$ (i.e., for which $\mathcal{A}$ does not have information from interactions with oracles).
3. Challenger throws an unbiased coin and selects $b \in \{0,1\}$ based on the output of the coin toss. If $b = 0$, it sends $p_b \in C_i$ to $\mathcal{A}$, else it sends $p_b \notin C_i$ as a challenge.
4. $\mathcal{A}$ decides whether p_b belongs to C_i and outputs b' (as a guess for b).

Then, the advantage of the adversary in the community anonymity game is measured in terms of how successful she is in correctly guessing the community of the challenge pseudonym compared to a random guess.

$$Adv_{s,\ell}^{CAN}(\mathcal{A}) = Pr(\mathcal{A} \text{ is correct}) - \frac{1}{2} \quad (1)$$

where, ℓ is the length of the interaction of the adversary with oracle O_A, and s is the number of messages collected by the adversary. The advantage gives the relationship between the number of interactions ℓ, the number of messages collected s, and the probability of success of the adversary. If the adversary guesses uniformly at random ($Pr(\mathcal{A} \text{ is correct}) = 1/2$), then the advantage is $Adv_{s,\ell}^{CAN} = 0$, i.e., lowest. Similarly, if the adversary surely knows the pseudonym membership ($Pr(\mathcal{A} \text{ is correct}) = 1$), then the advantage is $Adv_{s,\ell}^{CAN} = 1/2$, i.e., highest. In other words, the advantage always lies in $[0, 1/2]$. We can now define CAN based on the adversary's advantage in winning the above game:

DEFINITION 1. *A community pseudonym scheme provides Community Anonymity (CAN) if and only if the advantage $Adv_{s,\ell}^{CAN}(\mathcal{A})$ of any adversary $\mathcal{A}$ with s accesses to oracle O_P and l accesses to oracle O_A is negligibly small.*

3.3.2 Community Unlinkability

Community unlinkability or CUN guarantees that users of the same community cannot be linked to each other as belonging to the same community by third parties or non-community users. Informally, there is *community unlinkability* at any time t if and only if for all communities C_i and for any two pseudonyms $p_{i,j}$ and $p_{i,k}$ of C_i (at that time), only members of C_i are able to deterministically verify that $p_{i,j}$ and $p_{i,k}$ belong to the same community. Similar to the previous case, we use a challenge-response methodology to formalize the notion of community unlinkability.

1. $\mathcal{A}$ collects s messages, interacts with O_P and obtains $\mathbf{G}$. She interacts ℓ times with oracle O_A and obtains $\mathbf{G_A}$. She combines $\mathbf{G}$ and $\mathbf{G_A}$ to obtain $\mathbf{G}'$.
2. $\mathcal{A}$ randomly queries challenger two communities $C_i, C_j \notin \mathbf{G}'$ (i.e., for which $\mathcal{A}$ does not have information from interactions with oracles).
3. Challenger throws two independent unbiased coins and selects $b \in \{i, j\}$ and $d \in \{i, j\}$ based on the output of the coin tosses and sends $p_b \in C_b, p'_d \in C_d$ to $\mathcal{A}$ as a challenge.
4. $\mathcal{A}$ decides whether p_b and p'_d belong to the same community and outputs yes/no.

Then, the advantage of the adversary in the community unlinkability game is measured in terms of how successful she is in correctly linking the challenge pseudonyms compared to a random guess.

$$Adv_{s,\ell}^{CUN}(\mathcal{A}) = Pr(\mathcal{A} \text{ is correct}) - \frac{1}{2} \quad (2)$$

Similar to CAN, we can now define CUN based on the adversary's advantage in winning the above game:

DEFINITION 2. *A community pseudonym scheme provides Community Unlinkability (CUN) if and only if the advantage $Adv_{s,\ell}^{CUN}(\mathcal{A})$ of any adversary $\mathcal{A}$ with s accesses to oracle O_P and l accesses to oracle O_A is negligibly small.*

We have the following relationship between the CAN and CUN properties:

THEOREM 1. *CUN implies CAN, but CAN does not imply CUN.*

This theorem shows that unlinkability is a stronger notion than anonymity in community identification protocols. We prove the above theorem in Appendix A.

4. COMMUNITY PSEUDONYM SCHEMES

In order to provide efficient and anonymous identification of communities, we propose to generate community identifiers using symmetric cryptography. We describe four classes of community pseudonym mechanisms and evaluate their cost and security. Each

community pseudonym mechanism is composed of a *generate* and *verify* function (Fig. 2). We consider that pseudonyms are defined over B bits and thus there are $M = 2^B$ possible pseudonyms. We assume that when community pseudonyms change, the user pseudonym $ID_k(t)$ changes as well, and vice versa.

4.1 Single Pseudonym Schemes

A single *constant* pseudonym is used per community: we have $P_{C_i} = p_i$ where p_i is chosen uniformly at random in $\{0,1\}^B$ and $|P_{C_i}| = 1$. We consider two possible techniques to generate community pseudonyms:

$$i) \quad U_k \xrightarrow{p_{i,k}} C_i$$
$$ii) \quad U_k \xrightarrow{p_i} C_i$$

i) each user uses a single pseudonym per community that is different from that of other users from the same community (similar to linkable secret handshakes);
ii) One pseudonym is used by all users per community (similar to group signatures). In practice, the community pseudonym can be a Hash: $p_i = \mathcal{H}(SK_i)$, where $\mathcal{H}(.)$ is a Hash function.

The sender of a message has low computation and communication overhead: it selects one pseudonym per community, i.e., $O(m)$ lookups, and send one message per community it belongs to, i.e., $O(m)$ communications. For all community pseudonyms received from one neighbor, a receiver has to compare these community pseudonyms to all community pseudonyms of communities it belongs to. The complexity of such *lookups* depends on the data structure used to store community pseudonyms (e.g., hashmaps or trees). As wireless messages are broadcast in nature, lookup operations are done for each device in communication range. Assuming hashmaps, the number of lookups at the receiver is: $O(n_e m)$, where n_e is the number of encountered nodes.

One extension consists in relying on the k-anonymity concept [48]. For each community pseudonyms, users select $\mathsf{k}-1$ other community pseudonyms that they send together with their messages:

$$U_k \xrightarrow{p_i, \overbrace{p_r, p_r, p_r, p_r, \cdots}^{\mathsf{k}-1}} C_i$$

In practice, these extra community pseudonyms are chosen from communities the sender does not belong to (e.g., pseudonyms eavesdropped in previous interactions) chosen at random (p_r). This increases the cost at the receiver as the number of lookups is $O(n_e \mathsf{k} m)$.

