GeoPal: Friend Spam Detection in Social Networks Using Private Location Proofs

Bogdan Carbunar
FIU
Miami, FL
Email: carbunar@gmail.com

Mizanur Rahman
FIU
Miami, FL
Email: mrahm031@fiu.edu

Mozhgan Azimpourkivi
FIU
Miami, FL
Email: mojganaz@cs.fiu.edu

Debra Davis
FIU
Miami, FL
Email: dledavis@cs.fiu.com

***Abstract*—Friend spam, adversarial invitations sent to social network users, exposes victims to a suite of privacy, spear phishing and malware vulnerabilities. In this paper, we use the location history of users to detect friend spam. We posit that the user trust in friends is associated with their co-location frequency. We exploit this hypothesis to introduce GeoPal, a framework that carefully accesses the potentially sensitive location history of users to privately prove their past location claims, and to privately compute and update fuzzy co-location affinities with other users. We build GeoPal on PLP, a protocol we develop to privately collect proofs of user past locations.**

We confirm our hypothesis through a user study with 68 participants: 57% and 70% of the friends never met in person are not remembered and are not talked to, respectively, by the participants. In contrast, 86% of the friends met daily or weekly are either family, close or regular friends. We highlight the relevance of friend spam: 75% of the participants have at least one friend whom they do not recall. We show that GeoPal is practical: a Nexus 5 can process more thank 20K location proofs per second.

I. Introduction

Friend spam attacks [1]–[3] are friend invitations sent by attackers to the social network accounts of victims. Once accepted, the invitations enable the attackers to collect private information from the accounts of victims (including profiles, locations visited, friend lists), and perform subsequent attacks such as spear phishing [2] and malware dissemination [4], [5].

Friend spam is effective. In a user study with 68 participants, we observed that 75% (51) of the participants have declared not to remember at least one of their 20 randomly selected friends (see § V). While this percentage is likely to be larger when considering all the friends, it is consistent with previous work that showed that 47 to 77% of social network users accept invitations from strangers [6]–[8].

Our user study also reveals that victim naivety plays a part in the success of friend spam: only 38% and 37% of the participants are uncomfortable or very uncomfortable with accepting invitations from "anyone who is attractive" and from "anyone who is my age", respectively (see § V). Social network profile information (e.g., age, photos, name) is however easy to fabricate. In addition, social network mechanisms that encourage users to accept as friends, people with whom they share friends, can be exploited by adversaries to tailor their accounts and improve the success rate of their friend spam [7].

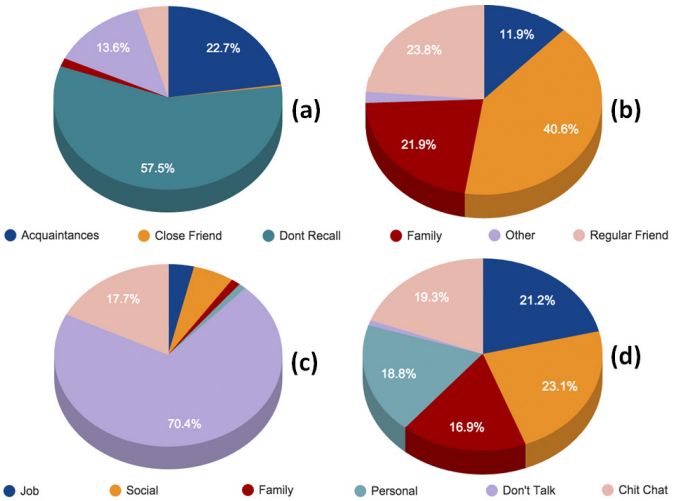


Fig. 1. Distribution of types of friends for (a) friends never met in person and (b) friends met daily or weekly, and of the topics of discussion for (c) friends never met in person and (d) friends met daily or weekly.

In this paper, we posit the existence of a relationship between social network trust and the frequency of physical co-location. Specifically, we conjecture that users tend to trust more the friends whom they have met or are meeting more frequently in person. To evaluate this hypothesis, we have developed GP.Quest, a mobile app questionnaire where users need to classify Facebook friends according to co-location and trust dimensions. Our study (with 68 participants) shows that 93.2% of the Facebook friends that were never met in person are either not remembered, considered to be acquaintances or non-friends (see Figure 1(a)). In contrast, 85.7% of the Facebook friends met daily or weekly are either family, close or regular friends (see Figure 1(b)). Furthermore, participants do not talk or only chit-chat with 88.1% of the friends they never met (see Figure 1(c)). However, the topics of discussion with 80% of the friends met daily or weekly involve family, personal, job and social matters (see Figure 1(d)).

We exploit this result, and the observation that physical co-location with victims is hard to engineer by online adversaries, to introduce GeoPal, a user transparent, location based, friend spam detection framework. The mobile devices of GeoPal users record locations visited throughout the day. GeoPal leverages this "location history" to detect co-location events and infer trust in social network friends.

The use of locations raises however privacy concerns. Out of 68 participants, only 1 said that they would make their location information public (see § V). The indiscriminate sharing of location information with inviters in social networks may thus defeat the very goal of a friend spam detection solution. In addition, revealing sensitive information (e.g., location history, friends) to social network providers further exposes users to significant risks, as providers have been shown to leak [9] and even sell [10] user data to third parties.

Thus, the challenge is to infer trust based on locations, without revealing sensitive information to potential adversaries. To address this challenge, we first propose PLP, a private location proof protocol. In PLP, the social network provider issues proofs of location to mobile device users, without being able to link users to their locations. Acquired proofs are stored on the user's mobile device, and are not shared with the provider. Second, we introduce GeoPal, a set of protocols that enable users to use the recorded proofs to prove past location claims with a desired level of privacy (or precision), and to privately compute and update co-location affinities with other users.

PLP and GeoPal essentially generate and process "fuzzy" location proofs. They enable a user to prove for instance to have lived in a city, by revealing location proofs recorded at a city church, with only zip code level precision. They further support imprecise co-location events, e.g., between users present at neighboring venues within a time frame. To achieve this, we introduce the notion of spatial and temporal *confusion zones*, nested regions of set dimensions with random coordinates, built around the user's location and time of presence. In addition, we propose the notion of *presence tokens*, pseudo-random values generated and updated periodically for supported venues, and revealed only to users who can prove their presence. In summary, we provide the following contributions:

- **Trust vs. co-location frequency dependence**. Conjecture dependence between the trust people have in social networking friends and their co-location frequency.
- **GeoPal and PLP**. Introduce GeoPal, a framework that exploits this conjecture to detect friend spam with privacy [§ IV-C]. GeoPal builds on PLP, a protocol we propose to privately generate location proofs [§ IV-B].
- **Validation**. Verify the conjectured dependence through a user study for GP.Quest, a mobile app that collects trust in and co-location frequency with Facebook friends [§ V]. Show that on a Nexus 5 smartphone, GeoPal can process more than 20,000 location proofs per second [§ VI].

