Toward Preserving Privacy and Functionality in Geosocial Networks

Mahmudur Rahman
Florida International University
mrahm004@cs.fiu.edu

Bogdan Carbunar
Florida International University
carbunar@cs.fiu.edu

Jaime Ballesteros
Florida International University
jball008@cs.fiu.edu

Naphtali Rishe
Florida International University
rishen@cs.fiu.edu

Athanasios V. Vasilakos
National Technical University
of Athens, Greece
vasilako@ath.forthnet.gr

1. INTRODUCTION

Storing user friend lists, preferences and messages, online social networks have become a significant source of sensitive personal information. A recent addition to this space, geosocial networks (GSNs) such as Yelp [1] or Foursquare [2], collect even user locations, through *check-ins* performed by users at visited venues. Overtly, personal information allows GSN providers to offer a variety of applications, including personalized recommendations and targeted advertising, and venue owners to promote their businesses through spatio-temporal incentives (e.g., rewarding frequent customers through accumulated badges). Providing personal information exposes however users to significant risks, as social networks have been shown to leak [3] and even sell [4] user data to third parties. There exists therefore a conflict. Without privacy people may be reluctant to use geosocial networks; without user information the provider and venues cannot support applications and have no incentive to participate.

In this work we take first steps toward breaking this deadlock, by introducing the concept of *location centric profiles* (LCPs), aggregate statistics built from the profiles of users that have visited a certain location. As we know, location privacy has been extensively studied before [5]. This work significantly extends the state of the art by (i) providing constructs that preserve the privacy of users when reporting private profile information (e.g., age, gender, location), and (ii) ensuring that the solutions enable providers to collect information needed to develop existing services. We introduce Profil_R , a framework that allows the construction of LCPs based on the profiles of present users, while ensuring the privacy and correctness of participants. To relieve the GSN provider from costly involvement in venue specific activities, Profil_R stores and builds LCPs at venues.

2. SYSTEM MODEL

We model the geosocial network (GSN) after Yelp [1]. It consists of a provider, S, hosting the system along with information about registered venues, and serving a number of subscribers. To use the provider's services, a client application needs to be downloaded and installed. Users register and receive initial service credentials, including a unique user id. We use the term *client* to denote the software provided by the service and installed by users on their devices.

Participating venue owners need to install inexpensive equipment, present on most recent smartphones. This equipment (solely one-time cost for the venue-owner) can be installed anywhere inside the venue and used for other purposes as well, including detecting fake user check-ins [6] preventing fake badges and incorrect rewards, and validating social network (e.g., Yelp [1]) reviews. We note that location verification solutions that do not rely on venue deployed equipment suffer from lack of ground truth problems (see [6] for a complete discussion of this topic).

2.1 Location Centric Profiles

Each user has a profile $P_U = \{u_1, u_2, .., u_d\}$, consisting of values on d dimensions (e.g., age, gender, home city, etc). Each dimension has a range, or a set of possible values. Given a set of users $\mathcal{U}$ at location L, the *location centric profile* at L, denoted by $LCP(L)$ is the set $\{S_1, S_2, .., S_d\}$, where S_i denotes the aggregate statistics over the i-th dimension of profiles of users from $\mathcal{U}$.

In the following, we focus on a single profile dimension, D. We assume D takes values over a range R that can be discretized into a finite set of sub-intervals (e.g., set of continuous disjoint intervals or discrete values). Then, given an integer b, chosen to be dimension specific, we divide R into b intervals/sets, $R_1, .., R_b$. For instance, gender maps naturally to discrete values ($b = 2$), while age can be divided into disjoint sub-intervals, with a higher b value. We define the aggregate statistics S for dimension D of $LCP(L)$ to consist of b counters $c_1, .., c_b$; c_i records the number of users from $\mathcal{U}$ whose profile value on dimension D falls within range R_i, $i = 1..b$.

3. Profil_R

Let Spotr_V denote the device installed at venue V. For each user profile dimension D, Spotr_V stores a set of *encrypted counters* – one for each sub-range of R. Initially, and following each cycle of k check-ins executed at venue V, Spotr_V initiates *Setup*, to request the provider S to

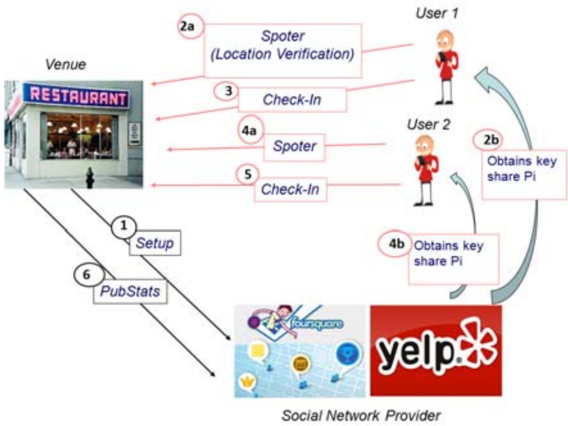


Figure 1: Solution architecture (k=2). The red arrows denote anonymous communication channels, whereas black arrows indicate authenticated (and secure) communication channels.

generate a new Benaloh key pair [7].