4.2 Multiple Pseudonyms over Entire Domain

Each community C_i is identified by a *set* of pseudonyms known to all community members $P_{C_i} = \{p_{i,j}\}$ where j is the jth pseudonym of C_i. To send a packet to C_i, a user randomly selects a pseudonym from P_{C_i}. Receivers determine whether it is sent by a member of their community by searching their local pseudonym repository for received community pseudonyms. We consider two possible generation mechanisms to assign pseudonyms across communities such that $P_{C_i} \bigcap P_{C_j} = \emptyset, \forall i \neq j$:

$$i) \quad U_k \xrightarrow{p_{i,j} \in P_{C_i}} C_i$$
$$ii) \quad U_k \xrightarrow{\text{RND} \,||\, \text{HMAC}_{SK_i}(\text{RND})} C_i$$

i) *Pre-computed Schemes:* The CA randomly splits all pseudonyms across communities: every C_i is assigned $\lfloor M/m \rfloor$ pseudonyms.
ii) *Self-generated Schemes:* Every user generates a community pseudonym for each message to C_i by choosing a random number RND and computing a message authentication code: $p_{i,j} = \text{RND} \,||\, \text{HMAC}_{SK_i}(\text{RND})$. For every message, a receiver verifies the HMAC with the key of all communities it belongs to.

Pre-computed schemes result in *large* storage costs. If there were at most $m = 100000$ communities and $B = 48$ bits long pseudonyms, every user would store $MB/8m \approx 16$ GB per community. The sender/receiver do not perform computations. The sender does $O(m)$ lookups and broadcasts $O(m)$ messages. The receiver does $O(m)$ lookups for each message. Given n_e encountered nodes, the receiver does a total of $O(n_e m)$ lookups.

In self-generated schemes, the sender computes $O(m)$ hashes and sends $O(m)$ messages. The receiver hashes all received messages with the secret key of all their communities, i.e., $O(n_e m^2)$ *Hashes*, and compares computed hashes to the received hashes.

Hash bins [34] can reduce computation overhead to $O(n_e m \log m)$ at the expense of $O(m)$ Hashes at the sender. Index-Hiding Message Encoding vectors (IHME) [34] can further decrease computation overhead to $O(n_e m)$ by using polynomial interpolation.

4.3 Multiple Pseudonyms over Shrunk Domain

These schemes assign fewer community pseudonyms to each community (i.e., some pseudonyms are not assigned at all). They reduce the size of sets of community pseudonyms according to a shrink factor $h \in [0,1]$. The goal is to reduce cost and make it more difficult for an adversary to relate community pseudonyms to communities (as some pseudonyms are not assigned). Formally, we have: $|P_{C_i}| = (M/m) \cdot h$ and $P_{C_i} \bigcap P_{C_j} = \emptyset, \forall i \neq j$. Similar to the earlier case, we consider two mechanisms to generate pseudonyms:

$$i) \quad U_k \xrightarrow{p_{i,j} \in P_{C_i,h}} C_i$$
$$ii) \quad U_k \xrightarrow{\mathcal{H}(p_{i,j})} C_i$$

i) *Pre-computed Schemes:* The CA assigns a subset of pseudonyms $P_{C_i,h}$ across communities: every community C_i receives $\lfloor h \cdot M/m \rfloor$ community pseudonyms;
ii) *Self-generated Schemes:* All users belonging to the same community generate a Hash chain in a synchronized fashion: $p_{i,1} = \mathcal{H}(SK_i)$, $p_{i,j+1} = \mathcal{H}(p_{i,j})$ for $1 < j < len$ where len is the Hash chain length.

In terms of storage cost of a pre-computed scheme, it is significantly lower than schemes over the entire domain. For example, users must now store only 160 megabytes of data if $h = 0.01$, $B = 48$ and $m = 100000$. The receiver cost is again $O(n_e m)$ lookups.

With self-generated schemes, users can verify if a message is destined to them by checking whether received community pseudonyms belong to their Hash chains. Self-generated schemes enable verification with $O(n_e m)$ lookups without online operations.

4.4 Hints: Overlapping Multiple Pseudonyms

The third multiple pseudonym scheme allows for an overlap in the set of pseudonyms used for each community. In other words, community pseudonyms can be used by more than one community. Overlapping pseudonym sets creates confusion for the adversary. We define the overlap factor $o \in [0,1]$ as the fraction of community pseudonyms shared by different communities. We use the term "hint" because a community pseudonym does not uniquely identify a community but rather hints receivers to determine whether a messages is destined to them. We have: $P_{C_i} \bigcap P_{C_j} \neq 0$ for some $i \neq j$. We consider two possible mechanisms:

$$i) \quad U_k \xrightarrow{p_{i,j} \in P_{C_i,o}} C_i$$
$$ii) \quad U_k \xrightarrow{\text{RND} \,||\, \lfloor \text{HMAC}_{SK_i}(\text{RND}) \rfloor_o} C_i$$

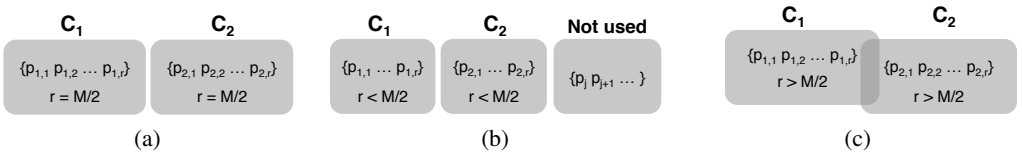


Figure 3: Community pseudonym schemes with multiple pseudonyms per community (assuming only two communities for illustration purposes). (a) Multiple pseudonyms over the entire domain. (b) Multiple pseudonyms over a shrunk domain. (c) Overlapping multiple pseudonyms, i.e., hints.

Table 1: Cost of community pseudonym schemes with m communities per participants and n_e participants.