Limitation. GeoPal and PLP do no protect against attackers who are able to prove co-location with the victim. They do however raise the bar, as online stalkers have to also become physical stalkers. Specifically, attackers need to both guess the locations frequented by their victims, and to either frequent them or fabricate proofs of presence, see § IV-D.

II. Related Work

Friend spam has been shown to be a source of significant vulnerabilities [1], [2], [5], [11]. When studying the wall posts of 3.5 million Facebook users Gao et al. [2] discovered more than 200K malicious posts with embedded URLs, with more than 70% pointing to phishing sites. Stringhini et al. [1] used "honey" profiles on 3 social network sites to study the behavior of contacting spammers. Brown et al. [11] found that in Facebook, attackers could send sophisticated context-aware email to approximately 85% of users, including to people with private profiles. The Koobface botnet [4] leveraged social network zombies to generate accounts, befriend victims, and send malware spam. Thomas and Nicol [5] showed that social networks are slow to respond to such threats, leading to 81% of vulnerable users to click on Koobface spam.

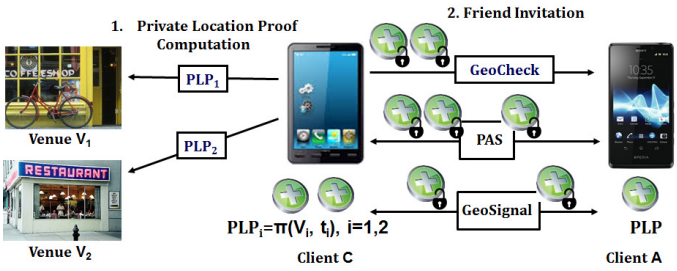


Fig. 2. GeoPal architecture. The user's mobile device privately captures and stores proofs of locations visited. The collected proof history is stored on the device and used to process friend invitations: privately prove past locations, determine fuzzy co-location affinity with invited friends, and detect current co-location with pending friends.

Cao et al. [12] proposed to detect the fake accounts behind friend spam, by extending the Kernighan-Lin heuristic to partition the social graph into two regions, that minimize the aggregate acceptance rate of friend requests from one region to the other. Wu et al. [13] utilized the posting relations between users and messages to combine social spammer and spam message detection. They extracted the social relations between users and the connections between messages, and used them as regularization terms over prediction results. The approaches of [12], [13] leverage access to large social graphs. In contrast, GeoPal makes local decisions, and leverages only information made available to a user during friend invitations.

Querica and Hailes [14] used a hybrid friend and location based approach to defend collaborative, mobile app users against Sybil attacks: maintain and use information about friendly and suspicious devices that the user encounters in time, to decide if an individual is launching a Sybil attack against it. A different flavor of friend spam attacks was introduced by Huber et al. [15], who exploit the unprotected communications between users and social networks.

Our Differences. Our work is the first to provide the experimental foundation for using co-location as a friend trust predictor. In addition, GeoPal and PLP are the first solutions that focus on the privacy dimension of friend spam detection.

III. System & Adversary Model

A. System Model

We consider an ecosystem that consists of a social network provider and a set of subscribed users. Similar to systems like Facebook and Yelp [16], we assume the provider, denoted by S in the following, offers both social and geosocial services.

The users. Each subscribed user has a social networking account. We assume subscribers have mobile devices equipped with a GPS receiver and a Wi-Fi interface (present on most smartphones). Users install and run on their devices a mobile app, that we call "client". Subscribed users receive initial service credentials, including a unique user id; let Id_A denote the id of user A.

Check-in. Users can record their locations either explicitly, e.g., by performing a a "check-in" operation provided by services like Foursquare [17] or Yelp [16], or implicitly, where their mobile device clients periodically record their location. In the following, we use the term "check-in" to denote both forms of location recording operations.

The provider. We build our work on semi-distributed, online social networks [18]–[20]. Specifically, each client locally stores and maintains its user's data, including profile, friend list, posts, and location history. For fault tolerance (e.g., if the device fails or is lost), the user should use backup storage with consistency guarantees. The centralized social network provider component retains however the following functionality:

- *Subscriber search & root of trust*. The provider maintains an online directory listing current subscribers, their contact information and public key certificates. Directory searches can be performed privately, using an efficient private information retrieval (PIR) technique [21], [22]. In the following, we assume all the communications between user clients and the provider take place over an anonymizer, denoted by Mix, e.g., Tor [23].
- *Private user location validation*. The provider verifies the location claims made by users during check-in operations, e.g., [24], [25]. For instance, [24] provides location verifications while ensuring user anonymity: the provider S can verify the validity of user reported locations/times, without learning the user's identity.
- *Venue maintenance*. The provider maintains data for a set of system supported venues (i.e., businesses in Yelp [16]), see § IV-A for details.

Friend invitations. We call the user that sends an invitation the *inviter* and the user receiving the invitation, the *invitee*. We denote by *private information*, user data (e.g., history of locations, list of friends) that should not be learned by strangers without the user's explicit consent.

Cryptographic tools. We rely on cryptographic hash functions H that are pre-image, second pre-image and collision resistant, as well as a semantically secure public and private key cryptosystem. We also use "one-more-forgery resistant" signatures: an adversary with access to the signatures of k messages of his choice, has only negligible advantage in fabricating a valid signature for a challenge message.

B. Adversary Model

We consider external adversaries that launch friend spam attacks. Such adversaries send friend invitations to specific target users, from accounts they control. We assume that adversaries are able to (i) register and control an arbitrary number of fake user accounts as well as to (ii) reverse engineer, modify and run corrupt, malicious versions of the mobile client. In addition, we assume that the provider S is semi-honest (honest but curious). Specifically, S will run its part of the protocols correctly, however, it will attempt to learn private subscriber information.