When a user U checks-in at venue V, it first engages in the *Spoter* protocol with SPOTR$_V$. This allows the venue to verify U's physical presence through a challenge/response protocol between SPOTR$_V$ and the user device. Furthermore, a successful run of *Spoter* provides U with a share of the secret key employed in the Benaloh cryptosystem of the current cycle. For each venue and user profile dimension, S stores a set Sh of shares of the secret key that have been revealed so far.

Subsequently, U runs $CheckIn$ with SPOTR$_V$, to first send its share of the secret key and to receive the encrypted counter sets. During $CheckIn$, for each dimension D, U increments the counter corresponding to her range, re-encrypts all counters and sends the resulting set to SPOTR$_V$. U and SPOTR$_V$ engage in a zero knowledge protocol that allows SPOTR$_V$ to verify U's correct behavior: exactly one counter has been incremented. SPOTR$_V$ stores the latest, proved to be correct encrypted counter set, and inserts the secret key share into the set Sh. Once k users successfully complete the $CheckIn$ procedure, marking the end of a cycle, SPOTR$_V$ runs $PubStats$ to reconstruct the private key, decrypt all encrypted counters and publish the tally.

3.1 The Solution

Let C_i denote the set of encrypted counters at V, following the i-th user run of $CheckIn$. $C_i = \{C_i[1], .., C_i[b]\}$, where $C_i[j]$ denotes the encrypted counter corresponding to R_j, the j-th sub-range of R. We write $C_i[j] = E(u_j, u'_j, c_j, j)$ $= [E(u_j, c_j), E(u'_j, j)]$, where u_j and u'_j are random obfuscating factors and $E(u, m)$ denotes the Benaloh encryption of message m using random factor u. That is, an encrypted counter is stored for each sub-range of domain R of dimension D. The encrypted counter consists of two records, encoding the number of users whose values on dimension D fall within a particular sub-range of R.

Let $RE(v_j, v'_j, E(u_j, u'_j, c_j, j)$ denote the re-encryption of the j-th record with two random values v_j and v'_j: $RE(v_j, v'_j, E(u_j, u'_j, c_j, j)) = [RE(v_j, E(u_j, c_j)), RE(v'_j, E(u'_j, j))]$ $= [E(u_j v_j, c_j), E(u'_j v'_j, j)]$. Let $C_i[j]++ = E(u_j, u'_j, c_j + 1, j)$ denote the encryption of the incremented j-th counter. Note that incrementing the counter can be done without decrypting $C_i[j]$ or knowing the current counter's value: $C_i[j]++ = [E(u_j, c_j)y, E(u'_j, j)] = [y^{c_j+1}u^r_j, E(u'_j, j)] = [E(u_j, c_j + 1), E(u'_j, j)]$.

In the following we use the above definitions to introduce PROFIL$_R$. PROFIL$_R$ instantiates $PP(k)$, where k is the privacy parameter. The notation $P(A(params_A), B(params_B))$ denotes the fact that protocol P involves participants A and B, each with its own parameters.

Setup(V(),S(k)):. The provider S runs the key generation function $K(k)$ of the Benaloh cryptosystem [7]. Let p and q be the private key and n and y the public key. S sends the public key to SPOTR$_V$. SPOTR$_V$ generates a signature key pair and registers the public key with S. For each user profile dimension D of range R, SPOTR$_V$ performs the following steps:

- Initialize counters $c_1, .., c_b$ to 0. b is the number of R's sub-ranges.
- Generate $C_0 = \{E(x_1, x'_1, c_1, 1), .., E(x_b, x'_b, c_b, b)\}$, where x_i, x'_i, $i = 1..b$ are randomly chosen values. Store C_0 indexed on dimension D.
- Initialize the share set $S_{key} = \emptyset$.

Spoter(U(K),V(),S(k)):. To ensure anonymity, U needs to generate fresh random MAC and IP addresses for each run of Spoter (and CheckIn) with SPOTR$_V$. No advantage can be gained by spoofing MAC and IP addresses. SPOTR$_V$ uses one of the location verification procedures proposed in [6] to verify U's presence. Let U be the i-th user checking-in at V. If the verification succeeds and $i \leq k$, S uses the (k, n) TSS to compute a share of p (Benaloh secret key, factor of the modulus n). Let p_i be the share of p. S sends the (signed) share p_i to U. If $i > k$, S calls *Setup* to generate new parameters for V.