Technique	Sender			Receiver		
	Lookups	Computation	Communication	Lookups	Computation	Memory
Single pseudonym	$O(m)$	$\emptyset$	$O(m)$	$O(n_e m)$	$\emptyset$	$O(m)$
k-anonymity	$O(km)$	$\emptyset$	$O(km)$	$O(n_e km)$	$\emptyset$	$O(km)$
Pre-computed entire	$O(m)$	$\emptyset$	$O(m)$	$O(n_e m)$	$\emptyset$	$O(M)$
Self-generated entire	$O(m)$	$O(m)$	$O(m)$	$\emptyset$	$O(n_e m^2)$	$O(m)$
Hash bins	$O(m)$	$O(m)$	$O(m)$	$\emptyset$	$O(n_e m \log(m))$	$O(m)$
IHME	$O(1)$	$\emptyset$	$O(m)$	$\emptyset$	$O(n_e m)$	$O(m)$
Pre-computed shrunk	$O(m)$	$\emptyset$	$O(m)$	$O(n_e m)$	$\emptyset$	$O(hM)$
Self-generated shrunk	$O(m)$	$\emptyset$	$O(m)$	$O(n_e m)$	$\emptyset$	$O(len \cdot m)$
Hints	$O(m)$	$O(m)$	$O(m)$	$\emptyset$	$O(n_e m^2)$	$O(m)$

i) *Pre-computed Schemes:* The CA splits the set of all pseudonyms across communities $P_{C_i,o}$ but assigns some pseudonyms to multiple communities;
ii) *Self-generated Schemes:* Hints can be implemented by truncating the output of the Hash method of self-generated schemes to a smaller number of bits. Hence, several RND values have the same Hash, thus creating identifier collisions. The larger the truncation of the Hash is, the larger the overlap will be.

In addition to cost of self-generated schemes over the entire domain, users get messages that are not destined to them because of Hash collisions (i.e., false positives). If these messages are encrypted, users unsuccessfully attempt decryption. The number of unsuccessful decryptions depends on the number of collisions.

Table 1 summarizes the asymptotic communication and computation costs of the four community pseudonym scheme types outlined above. As computations correspond to Hash operations, they are significantly lower than the cost of asymmetric cryptography. Similarly, lookup operations are considerably cheaper than Hash operations. Some schemes avoid online computations but suffer from trivial linkability or high storage costs. Other schemes overcome high storage costs by introducing online computations. Although Hash bins provide logarithmic complexity, they are inefficient communication-wise because all bins must be sent even if users belong to few communities. IHME-based schemes are efficient, but trivially linkable because of the polynomial's uniqueness.

5. EVALUATION

We evaluate community privacy (in terms of CAN and CUN) with respect to probabilistic and deterministic attacks. As previously described, $\mathcal{A}$ can obtain $\mathbf{G}$ from probabilistic attacks, $\mathbf{G_A}$ from deterministic attacks and $\mathbf{G}'$ from the combination of $\mathbf{G}$ and $\mathbf{G_A}$.

5.1 Community Anonymity Analysis

Let us define $\rho =$ "$\mathcal{A}$ solves the CAN challenge". The probability that $\mathcal{A}$ successfully answers a CAN challenge depends on its information about community pseudonyms:

$$\begin{aligned} \sigma = Pr(\rho) &= Pr(\rho | p_b \in \mathbf{G_w}) Pr(p_b \in \mathbf{G_w}) \\ &+ Pr(\rho | p_b \in \mathbf{G_f}) Pr(p_b \in \mathbf{G_f}) \end{aligned} \quad (3)$$

where $\mathbf{G_f}$ is the subgraph of $\mathbf{G}$ containing all edges with weight equal to 0 or 1, $\mathbf{G_w}$ is the subgraph of $\mathbf{G}$ containing all edges with weight in $(0,1)$ and p_b is the community pseudonym send to adversary by the challenger. More specifically,

$$Pr(\rho) = \sum_{C_i \notin \mathbf{G_A}} Pr(\text{"}\mathcal{A} \text{ picks } C_i\text{"}) Pr(\rho_i) \quad (4)$$

where $Pr(\rho_i)$ is $Pr($"$\mathcal{A}$ solves the CAN challenge for C_i"$)$. In other words, $\mathcal{A}$ may know the community of p_b ($\mathbf{G_f}$) or have statistical information ($\mathbf{G_w}$) about the community of p_b. The probabilities $Pr(p_b \in \mathbf{G_x})$ depend on the type of adversary (i.e., probabilistic or deterministic attacker), its strength (s and ℓ) and on community pseudonym schemes.

5.1.1 Probabilistic Adversary

Given Eq. (1) and (3), the CAN advantage for $\mathcal{A}$ is:

$$Adv_s^{CAN} = Pr(\rho | p_b \in \mathbf{G_w}) - \frac{1}{2} \quad (5)$$

Indeed, in the probabilistic case, the graph $\mathbf{G_f}$ is empty, as the adversary cannot know with probability 1 the relation between community pseudonyms and communities. Hence, we have $Pr(p_b \in \mathbf{G_f}) = 0$, or equivalently, $Pr(p_b \in \mathbf{G_w}) = 1$. Equation (5) shows that CAN exclusively depends on the information contained in $\mathbf{G_w}$, i.e., $\mathcal{A}$'s ability to link community pseudonyms to communities.

5.1.2 Deterministic Adversary

An deterministic adversary $\mathcal{A}$ knows the relation between *some* community pseudonyms and communities with probability 1. The

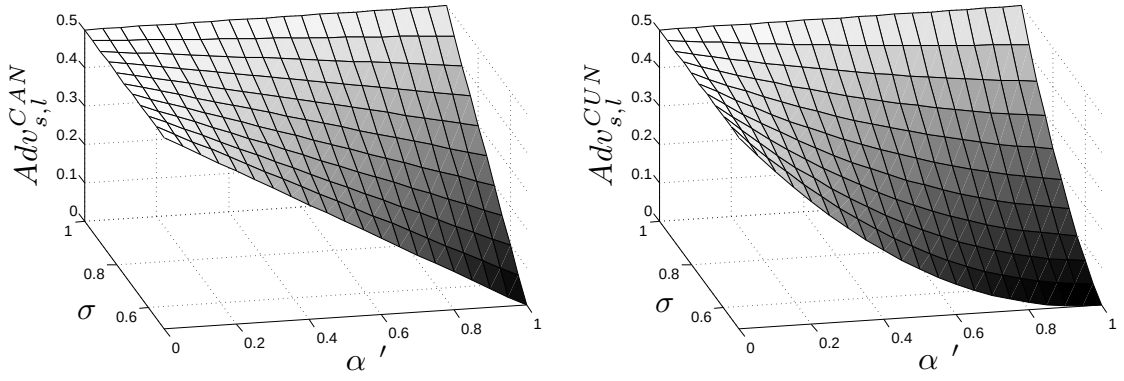


Figure 4: Numerical evaluation of advantage of probabilistic and deterministic adversary. (a) $Adv_{s,\ell}^{CAN}$ and (b) $Adv_{s,\ell}^{CUN}$. Adv_s^{CAN} (Adv_s^{CUN}) corresponds to $Adv_{s,\ell}^{CAN}$ ($Adv_{s,\ell}^{CUN}$) with $\alpha' = 1$.