Notation	Definition
$Tk_{V,e}$	Presence token for V during epoch e
$Vic(V,e)$	Vicinity set: Tokens of venues near V during e
g	Number of confusion zones
$d_i, t_i, i = 1..g$	Spatial and temporal confusion zone sizes
$\overline{V}, \overline{T}$	Spatial and temporal confusion zones for V
$K_V, K_{Vi}, i = 1..g$	Spatial encryption keys
$K_t, K_{tj}, j = 1..g$	Temporal encryption keys
$\mathcal{E}_V, \mathcal{E}_T$	Encrypted spatial and temporal confusion sets
$1 \leq \alpha, \tau \leq g$	Spatial and temporal precision levels

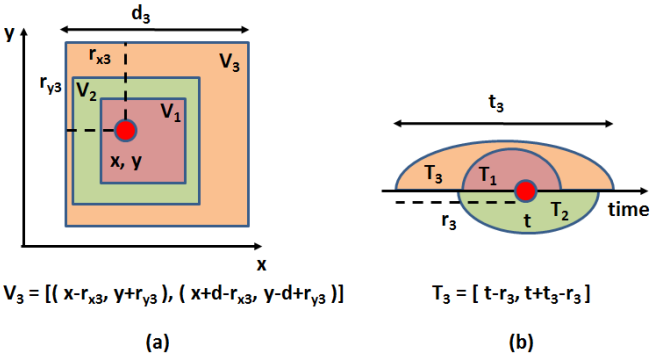


Fig. 3. Illustration of spatial and temporal confusion zones for a check-in point (x, y, t) shown in red (no height dimension considered). The confusion zones are placed at random coordinates around the point.

IV. GeoPal: Location Based Account Validation

We introduce GeoPal, a location based friend invitation verification framework. GeoPal consists of a mobile client, that needs to be installed by subscribed users, and a social network component. In the following, we first describe background definitions and concepts, then introduce the private location proof (PLP) protocol and develop verification protocols for detecting friend spam with privacy assurances, see Figure 2.

A. Background

The provider S generates a public and private key pair. The GeoPal client installed by subscribers stores S's public key and uses it to verify signatures generated by S.

Each client is responsible for storing the location history of its user. Let Π_A denote the location history of a client A. Π_A contains a separate proof for each "check-in" of A. For simplicity of exposition, in the following we focus on two dimensional spatial locations. We now introduce the confusion zone concept.

Confusion zones. Let an integer g, real values $d_1 < .. < d_g$, and real values $t_1 < .. < t_g$ be system parameters. We define the set of spatial confusion zones for a check-in performed at a venue V with location (x, y) at time t, as squares $\overline{V} = \{V_1, .., V_g\}$, of dimensions $d_1, .., d_g$, where $(x, y) \in V_1$ and $V_i \in V_{i+1}$, $\forall i = 1..g-1$. We define the temporal confusion

zones for time t, to be time intervals $\overline{T} = \{T_1, .., T_g\}$ of length $t_1, .., t_g$, where $t \in T_1$ and $T_i \in T_{i+1}$, $\forall i = 1..g-1$.

Figures 3(a) and 3(b) shows examples of spatial and temporal confusion zones, for g=3. To prevent the identification of the (x, y, t) coordinates, confusion zones are not centered at the point (x, y, t).

Presence tokens and vicinity sets. GeoPal divides time into fixed length epochs (e.g., 1 hour, 1 day long). For each supported venue V and during each epoch e, the provider S generates a random *presence token*, $Tk_{V,e}$. S reveals $Tk_{V,e}$ only to clients that are present at V during epoch e. For a venue V at epoch e, let the *vicinity set* $Vic(V, e)$, denote the set containing presence tokens for a pre-defined set of neighboring venues. At the beginning of each epoch e, for each registered venue V, S generates a fresh random presence token, $Tk_{V,e}$, then updates the vicinity set $Vic(V, e)$ with the new presence tokens of all of V's neighboring venues. Table I summarizes our notations.

Fuzzy co-location. We say that a fuzzy co-location event occurs between two users, when the users are located at neighboring venues, within the same epoch.

B. PLP: Private Location Proofs

We present PLP, a private location proof protocol that enables a client C to privately acquire a proof of location at coordinates (x, y) at time t, during epoch e, i.e., $t \in e$. The proofs are then accumulated by the client and used later to demonstrate trustworthiness with invited friends, through GeoPal's protocols (see § IV-C). As previously mentioned, we consider that S has verified C's location, e.g., using [24]. Furthermore, all the communications between C and S take place over an anonymizer (see § III-A).

The protocol works as follows. C generates a fresh random key k and computes a verifiable pseudonym $E_k(Id(C))$. C sends $E_k(Id(C))$, along with the venue identifier V, coordinates (x, y), and time t to S. S performs the following steps:

1) **Generate confusion zones**. Generate the spatial confusion zones $\overline{V}$ =$\{V_1..V_g\}$, according to granularity levels $d_1, .., d_g$. Specifically, for the i-th confusion zone V_i $(i = 1..g)$, generate random $r_{xi} < d_i$ and $r_{yi} < d_i$ values. Then, generate the rectangle V_i defined by its upper left and lower right corners, $V_i = [(x - r_{xi}, y + r_{yi}), (x + d_i - r_{xi}, y - d + r_{yi})]$, see Figure 3. Similarly, generate temporal confusion zones $\overline{T}$ =$\{T_1..T_g\}$ according to granularity levels $t_1, .., t_g$. Specifically, for confusion zone T_i, generate random r_i, then generate $T_i = [t - r_i, t + t_i - r_i]$.
2) **Generate spatial and temporal keys**. Generate fresh, one-time use keys K_V and K_t. Generate a chain of *spatial encryption keys*, $K_{Vi} = H^i(K_V)$, $i = 1..g$ and a chain of *temporal encryption keys* $K_{tj} = H^j(K_t)$, $j = 1..g$. H is a cryptographic hash function and $H^i(M)$ denotes the application of H to M, i times.
3) **Encrypt confusion zones**. Use the spatial chain keys to encrypt the spatial confusion set $\overline{V}$, and produce $\mathcal{E}_V = \{E_{K_{Vi}}(V_i) | i = 1..g\}$. That is, $\mathcal{E}_V$ contains each confusion zone V_i encrypted with the key K_{Vi}. Similarly, use the temporal chain keys to encrypt the temporal confusion set $\overline{T}$, producing $\mathcal{E}_T = \{E_{K_{tj}}(T_j) | j = 1..g\}$.
4) **Generate signature**. Generate signature $\sigma_{V,t} = \{S_S(E_k(Id(C)), \mathcal{E}_V, \mathcal{E}_T)\}$, that binds C's pseudonym to the encrypted confusion sets.
5) **Generate location proof**. Generate

$$\pi(V, t) = (E_k(Id(C)), V, t, e, Tk_{V,e}, Vic(V, e), k, K_V, K_t, \overline{V}, \overline{T}, \mathcal{E}_V, \mathcal{E}_T, \sigma_{V,t}).$$

Send $\pi(V, t)$ to C. C adds $\pi(V, t)$ to its location proof history: $\Pi_C = \Pi_C \cup \pi(V, t)$.