CAN advantage is then:

$$Adv_{s,\ell}^{CAN} = \begin{cases} (Pr(\rho|p_b \in \mathbf{G'_w})\alpha' + (1-\alpha')) - \frac{1}{2} & \text{if } |v'_f| < |\mathcal{P}| - 1 \\ \frac{1}{2} & \text{else} \end{cases} \quad (6)$$

with

$$\alpha' = \frac{1}{2}\frac{1}{|V|}\sum_{v\in V}\frac{|e_w^v|}{|e^v|} + \frac{1}{2}\frac{1}{|V|}\sum_{v\in V}\frac{|e_w| - |e_w^v|}{|e| - |e^v|} \quad (7)$$

where $\mathbf{G'_w}$ is the subgraph of $\mathbf{G'}$ containing all edges with weight in $(0,1)$, $\mathbf{G'_f}$ is the subgraph of $\mathbf{G'}$ containing all edges with weight equal to 0 or 1, V is the set of nodes in the communities of $\mathbf{G'}$, v'_f are the nodes in V with an edge with weight 1, e_w are the edges in $\mathbf{G'_w}$, e^v are the edges connected to a node v, e_w^v are the edges in $\mathbf{G'_w}$ connected to a node v, $\mathcal{P}$ is the set of all community pseudonyms, α' is the probability that a pseudonym belongs to $\mathbf{G'_w}$ and $1-\alpha'$ is the probability that a pseudonym belongs to $\mathbf{G'_f}$. Equation (6) indicates that with probability $1-\alpha'$ the adversary knows the challenge and is always successful, whereas with probability α', the adversary guesses based on $\mathbf{G'_w}$. The probability α' first depends on the probability that the challenger selects the community C_i queried by $\mathcal{A}$ (1/2) and on edges' proportion that belong to $\mathbf{G'_w}$ in C_i. Second, it depends on the probability that the challenger does not select the C_i queried by $\mathcal{A}$ (1/2) and on edges' proportion that belong to $\mathbf{G'_w}$ in $\mathcal{C} - C_i$.

If $\alpha' = 0$ (i.e., community pseudonyms exclusively belong to $\mathbf{G'_f}$ as ℓ is large), then $Adv_{s,\ell}^{CAN} = 1/2$ indicating that $\mathcal{A}$ always guesses right. If $\alpha' = 1$ (i.e., community pseudonyms exclusively in $\mathbf{G'_w}$), then $Adv_{s,\ell}^{CAN} = Pr(\rho|p_b \in \mathbf{G'_w}) - 1/2$ as for probabilistic $\mathcal{A}$.

5.2 Community Unlinkability Analysis

Let us define $\upsilon =$ "$\mathcal{A}$ solves the CUN challenge". As before, we can write:

$$\begin{aligned} \mu = Pr(\upsilon) &= Pr(\upsilon|p_b, p_d \in \mathbf{G_f})Pr(p_b, p_d \in \mathbf{G_f}) \quad (8) \\ &+ Pr(\upsilon|p_b, p_d \in \mathbf{G_w})Pr(p_b, p_d \in \mathbf{G_w}) \\ &+ Pr(\upsilon|p_b \in \mathbf{G_f}, p_d \in \mathbf{G_w})Pr(p_b \in \mathbf{G_f}, p_d \in \mathbf{G_w}) \\ &+ Pr(\upsilon|p_b \in \mathbf{G_w}, p_d \in \mathbf{G_f})Pr(p_b \in \mathbf{G_w}, p_d \in \mathbf{G_f}) \end{aligned}$$

5.2.1 Probabilistic Adversary

Given Eq. (2) and (8), the CUN advantage is:

$$Adv_s^{CUN} = Pr(\upsilon|p_b, p_d \in \mathbf{G_w}) - \frac{1}{2} \quad (9)$$

Theorem 1 shows that if $\mathcal{A}$ breaks CAN challenge, i.e., $\sigma \in (0.5, 1]$, then it also breaks CUN. Hence, the probability of breaking the CUN challenge in the passive case η is:

$$\eta = Pr(\upsilon|p_b, p_d \in \mathbf{G_w}) = \sigma^2 + \frac{(1-\sigma)^2}{2} + \frac{\sigma(1-\sigma)}{2} \quad (10)$$

The advantage is obtained as follows: in the probabilistic case, $\mathbf{G_f}$ is empty. Hence, we have $Pr(p_b, p_d \in \mathbf{G_f}) = Pr(p_b \in \mathbf{G_w}, p_d \in \mathbf{G_f}) = Pr(p_b \in \mathbf{G_f}, p_d \in \mathbf{G_w}) = 0$ and $Pr(p_b, p_d \in \mathbf{G_w}) = 1$. We relate the probability of success υ to the probability of success σ as $\mathcal{A}$ can run the CAN challenge response protocol for both communities, and this determines its success rate for the CUN challenge: $\mathcal{A}$ can guess both CAN challenges correctly (σ^2); $\mathcal{A}$ cannot guess both CAN challenges correctly but answers the CUN challenge correctly ($(1-\sigma)^2/2$); or $\mathcal{A}$ can guess one of the CAN challenges correctly and CUN correctly ($\sigma(1-\sigma)/2$).

We observe that if $\sigma = 0$ (meaning that $\mathcal{A}$ does not break CAN), then $\eta = 1/2$ and the advantage is minimum. Instead, if $\sigma = 1$, then the probability of success $\eta = 1$, indicating that the adversary has maximum advantage.

5.2.2 Deterministic Adversary

If $\mathcal{A}$ is a deterministic adversary, it discovers the relation between some community pseudonyms and communities. Using Eq. (3) and (8), we compute the advantage:

$$Adv_{s,\ell}^{CUN} = (1-\alpha')^2 + \eta\alpha'^2 + 2\sigma'\alpha'(1-\alpha') - \frac{1}{2} \quad (11)$$

We obtain the above formula by relating the CUN advantage to the CAN challenge response game. With probability $(1-\alpha')^2$, $\mathcal{A}$ is given two pseudonyms that it knows (in $\mathbf{G'_f}$) and always guesses correctly. With probability α'^2, $\mathcal{A}$ does not know any of the pseudonyms and guess with success η. Finally, with probability $\alpha'(1-\alpha')$, $\mathcal{A}$ knows one of the two pseudonyms and guesses the other pseudonym with success σ'.

If $\sigma' = 1/2$ (i.e., CAN algorithm does not help), then the advantage is $1 - \alpha' + \alpha'^2/2$. If $\alpha' = 0$, then the advantage is maximal

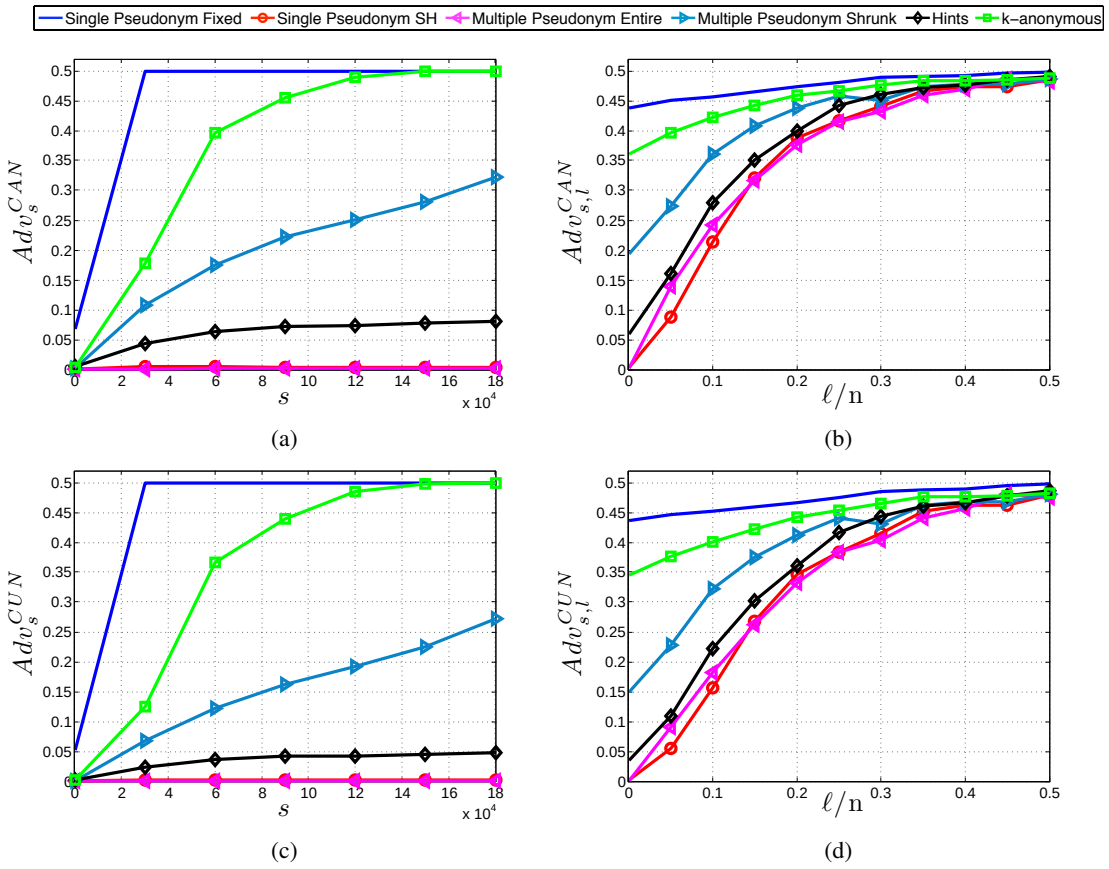


Figure 5: CAN and CUN Advantages obtained in simulations of a probabilistic and deterministic adversary for different community pseudonym schemes. (a) Adv_s^{CAN} with respect to the number of collected messages s and (b) $Adv_{s,l}^{CAN}$ with respect to the fraction of compromised devices ℓ/n. (c) Adv_s^{CUN} with respect to the number of collected messages s and (d) $Adv_{s,l}^{CUN}$ with respect to the fraction of compromised devices ℓ/n.

(0.5). Instead, if $\alpha' = 1$, then the advantage is minimal (0). If $\sigma = 1'$, then the advantage is maximal (0.5) for any value of α' indicating that if $\mathcal{A}$ solves the CAN challenge, it also solve the CUN challenge. As we assume CUN attack based on CAN attack, Eqs. 10 and 11 are lower-bounds for CUN attack's success. $\mathcal{A}$ could obtain better solutions independent of CAN advantage.

5.3 Empirical Evaluation

In order to better understand the analysis presented earlier, we numerically evaluate the CAN/CUN advantages for the community pseudonym schemes outlined earlier under different adversarial assumptions. We later verify these numerical evaluations using simulation experiments.

5.3.1 Numerical Evaluation

In Fig. 4, we numerically evaluate CAN and CUN advantages with respect to σ and α' by plotting (6) and (11). We observe in Fig. 4 (a) that the probabilistic or passive CAN advantage ($\alpha' = 1$) increases linearly with σ. Probability σ is an increasing function of s that depends on adversary's attack and on community pseudonym schemes. In general, the higher the number of collected messages s is, the higher the advantage of the adversary is. In contrast, the passive CUN advantage (Fig. 4 (b) with $\alpha' = 1$) increases non-linearly as indicated in Eq. (10). We observe that as α' decreases, the advantage dramatically increases meaning that compromising devices considerably helps the adversary. This shows how breaking CAN affects the success probability of CUN. We now investigate with simulations the missing relation between σ and s.

5.3.2 Simulation Setup

Our simulator models mobile users using aforementioned community pseudonym schemes and probabilistic/deterministic attacks on community privacy. We consider $n = 50$ users moving at constant speed on a grid of 1km $\times$ 1km with one meter steps using traditional random walk mobility [46]. Directions are chosen out of $[0, 2\pi]$ with granularity $\pi/2$. Nodes are in communication range if they are within 100 meters. We consider that there are $m = 20$ communities and $M = 100000$ community pseudonyms, that users belong to $n_c = 6$ communities, and that users interact with communication protocol of Section 2. In a time slot t, devices send a message to all their communities and, if possible, change community pseudonyms. $\mathcal{A}$ collects all messages: $s = t \cdot n \cdot n_c$.

5.3.3 Attack Description

In our simulations, the goal of the probabilistic adversary $\mathcal{A}$ is to obtain the graph $\mathbf{G}$ relating community pseudonyms and communities. The attack consists of traffic analysis and community detection. With traffic analysis, $\mathcal{A}$ obtains a graph $\mathbf{G_e}$ where community pseudonyms are nodes, and weighted edges indicate the probability that community pseudonyms belong to the same community.

$\mathcal{A}$ links community pseudonyms based on wireless communication patterns. A unicast communication occurring after two nodes A and B exchange community pseudonyms indicates that they have at least one community in common. $\mathcal{A}$ thus links community pseudonyms of A to those of B.