C. PLP Based Account Validation

We now introduce GeoCheck, PFAS and GeoSignal, protocols that build on PLP to validate the location information of social network accounts. $GeoCheck$ uses the confusion sets of a user's location proofs to validate the user's location claims, while protecting the user's spatial and temporal privacy. The PFAS (Private Fuzzy Affinity Score) protocol uses the PLP's location tokens and vicinity sets to privately and distributively determine the past co-location frequency of GeoPal users. The GeoSignal protocol enables users to update their co-location affinity, in real time.

GeoCheck: Privacy Preserving Past Location Verification. $GeoCheck$ allows a client C to selectively reveal information about it's user's past locations (e.g., places where C grew up, went to school, lives, works), while retaining a desired level of privacy. $GeoCheck$ takes as input the user's accepted spatial ($\alpha \in \{1..g\}$) and temporal ($\tau \in \{1..g\}$) privacy levels. α and τ can be negotiated by the client and the invited friend.

Let (V, t) be a location and time pair that C seeks to reveal, with precision levels α and τ. Let $\pi(V, t) \in \Pi_C$ be the corresponding location proof of C. Thus, $\pi(V, t)$ = $(E_k(Id(B))$, V, t, e, $Tk_{V,e}$, $Vic(V, e)$, k, K_V, K_t, $\overline{V}$, $\overline{T}$, $\mathcal{E}_V$, $\mathcal{E}_T$, $\sigma_{V,t})$. $GeoCheck$ proceeds as follows.

C computes $K_{V\alpha} = H^\alpha(K_V)$ and $K_{t\tau} = H^\tau(K_t)$, the keys that will enable the decryption of confusion zones V_α of V and T_τ of T that have the desired α and τ precision levels. C then sends $E_k(Id(C))$, k, $K_{V\alpha}$, $K_{t\tau}$, $\mathcal{E}_V$, $\mathcal{E}_T$ and $\sigma_{V,t}$ to the invitee, who then performs the following verifications:

- Use $\sigma_{V,t} = \{S_S(E_k(Id(B)), \mathcal{E}_V, \mathcal{E}_T)\}$ and the key k, to verify that S's signature binds C's pseudonym $E_k(Id(C))$ to the encrypted confusion sets $\mathcal{E}_V$ and $\mathcal{E}_T$.
- Use the key $K_{V\alpha}$ to decrypt the α-th entry from $\mathcal{E}_V$. Use the key $K_{t\tau}$ to decrypt the τ-th entry from $\mathcal{E}_V$. Verify that the resulting V_α and T_τ are within the space and time window claimed by C in its profile.

If either verification fails, the protocol is aborted.

PFAS: Private Fuzzy Affinity Score. PFAS enables an inviter C to reveal to an invitee I, their "fuzzy affinity" score: their number of past fuzzy co-location events. PFAS provides privacy assurance to both I and C: C does not learn anything from the process, while I only learns the affinity score, but not the locations visited by C or details of the co-location events. The PFAS protocol consists of the following steps.

Let Π_C and Π_I denote the location proof history sets of the inviter and the invitee, respectively. C and I agree on a random blinding factor u (using a pre-commitment step). The inviter C initializes an empty set P. For each proof $\pi(V,t) \in \Pi_C$, C retrieves the vicinity set $Vic(V,e)$. For each token $Tk_{V',e} \in Vic(V,e)$ corresponding to a venue V' in the vicinity of V, C computes the value $H(u, Tk_{V',e})$ and inserts it in the set P (without duplicates).

C generates a value r uniformly at random from $[1..max_r]$, where max_r is a system parameter. If P has less then r elements, C generates $r - |P|$ random values (of the same bit size as the output of the hash function H) and adds them to P. The value r and the padding step are used to hide the total number of check-ins of the inviter C from the invitee I. C randomly permutes the set P and sends it to I.

For each entry in its location proof history $\pi(V,t) = \Pi_I$, the invitee I retrieves the presence token $Tk_{V,e}$ and computes $H(u, Tk_{V,e})$. The affinity score for I and C is equal to the number of such values $H(u, Tk_{V,e})$ contained in P.

GeoSignal: Private co-location signals. We introduce the concept of *probation* friend lists: Each client C maintains a list of accounts that have sent a friend invitation to C's user, but who have a low co-location affinity score with C. "Probation" friends can be provided restricted access to the sensitive information in C's account.

The GeoSignal procedure leverages location proofs to enable C to adjust its co-location affinity score with its probation friends, as well as with friends that have C in their probation lists (i.e., users who C has invited). Specifically, for each of its probation friends, the client C generates a public and private key pair, and shares the public key with the probation friend. GeoSignal is executed by C once per epoch, after C obtains the proof $\pi(V,t)$ for its current location V at time t in epoch e. GeoSignal uses the presence token $Tk_{V,e}$ from $\pi(V,t)$. The communications take place over an anonymizer. C performs the following steps:

1) Turn on the Wi-Fi network interface and set up an ad hoc network with a randomly generated SSID, $ssid_C$.
2) Notify friends. Use $Tk_{V,e}$ to produce a symmetric key, e.g., $K = H(Tk_{V,e})$. Send to each friend A whose probation list contains C, the message $M = E_K(Id(C), V, T_c, ssid_C)$. T_c denotes the current time.
3) Detect co-location with probation friends. From each of its currently active probation friend B, C receives a message M (see step 2 above). C uses the key $K = H(Tk_{V,e})$ of its current location to decrypt M. It then verifies that the result contains $Id(B)$, followed by the location and current time, and a random value $ssid_B$. It then verifies that a Wi-Fi network with this id is locally accessible. If either verification fails, C discards the message. Otherwise, it increments the affinity score of B. If the score exceeds a desired threshold, C promotes B to a full friend status.

D. Analysis

In the following, we consider an adversary $\mathcal{A}$ that controls the provider S and a set of users, including a user B. $\mathcal{A}$ interacts with a challenger $\mathcal{C}$ that controls a user C.

Privacy of the invitee. If $\mathcal{A}$ and $\mathcal{C}$ run GeoPal through the users they control, B and C respectively, then $\mathcal{A}$ does not learn location information from C. To see why this is the case, we observe that during GeoCheck and PFAS, the flow of information is just the reverse, and $\mathcal{A}$ only learns the result of the protocols (i.e., success or failure). During GeoSignal, $\mathcal{A}$ sends B's location encrypted with material generated from B's location. C can decrypt this data only if co-located with B. $\mathcal{A}$ cannot infer C's location since C's location is encrypted with keys that are not available to $\mathcal{A}$.