$\mathcal{A}$ then groups community pseudonyms into communities using community detection algorithms on graph $\mathbf{G_e}$. Previous work [14] obtained efficient community detection algorithms on graphs. In our setting, we use *adversarial* community detection [35] algorithms as standard community detection algorithms may fail because of the use of community pseudonyms. After $\mathcal{A}$ infers communities, it must guess the relation between inferred and real communities relying on background information on communities profile. We consider a strong $\mathcal{A}$ that correctly maps inferred communities to corresponding real identities. In reality, determining this mapping is non-trivial, but we leave this for future work. We compute the probability of success of $\mathcal{A}$ by considering the overlap and non-overlap between inferred and real communities.

The deterministic adversary $\mathcal{A}$ performs a similar attack except that she, in addition to the above, selects at random ℓ devices to compromise. She is able to determine all the communities (and their pseudonyms) of the compromised devices.

The above adversary model could be extended in the future with attacks that are 'tailored' to each community pseudonym scheme.

5.3.4 Simulation Results

The goal of this section is to study how the different community pseudonym schemes perform with respect to each other under the above attack.

In Fig. 5 (a) and (c), we show the CAN and CUN advantages in the case of a probabilistic adversary. We observe that the adversary's advantage usually increases with the number of collected messages s. The fixed single pseudonym scheme (*Single Pseudonym Fixed*) does not provide CUN because the adversary can trivially link together broadcast community pseudonyms. In contrast, the single pseudonym scheme similar to linkable secret handshakes (*Single Pseudonym SH*) results in a low advantage because community pseudonyms broadcasted by mobile devices are always the same for a given user and different from others. Linkable secret handshakes schemes protect community privacy. Nevertheless, $\mathcal{A}$ trivially tracks users' whereabouts, thus jeopardizing location privacy.

The multiple pseudonym scheme over the entire domain results in the lowest advantage because pseudonyms are rarely reused (i.e., M is large) and provides location privacy. As soon as M decreases (*Multiple Pseudonym Shrunk* with $h = 0.01$), the advantage increases considerably, i.e., reusing community pseudonyms reduces community privacy, as $\mathcal{A}$ can then correlate different messages.

Hints attenuate the negative effect of a shrunk community pseudonym set. We implement Hints with the shrunk scheme $h = 0.01$ and by selecting community pseudonyms with repetitions. Hints have lower advantage than the shrunk scheme as they introduce confusion: community pseudonyms are reused for different purposes. Hence, Hints extend the lifetime of shrunk sets of pseudonyms.

The *k-anonymous* scheme complements the *single pseudonym SH* scheme by selecting $\mathsf{k}-1$ other community pseudonyms [48]. Extra community pseudonyms are chosen from communities the sender does not belong to (e.g., pseudonyms eavesdropped in previous interactions). We observe that the k-anonymous scheme with $\mathsf{k} = 3$ performs worse than the *single pseudonym SH* scheme. The graph $\mathbf{G_e}$ (Fig. 6) shows that $\mathcal{A}$ can distinguish communities because the $\mathsf{k}-1$ community pseudonyms leak additional information: $\mathcal{A}$ learns that groups of pseudonyms do not belong to the same community. With a k-anonymous scheme, the advantage even increases faster than the shrunk domain approach. Hence, k-anonymous schemes are similar to raw single pseudonym SH schemes and do not provide increased privacy.

In Figures 5 (b) and (d), we show the advantage of the deterministic adversary with respect to the fraction of compromised devices ℓ/n averaged across all values of s. We observe that the increase of the advantage is non-linear: even if devices are compromised at random, compromising a fraction of those devices suffices to affect community privacy. $\mathcal{A}$ may target devices that belong to many communities to improve its effectiveness.

The simulation-based advantages can be mapped to numerical results in Fig. 4. Our numerical model allows us to evaluate the performance of the attack and the community pseudonym scheme. For example, our numerical results give minimum/maximum advantage of deterministic adversary for different α'. We can map simulation-based advantage with a value of s and a number of compromised devices l to a point in Fig. 4.

6. DISCUSSION

The notion of community pseudonyms applies to any underlying encryption mechanisms. We assume for simplicity that symmetric keys are centrally generated and distributed to mobile devices. Other distributed solutions such as attributed-based encryption [9] provide easier management of revocation and forward secrecy. Unfortunately, they generally incur higher computation and communication costs. We describe for completeness how to provide revocation and forward secrecy with centrally generated keys. We also discuss the relation between this work and secret handshakes.

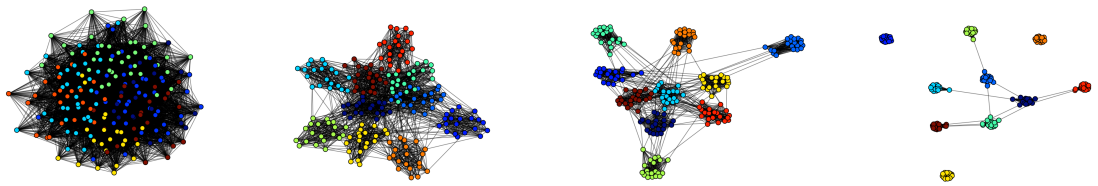
6.1 Forward Secrecy and Revocation

The symmetric key shared by community members can be used as a digital credential to authenticate other community members and encrypt communications. However, such a widely shared secret could leak if one community member is compromised or malicious. Hence, pseudonym schemes using symmetric cryptography do not provide *forward secrecy*: if the secret key of a community is leaked, then all community pseudonyms, even the ones generated before the leak [47], are no longer trustworthy. In addition, an adversary that obtains a community secret can break community privacy by observing the messages broadcast by other users.

For forward secrecy, community pseudonym schemes based on symmetric key cryptography can be modified to *change over time* the shared secret of communities. As investigated in [47], symmetric key updates should be generated in mobile devices in a distributed fashion, e.g., relying on pseudo-random functions. The symmetric key re-keying can also be done relying on the asymmetric credentials of users as described in [37]. Such re-keying operations are also needed when new members join a community or existing members leave the community (e.g., for revocation).

Rekeying operations require coordination among all community members. We argue that the cost of coordinating key updates is lower than the cost of relying on asymmetric cryptography to obtain similar properties. In the case of peer-to-peer wireless networks, symmetric keys of communities could be changed at regular interval as suggested in [43] to minimize costs. In addition, mobile devices can communicate with the CA in order to check whether they are using the correct secret.