Correctness of the inviter. GeoPal ensures the correctness of the inviter B. Specifically, the PLP protocol ensures that clients cannot obtain presence tokens for venues they have not visited. This, coupled with the "one-more-forgery" property of the signature of S on $\sigma_{V,t}$ enables C to verify that (i) B has been at location V at time t and that (ii) the sets $\mathcal{E}_V$ and $\mathcal{E}_T$ are bound to B's identity.

In particular, during GeoCheck, C verifies first the signature of S on each of B's revealed H_B entries. Second, it verifies that the encrypted sets $\mathcal{E}_V$ and $\mathcal{E}_T$ were generated for B. The encrypted sets $\mathcal{E}_V$ and $\mathcal{E}_T$, along with the keys $K_{V\alpha}$ and $K_{t\tau}$ enable C to verify B's location with a pre-defined spatial and temporal precision.

During PFAS, B cannot fraudulently boost its location affinity with C: the random presence tokens can only be known by either party if they checked-in at the corresponding locations. Similarly, during GeoSignal, the presence tokens $Tk_{V,e}$ prevent B from claiming a fake current location. In addition, the detection of B's ssid enables C to verify its co-location with B.

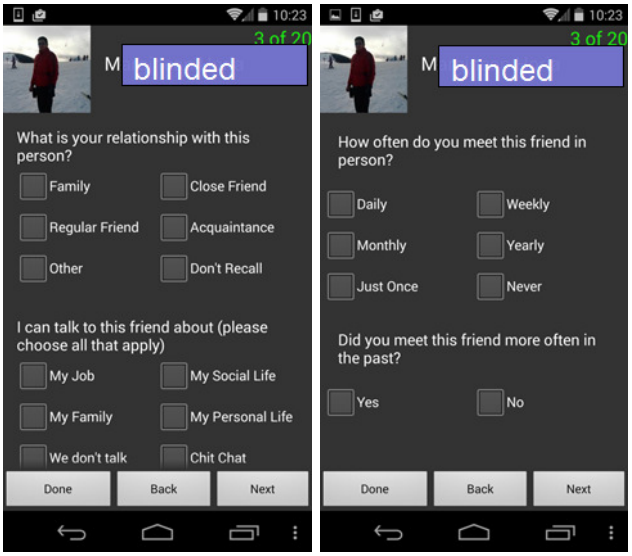
Friend spam protection. GeoPal protects against friend spam: An inviter that does not know the invitee will need to both guess the locations frequented by the invitee, and also be present at those locations during the same epochs with the invitee. GeoCheck fails for inviters that cannot prove their past location claims. PFAS fails for inviters with insufficient past locations in common with the invited user. GeoSignal promotes a probation friend only if the friend can prove real time co-location that exceeds a threshold value.

V. User Study: Trust vs. Co-Location

We now introduce the tool we have developed to evaluate our conjecture of a relationship between trust and co-location frequency, then detail a user study we performed, and present its results.

A. GP.Quest

We have built GP.Quest, a mobile app designed to deliver a set of questions that capture the user's trust and co-location frequency with friends. GP.Quest requires users to login using their Facebook credentials. GP.Quest uses Facebook's mobile API to retrieve information from a random subset of



(a) (b)

Fig. 4. Snapshots of GP.Quest questions designed to capture (a) the user's trust in the friend, and (b) co-location events and habits with the friend.

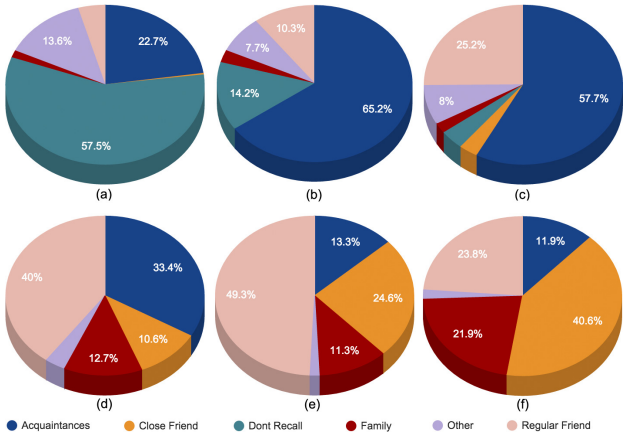


Fig. 5. Pie charts showing the distribution of the type of relationships with Facebook friends, categorized by co-location patterns with those friends that (a) were never met in person, (b) were only met once, (c) are no longer met in person, (d) are met yearly, (e) are met monthly, and (f) are met daily or weekly. We observe that an increase in the frequency of co-location generates a decrease in the the percentage of "don't recall" friends, and an increase in the percentage of "family" and "close friends".

friends (name and thumbnail photo). For each selected friend, GP.Quest presents the user with 4 questions, organized in 2 screens, as illustrated in Figure 4.

Specifically, two questions seek to capture the user's trust in the friend. In the first question, the user needs to select the relationship with the friend, that can be exactly one of "Family", "Close Friend", "Regular Friend", "Acquaintance", "Other" and "Don't Recall". In the second question, the user needs to select discussion topics that are possible with the friend, from among "Job", "Social Life", "Family", "Personal Life", "We Don't Talk" and "Chit Chat". The user can select any number of topics to answer the second question.

The remaining 2 questions seek to extract the user's co-location frequency with the friend. First, the user needs to describe the present co-location frequency. The possible answers are presented in decreasing order of frequency, "Daily", "Weekly", Monthly", "Yearly", "Just Once", and "Never". The second question determines if the user has met this friend more frequently in the past. In both questions, the user can only select one answer.

B. Questionnaire

We have developed a questionnaire to determine background information of the participants. In addition to gender and age, the questions include the type of device used by participants to connect to Facebook, the location, frequency and duration of their Facebook access, reasons for using Facebook, the types of information they feel comfortable sharing, and the people from whom they feel comfortable accepting friend invitations.

C. Procedure and participants

Each participant used GP.Quest, implemented as an Android app, on a Nexus 5 device. Following the GP.Quest session, the participant filled out the above questionnaire. We have recruited participants through class and e-mail announcements, as well as campus ads. Of the 69 recruited participants, 68 (57 male, 11 female; 18-50 years old, M=21, SD=4.91) completed the user study session.

Ethical considerations. We have worked with the Institutional Review Board (protocol number IRB-14-0168) at FIU to ensure an ethical interaction with the participants.