The detection of misbehaving community members can be difficult with symmetric cryptography. For example, any community member can broadcast spam messages to other community members, and as the sender is not uniquely authenticated, he may be



(a) (b) (c) (d)

Figure 6: Graph G_e resulting from a traffic analysis attack on a k-anonymous scheme with number of communities $m = 10$, number of communities users belong to $n_c = 3$ and $k = 3$. Nodes are community pseudonyms and edges indicate the possibility that community pseudonyms belong to the same community. The color of a node indicates the most likely community it is inferred to belong to. (a) $t = 1$, (b) $t = 2000$, (c) $t = 5000$ and (d) $t = 10000$.

difficult to identify. This problem can be overcome with the use of digital signatures within the secure channel established with the symmetric key of the community. In the event of a spamming attack, community members can require other members to use their PKI credentials in their messages [45]. This will induce a larger cost on all community members in the region of the network where the spamming attack is taking place. Users now authenticated with their personal credentials can be reported to a central server and revoked by using traditional revocation algorithms [49]. Such mechanism could also detect the presence of Sybil attacks.

6.2 Relation with Secret Handshakes

Community pseudonym schemes achieve similar properties as secret handshake schemes. Notably as in [45], they consider a special type of secret handshake in order to reduce cost: i) for computation cost, symmetric cryptography is used; ii) for communication cost, a single message is required to determine whether a message is destined to a community. In general, results obtained with community pseudonym schemes affect secret handshakes as follow:

- Most secret handshakes use a single pseudonym in the initiation message and are thus linkable. This is similar to single pseudonym schemes. Our results show that linkability defeats the privacy provided by secret handshakes as an adversary can easily infer communities users belong to.
- Previous work suggests heuristics to obtain unlinkable secret handshakes. One solution is to rotate through a small set of pseudonyms. This is similar to the shrunk community pseudonym scheme. Our results show that if the set of pseudonyms is not large enough, the adversary will infer communities that users belong to.
- Previous work also suggests to set strict time limits on the use of each pseudonym. This is similar to the scheme where multiple pseudonyms are used over the entire domain. Our results show that such approach provides privacy as long as the time period over which a given pseudonym is used is kept short.
- Previous work also suggests the use of k-anonymity. Besides significantly increasing cost, our results also indicate that k-anonymous schemes are, at best, detrimental to community privacy.

In other words, our work shows how unlinkable secret handshake heuristics may fail in practice and highlights the need to understand the context in which security primitives are used.

Secret handshakes schemes also assume that dummy messages obfuscate unsuccessful handshakes to the adversary. Hence, an adversary is unable to learn whether an interaction between two devices was successful. Unfortunately, this generates a significant cost because in practice: i) most interactions are unsuccessful; ii) it is not trivial to generate dummy messages that appear legitimate. In this work, as we are cost-averse, we assumed that dummy messages are not used. The adversary is thus able to observe unsuccessful interactions and learn from them. Even if an adversary was unable to do so (i.e., dummy messages are used), it would still learn from interactions because multiple community pseudonyms coming from one device have a higher chance of being from different communities. Hence, the adversary would still break community privacy, albeit with a lower advantage.

7. CONCLUSION

In this paper, we considered the problem of community privacy in peer-to-peer wireless networks and evaluated privacy risks of information sharing within communities in such networks. Identifying the need to protect community privacy, we proposed a framework based on challenge-response games to study it. An interesting outcome of the framework is the analytical relation obtained between community anonymity and community unlinkability. The relation between these two properties was previously studied [40]. To the best of our knowledge, we are the first to analytically relate these properties.

By means of simulations, we evaluated the privacy provided by different pseudonym-based community privacy-preserving schemes. Our results throw light on the relationship between community pseudonym-based and secret handshake schemes: shrinking the number of possible community pseudonyms significantly reduces the achievable privacy. Hence, it is not advisable to cycle through a small set of pseudonyms with secret handshakes. This result illustrates the delicate trade-off between the achievable community privacy and the cost of community pseudonym schemes. Our analysis enables system designers to tune their shrunk scheme to a desired privacy level by, for example, regularly changing the set of community pseudonyms. We also showed that reusing pseudonyms across communities (Hints) can provide a good cost/privacy trade-off and demonstrated that k-anonymous schemes are, at best, detrimental to community privacy. In the future, we intend to investigate other communication models and, by means of practical implementations, study the extra overhead introduced by community pseudonym schemes.

8. REFERENCES

[1] http://en.wikipedia.org/wiki/Bluedating.
[2] http://en.wikipedia.org/wiki/Lovegetty.
[3] Aka aki. http://www.aka-aki.com.

[4] Blue star. http://www.csg.ethz.ch/research/projects/Blue_star.

[5] Game mobile. http://www.gamemobile.co.uk/bluetoothmobilegames.

[6] Social serendipity. http://reality.media.mit.edu/serendipity.php.

[7] M. Arrington, 2007. http://www.techcrunch.com/2007/09/11/the-holy-grail-for-mobile-social-networks.

[8] N. Asokan and P. Ginzboorg. Key agreement in ad-hoc networks. *Computer Communications*, 23:1627–1637, 1999.

[9] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin. Persona: an online social network with user-defined privacy. *ACM SIGCOMM Computer Communication Review*, 39(4):135–146, 2009.

[10] D. Balfanz, G. Durfee, N. Shankar, D. Smetters, J. Staddon, and H.-C. Wong. Secret handshakes from pairing-based key agreements. In *IEEE S & P*, 2003.

[11] A. Barth, D. Boneh, and B. Waters. Privacy in encrypted content distribution using private broadcast encryption. In *FC*, 2006.

[12] A. R. Beresford. Location privacy in ubiquitous computing. Ph.D. thesis, University of Cambridge, 2005.

[13] O. N. Blog. Nokia instant community gets you social, 2010. http://conversations.nokia.com/2010/05/25/nokia-instant-community-gets-you-social.

[14] V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008.

[15] R. W. Bradshaw, J. E. Holt, and K. E. Seamons. Concealing complex policies with hidden credentials. In *CCS*, 2006.

[16] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *MobiCom*, pages 85–97, 1998.

[17] L. Buttyan, T. Holczer, and I. Vajda. On the effectiveness of changing pseudonyms to provide location privacy in VANETs. In *ESAS*, 2007.

[18] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich. How to win the clone wars: efficient periodic n-times anonymous authentication. In *CCS*, 2006.

[19] D. Chaum and E.V.Heyst. Group signatures. In *EUROCRYPT*, 1991.

[20] C.-H. O. Chen, C.-W. Chen, C. Kuo, Y.-H. Lai, J. M. McCune, A. Studer, A. Perrig, B.-Y. Yang, and T.-C. Wu. GAnGS: gather, authenticate 'n group securely. In *MobiCom*, 2008.