D. GP.Quest Study Results

We have used the data collected from the GP.Quest user study to investigate the relationship between friend co-location frequency and trust. In the following, we focus separately on two dimensions of trust, (i) the type of friend relationship declared by the user, and (ii) the topics of discussion considered by the user with the friend.

Co-Location frequency vs. friend relationship. Figure 5 shows the pie chart friend relationship distributions for friends categorized according to the frequency of co-location. Specifically, Figure 5(a) shows the relations declared for friends that were never met in person. A majority of such friends were either reported as "don't recall" (57%), "other" (13%) or "acquaintance" (23%), while only (6.2%) of such friends were reported as "family", "close friend" or "regular friend".

Figure 5(b) shows the relationship distribution for friends that were only met once. The distribution already differs, with "acquaintance" being the most popular relationship (65.2%), followed by 'don't recall" (14.2%) and "other" (7.7%). Figure 5(c) shows the plot for friends no longer met in person, but who used to be met more in the past. We observe a significantly smaller percentage of such friends labeled as "don't recall". Again, a majority are labeled as either "acquaintance" (58%) or "other" (8%). However, we notice that the participants did not specify the relationship for 25% of these friends.

The distributions of the infrequently met friends are in sharp contrast with friends met yearly, monthly, weekly or daily. 63.5% of the friends met yearly, see Figure 5(d), are

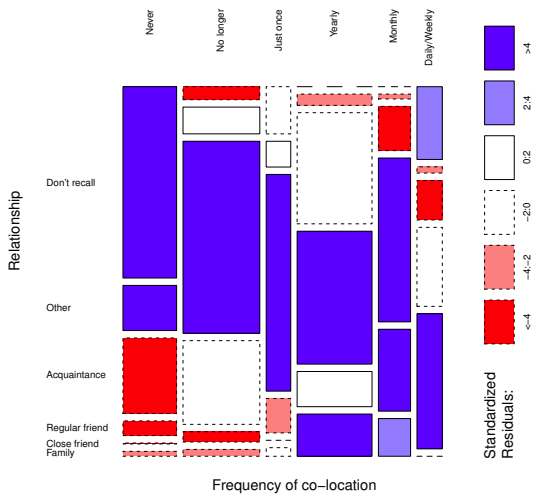


Fig. 6. Mosaic plot showing the relation between co-location and friend relationships. The size of a rectangle denotes the probability of the corresponding co-location frequency and the conditional probability of the corresponding friend relationship. Pearson's chi^2 test reveals that co-location frequency and friend relationship types are not independent.

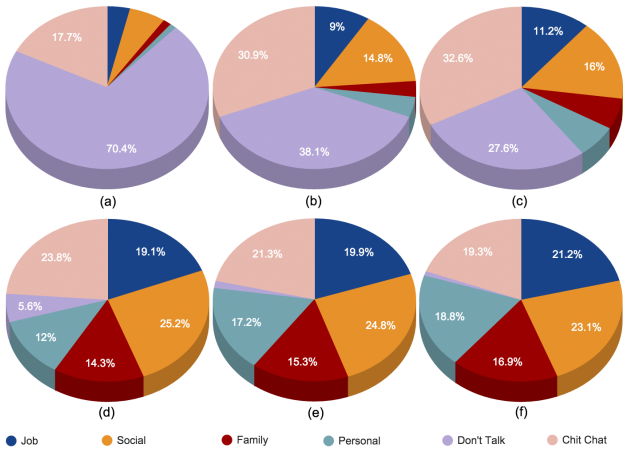


Fig. 8. Pie charts showing the distribution of the discussion topic types considered with Facebook friends (shown as colored pie sectors), categorized by co-location patterns with those friends that (a) were never met in person, (b) were only met once, (c) are no longer met in person, (d) are met yearly, (e) are met monthly, and (f) are met daily or weekly. An increase in the frequency of co-location generates a decrease in the percentage of "don't talk" and "chit chat", and a consistent increase in topics that include the participant's job, social life, family and personal issues.

labeled either "family", "close friend" or "regular friend". Similarly, 85.2% and 85.7% of the monthly (Figure 5(e)) and weekly/daily (Figure 5(f)) met friends are labeled as "family", "close" or "regular friends".

Figure 6 further explores the relation between the co-location frequency and friend relationship types, both categorical variables. We have used Pearson's χ^2 test to test the dependency between the two categorical variables [26]. The standard residuals (shown as multiple of standard deviations) indicate the importance of the cell to the χ^2 value. Since the observed level of significance is extremely low (p-value is 2.2×10^{-16}) we reject the null hypothesis and conclude that there exists a dependency between co-location frequency and friend relationships.

Co-Location frequency vs. discussion topics. Figure 7 shows the topics of discussion considered by the participants with their friends, grouped on the friend relationship and co-location frequency categories. We observe that participants tend to not talk or only consider inconsequential topics of discussion with friends never met in person. As the co-location frequency increases, the distribution of the topics of discussion becomes more balanced. We also observe that participants have very few friends met weekly or daily, with whom they do not speak. We also note that the type of relationship influences the topics of discussion, across co-location frequency types. For instance, participants discuss more sensitive topics with close friends, for multiple co-location frequency types. However, as the co-location frequency increases, such sensitive topics become more prevalent, see the evolution from Figure 7(a)-Figure 7(c).

Figure 8 reinforces this result, showing the distribution of the topics of discussion considered with friends, based on their co-location frequencies. Figure 8(a) reveals that participants do not talk to or only chit-chat with 88.1% of the friends they never met. The percentage of friends with whom participants

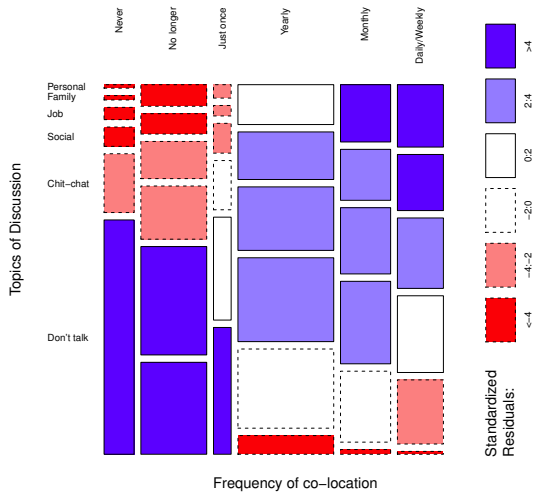
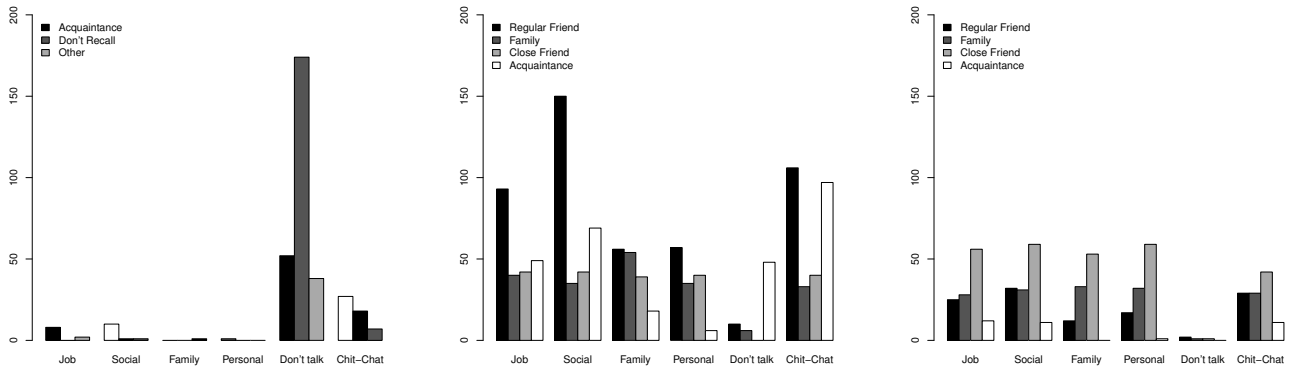


Fig. 9. Mosaic plot showing the relation between co-location and topics of discussion. We observe a dependency between co-location frequency and the quality of the topics of discussion. Pearson's chi^2 test confirms this: the co-location frequency and the discussion topics are not independent.

do not talk decreases significantly as the frequency of co-location increases: participants do not talk to only 0.8% of the friends with whom they meet daily or weekly (see Figure 8(f)). In contrast, the percentage of friends with whom participants consider more sensitive topics of discussion (e.g., "family" and "personal" matters), increases consistently with co-location frequency, from 2.3% for friends never met in person (Figure 8(a)), to 35.7% for friends met daily or weekly (Figure 8(f)).

Figure 9 illustrates the relation between the co-location frequency and discussion topics with friends. Pearson's χ^2 test reveals a close to 0 level of significance (p-value is 2.2×10^{-16}). Thus, we reject the null hypothesis and conclude that there exists a dependency between a participant's co-location frequency and discussion topics with friends.



(a) (b) (c)

Fig. 7. Distribution of topics of discussions, per relationship type, considered with friends (a) never met in person, (b) met yearly, and (c) met weekly or daily. The sensitivity of the topic of discussion increases both with the quality of the friend relationship and the co-location frequency.

Conclusions. We observe a significant relation between frequency of co-location and trust. Pearson's χ^2 test shows a dependency between co-location frequency and both the type of friend relationship and the topics of discussion. Specifically, friends that were never or only infrequently met are mostly not remembered or form lower quality friend relationships (e.g., "other", "acquaintance"). However, frequently met friends form more significant relationships. Similarly, participants mostly either do not communicate or choose shallow topics of discussion with infrequently met friends, but consider more substantial topics of conversation with friends that they frequently meet.

E. Questionnaire Results

Facebook access. 88% (60) of the participants use mobile phones to connect to Facebook, while 75% (51) also use laptops, 48% (33) use desktops, and 35% (24) use tablets. Of the 60 mobile device users, 45% (27) said they use Android while the remaining 55% (33) said they use iPhone devices. In addition, 88% (55) of the participants said that they access Facebook from home, 60% (41) said they do it everywhere, 53% (36) from school, and 13% (9) from public libraries.

73% of the participants said they access Facebook at least every couple of days, with 21% declaring continuous access throughout the day. 93% (63) of the participants said that they spend less than 30 minutes, while only 5% (4) participants said they spend 1-2 hours.

Location service. Only 15% (10) participants responded that their Facebook location service is activated, while 71% (48) said it is not activated, and 15% (10) were not sure. 90% of participants said they know how to activate/deactivate their Facebook location service, while 10% (4) said they do not.

Information sharing. While between $56 - 68\%$ $(37 - 45)$ of participants said they access Facebook in order to keep updated on (each of) family, friends, people known personally, and general events, 17% (11) of the participants said they do it to keep updated on people they only met online. The participants were well aware of their location privacy. When asked with whom do they feel comfortable sharing their current or recent locations, only 1 participant admitted to making this information public, and 1 participant said to use Facebook's default settings. Similarly, 1%, 6%, 15%, 16% and 24% of participants said they make public or use Facebook's defaults for personal information, life events, wall posts, photos, and friend lists, respectively.

Access frequency vs. location sharing. We conjecture that there exists a relationship between the frequency of user access to Facebook and the user's location sharing practices. The likelihood test performed using these categorical variables revealed a p-value of 0.043, thus we conclude that there exists an association between these variables.

Accepting friend invitations. Few people said they would be uncomfortable or very uncomfortable accepting friend invitations from "family" (6%), "friends" (0%), or "acquaintances" (7%). Conversely, only 3% of the participants said to be comfortable or very comfortable accepting "any" friend invitation. However, only 38% (26) and 37% (25) of the participants said they uncomfortable or very uncomfortable with accepting invitations from "anyone who is attractive" and from "anyone who is my age", respectively.

Questionnaire conclusions. The 68 participants are active Facebook users. While most seem to be concerned about their privacy, in particular their sharing practice in Facebook, a surprisingly high percentage of users are vulnerable to friend spam initiated from attractive and similar accounts.

VI. Overhead Evaluation

We have implemented GeoPal in Android and have tested it on (i) an early generation Motorola Milestone smartphone featuring an ARM Cortex A8 CPU @ 600 MHz and 256MB RAM and (ii) a Nexus 5 with a Quad-core 2.3 GHz CPU and 2GB RAM. We have used industrial strength crypto: RSA with 2048 bit keys for signatures, AES for symmetric encryption and SHA-512 for cryptographic hashes.