[21] N. Eagle, A. S. Pentland, and D. Lazer. Inferring friendship network structure by using mobile phone data. *National Academy of Sciences*, 106(36):15274–15278, 2009.

[22] A. Fiat and M. Naor. Broadcast encryption. In *CRYPTO*, 1994.

[23] M. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *EuroCRYPT*, pages 1–19, 2004.

[24] M. Gruteser and D. Grunwald. Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis. *Mob. Netw. Appl.*, 10(3):315–325, 2005.

[25] L. Huang, K. Matsuura, H. Yamane, and K. Sezaki. Enhancing wireless location privacy using silent period. In *WCNC*, 2005.

[26] S. Jarecki, J. Kim, and G. Tsudik. Beyond secret handshakes: Affiliation-hiding authenticated key exchange. In *CT-RSA*, 2008.

[27] S. Jarecki and X. Liu. Unlinkable secret handshakes and key-private group key management schemes. In *ACNS*, pages 270–287, 2007.

[28] S. Jarecki and X. Liu. Private Mutual Authentication and Conditional Oblivious Transfer. *CRYPTO*, pages 90–107, 2009.

[29] S. Jarecki and X. Liu. Affiliation-hiding envelope and authentication schemes with efficient support for multiple credentials. *Automata, Languages and Programming*, 2010.

[30] M. Khiabani. Metro-sexual, 2009. http://bit.ly/theranMetroSexual.

[31] R. Laroia. Future of wireless? the proximate internet. keynote presentation, 2010. http://www.cedt.iisc.ernet.in/people/kuri/Comsnets/Keynotes/Keynote-Rajiv-Laroia.pdf.

[32] M. Li, K. Sampigethaya, L. Huang, and R. Poovendran. Swing & Swap: user-centric approaches towards maximizing location privacy. In *WPES*, pages 19–28, 2006.

[33] N. Li, W. Du, and D. Boneh. Oblivious signature-based envelope. *Distributed Computing*, 17(4):293–302, 2005.

[34] M. Manulis, B. Pinkas, and B. Poettering. Privacy-Preserving Group Discovery with Linear Complexity. In *ACNS*, pages 420–437, 2010.

[35] S. Nagaraja. The impact of unlinkability on adversarial community detection: Effects and countermeasures. In *PETS*, 2010.

[36] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[37] D. Park, C. Boyd, and S. J. Moon. Forward secrecy and its application to future mobile communications security. In *Public Key Cryptography*, pages 433–445, 2004.

[38] Patently Apple. iGroups: Apple's new iPhone social app in development, 2010. http://www.patentlyapple.com/patently-apple/2010/03/igroups-apples-new-iphone-social-app-in-development.html.

[39] E. Paulos and E. Goodman. The familiar stranger: anxiety, comfort, and play in public places. In *CHI*, pages 223–230, 2004.

[40] A. Pfitzmann and M. Kohntopp. Anonymity, unobservability, and pseudonymity - a proposal for terminology. In *Workshop on Design Issues in Anonymity and Unobservability*, 2001.

[41] R. Putnam. Bowling alone: America's declining social capital. *Journal of democracy*, 6(1):65–78, 1995.

[42] R. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACrypt*, 2001.

[43] S. Setia, S. Koussih, S. Jajodia, and E. Harder. Kronos: A scalable group re-keying approach for secure multicast. In *IEEE S&P*, 2002.

[44] M. Steiner, G. Tsudik, and M. Waidner. Key agreement in dynamic peer groups. *Transactions on Parallel and Distributed Systems*, 2000.

[45] G. Tsudik and S. Xu. A flexible framework for secret handshakes. In *PETS*, 2006.

[46] M. Vojnovic and J.-Y. L. Boudec. Perfect simulation and stationarity of a class of mobility models. In *Infocom*, 2005.

[47] S. Xu. On the security of group communication schemes based on symmetric key cryptosystems. In *SASN*, pages 22–31, 2005.

[48] S. Xu and M. Yung. K-anonymous secret handshakes with reusable credentials. In *CCS*, 2004.

[49] P. Zheng. Tradeoffs in certificate revocation schemes. *SIGCOMM Comput. Commun. Rev.*, 33(2):103–112, 2003.

APPENDIX

A. PROOF OF THEOREM 1

PROOF. We prove the first part of Theorem 1 by showing that ability to breach community anonymity (*CAN*) implies ability to breach community unlinkability (*CUN*) as well.

Hence, let an arbitrarily chosen algorithm for breaching community anonymity be $A_{CAN}(p_j, C_i)$. The algorithm outputs *yes* if $p_j \in C_i$ and *no* if $p_j \notin C_i$. We have for some communities C_i (that do not belong to the graph G'):

$$\sigma \quad = \quad Pr(A_{CAN}(p_j, C_i) \text{ is correct}) > \frac{1}{2}$$

Given A_{CAN}, we can now construct a probabilistic algorithm, $A_{CUN}(p_j, p_k)$, for deciding whether any two community pseudonyms belong to the same community or not:

1. Given community pseudonyms p_j and p_k each of which belong to either a community C_0 or to a community C_1.
2. Call $A_{CAN}(p_j, C_0)$ and guess if $p_j \in C_0$.
3. Call $A_{CAN}(p_k, C_0)$ and guess if $p_k \in C_0$.
4. Output yes if the two guesses both say yes or both say no, else output no.

The probability of success of $A_{CUN}(p_j, p_k)$ is $\mu = \sigma^2 + (1-\sigma)^2$ where σ^2 corresponds to the case A_{CAN} guesses both p_j and p_k correctly, and $(1-\sigma)^2$ corresponds to the case where A_{CAN} does not guess either p_j or p_k correctly (but the final answer still is correct).

We observe that when $\sigma = 0.5$, we have $\mu = 0.5$, when $\sigma > 0.5$, we have $\mu > 0.5$ and when $\sigma = 1$, we have $\mu = 1$. Hence, regardless of how the challenger chooses C_0 and C_1, we obtain that A_{CUN} succeeds with probability greater than a random guess. This completes the first part of the proof.

We prove the second part by giving an example of a pseudonym scheme that has the property of CAN but not the property of CUN. We consider a scheme where every community is given a single community pseudonym. This kind of scheme was introduced in section 4.1. Within a community, all users share the same pseudonym which has been chosen randomly. Consequently, community messages are trivially linkable, hence we do not have CUN. On the other hand, an adversary cannot break anonymity because it does not know how to relate pseudonyms to communities. Hence, there is CAN.

□