Figure 10 shows the results of our experiments on the smartphones: the time taken by GeoCheck to verify 1000

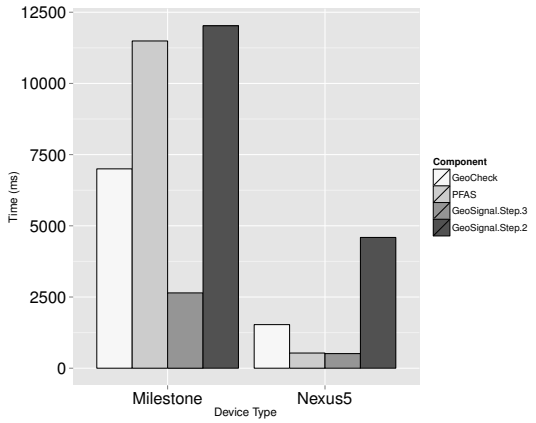


Fig. 10. Smartphone performance for GeoCheck (1000 location proofs), PFAS (10,000 location proofs) and GeoSignal (1000 probation and 1000 pending friends). PFAS takes only 26s on Nexus 5 to process 1 year's worth of location proofs (20K+ proofs per second).

location proofs (1st column for each device), for PFAS to process 10,000 location proofs (2nd column) and for GeoSignal to process 1000 probation friends (3rd column) and 1000 friends in whose probation list it is (4th column). At 1.5ms per location proof (on Nexus 5), GeoCheck imposes a negligible overhead. This is especially the case as in real life users have only a few key locations in their profiles. PFAS is also reasonable: Even if a device collects 1 location proof per minute, a Nexus 5 can process the resulting 525,000 location proofs collected over 1 year, in roughly 26s. GeoSignal's 2nd step (0.5s on Nexus 5) is more efficient than its 3rd step (4.5s for 1000 probation friends), as encryption with the public key is more efficient than decryption.

VII. Conclusions

In this paper we have proposed and validated the hypothesis of association between a social network user's trust in a friend and their co-location frequency. We have leveraged this result to introduce GeoPal, a privacy preserving framework for detecting friend spam attacks. We have built GeoPal on PLP, a protocol that issues proofs of location with privacy. We have shown that GeoPal is efficient, being able to process more than 20K location proofs per second. Future work plans include developing visual warnings that incorporate the outcome of GeoPal verifications, and performing user studies to validate its ability to block friend spam.

VIII. Acknowledgments

This research was supported in part by NSF grants 1450619 and 1527153, and by DoD grant W911NF-13-1-0142.

References

[1] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proceedings of the 26th Annual Computer Security Applications Conference*. ACM, 2010, pp. 1–9.

[2] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao, "Detecting and characterizing social spam campaigns," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 2010, pp. 35–47.

[3] "Report: Iranians Spied On U.S. Officials Using Social Media," Time Magazine, http://time.com/137978/report-iranians-spied-on-u-s-officials-using-social-media/, 2014.

[4] "Koobface," http://en.wikipedia.org/wiki/Koobface.

[5] K. Thomas and D. M. Nicol, "The koobface botnet and the rise of social malware," in *5th International Conference on Malicious and Unwanted Software, MALWARE*, 2010, pp. 63–70.

[6] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda, "All your contacts are belong to us: automated identity theft attacks on social networks," in *Proceedings of the 18th international conference on World wide web (WWW)*, 2009.

[7] R. Potharaju, B. Carbunar, and C. Nita-Rotaru, "ifriendu: leveraging 3-cliques to enhance infiltration attacks in online social networks," New York, NY, USA, pp. 723–725, 2010. [Online]. Available: http://doi.acm.org/10.1145/1866307.1866410

[8] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu, "The socialbot network: When bots socialize for fame and money," in *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2011.

[9] B. Krishnamurthy and C. E. Wills, "On the leakage of personally identifiable information via online social networks," *Computer Communication Review*, vol. 40, no. 1, pp. 112–117, 2010.

[10] E. Steel and G. Fowler, "Facebook in privacy breach," http://online.wsj.com/article/SB10001424052702304772804575558484075236968.html.

[11] G. Brown, T. Howe, M. Ihbe, A. Prakash, and K. Borders, "Social networks and context-aware spam," in *Proceedings of the 2008 ACM conference on Computer supported cooperative work*. ACM, 2008, pp. 403–412.

[12] Q. Cao, M. Sirivianos, X. Yang, , and K. Munagala, "Combating Friend Spam Using Social Rejections," in *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2015.

[13] F. Wu, J. Shu, Y. Huang, and Z. Yuan, "Social spammer and spam message co-detection in microblogging with social context regularization," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, ser. CIKM '15. New York, NY, USA: ACM, 2015, pp. 1601–1610. [Online]. Available: http://doi.acm.org/10.1145/2806416.2806560

[14] D. Quercia and S. Hailes, "Sybil attacks against mobile users: friends and foes to the rescue," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–5.

[15] M. Huber, M. Mulazzani, and E. Weippl, "Who on earth is mr. cypher: automated friend injection attacks on social networking sites," in *Security and Privacy–Silver Linings in the Cloud*. Springer, 2010, pp. 80–89.

[16] "Yelp," http://www.yelp.com.

[17] "Foursquare," https://foursquare.com/.

[18] S. Buchegger, D. Schiöberg, L.-H. Vu, and A. Datta, "Peerson: P2p social networking: early experiences and insights," in *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*. ACM, 2009, pp. 46–52.

[19] A. Cutillo, R. Molva, and T. Strufe, "Safebook: Feasibility of transitive cooperation for privacy on a decentralized social network," in *IEEE WOWMOM*, 2009, pp. 1–6.

[20] J. A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. Epema, M. Reinders, M. R. Van Steen, and H. J. Sips, "Tribler: a social-based peer-to-peer system," *Concurrency and Computation: Practice and Experience*, vol. 20, no. 2, pp. 127–138, 2008.

[21] W. I. Gasarch, "A survey on private information retrieval (column: Computational complexity)," *Bulletin of the EATCS*, vol. 82, pp. 72–107, 2004.

[22] W. Gasarch and A. Yerukhimovich, "Computational Inexpensive PIR (unpublished manuscript)," 2006.

[23] R. Dingledine, N. Mathewson, and P. F. Syverson, "Tor: The second-generation onion router," in *USENIX Security Symposium*, 2004, pp. 303–320.

[24] B. Carbunar and R. Potharaju, "You unlocked the Mt. Everest Badge on Foursquare! Countering Location Fraud in GeoSocial Networks," in *Proceedings of the 9th IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS)*, 2012.

[25] I. Polakis, S. Volanis, E. Athanasopoulos, and E. P. Markatos, "The man who was there: Validating check-ins in location-based services," in *Proceedings of the 29th Annual Computer Security Applications Conference*. ACM, 2013, pp. 19–28.

[26] K. P. F.R.S., "On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," *Philosophical Magazine Series 5*, vol. 50, no. 302, pp. 157–175, 1900.