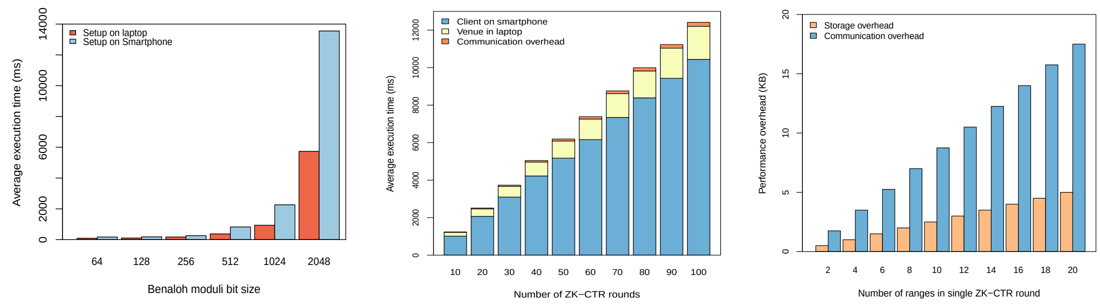
CheckIn(U(p_i, n, V), V(n, y, C_{i-1}, S_{key})). : Executes only if the previous run of *Spoter* is successful. Let U be the i-th user checking-in at V. Then, C_{i-1} is the current set of encrypted counters. SPOTR$_V$ sends C_{i-1} to U. Let v, U's value on dimension D, be within R's j-th sub-range, i.e., $v \in R_j$. U runs the following steps:

- Generate b pairs of random values $\{(v_1, v'_1), .., (v_b, v'_b)\}$. Compute the new encrypted counter set C_i, where the order of the counters in C_i is identical to C_{i-1}: $C_i =$ $\{RE(v_l, v'_l, C_{i-1}[l])|l = 1..b, l \neq j\} \cup RE(v_j, v'_j, C_{i-1}[j]++)$.
- Send C_i along with the signed (by S) share p_i of the private key p to V.

If SPOTR$_V$ successfully verifies the signature of S on the share p_i, U and SPOTR$_V$ engage in a zero knowledge protocol ZK-CTR (see Section 3.2). ZK-CTR allows U to prove that C_i is a correct re-encryption of C_{i-1}: only one counter of C_{i-1} has been incremented. If the proof verifies, SPOTR$_V$ replaces C_{i-1} with C_i and ads the share p_i to the set S_{key}.

PubStats(V(C_k,Sh,V),S(p,q)). : SPOTR$_V$ performs the following actions:

- If $|Sh| < k$, abort.
- If $|Sh| = k$, use the k shares to reconstruct p, the private Benaloh key.
- Use p and $q = n/p$ to decrypt each record in C_k, the final set of counters at V. Publish results.



(a) (b) (c)

Figure 2: (a) *Setup* dependence on Benaloh mod size. (b) ZK-CTR Performance: Dependence on number of proof rounds. (c) Storage and communication overhead (in KB) as a function of range count.

3.2 ZK-CTR: Proof of Correctness

U and SPOTR$_V$ run the following steps s times:

- U generates random values $(t_1, t'_1), .., (t_b, t'_b)$ and random permutation π, then sends to SPOTR$_V$ the proof set $P_{i-1} = \pi\{RE(t_l, t'_l, C_{i-1}[l]), l = 1..b\}$.
- U generates random values $(w_1, w'_1), .., (w_b, w'_b)$, then sends to SPOTR$_V$ the proof set $P_i = \pi\{RE(w_l, w'_l, C_i[l]), l = 1..b\}$
- SPOTR$_V$ generates a random bit a and sends it to U.
- If $a = 0$, U reveals random values $(t_1, t'_1), .., (t_b, t'_b)$ and $(w_1, w'_1), .., (w_b, w'_b)$. SPOTR$_V$ verifies that for each $l = 1..b$, $RE(t_l, t'_l, C_{i-1}[l])$ occurs in P_{i-1} exactly once, and that for each $l = 1..b$, $RE(w_l, w'_l, C_i[l])$ occurs in P_i exactly once.
- If $a = 1$, U reveals $o_l = v_l w_l t_l^{-1}$ and $o'_l = v'_l w'_l t'^{-1}_l$, for all $l = 1..b$ along with j, the position in P_{i-1} and P_i of the incremented counter. SPOTR$_V$ verifies that for all $l = 1..b, l \neq j$, $RE(o_l, o'_l, P_{i-1}[l]) = P_i[l]$ and $RE(o_j, o'_j, P_{i-1}[j]y) = P_i[j]$.
- If any verification fails, SPOTR$_V$ aborts the protocol.

4. EVALUATION

We have implemented PROFIL$_R$ using Android. For secret sharing, we used Shamir's scheme [8] and for digital signatures we used RSA. We have used Android Samsung Admire smartphones (800MHz CPU) and a Dell laptop (2.4GHz Intel Core i3, 4GB of RAM) for the server. For local connectivity the devices used their 802.11b/g Wi-Fi interfaces. We plot averages taken over 10 independent protocol runs.

We have first measured the overhead of the *Setup* operation. We set the number of ranges of the domain D to be 10, Shamir's TSS group size to 1024 bits and RSA's modulus size to 1024 bits. Figure 2(a) shows the *Setup* overhead on the smartphone and laptop platforms, when the Benaloh modulus size ranges from 64 to 2048 bits. Note that even a resource constrained smartphone takes only 2.2s for 1024 bit sizes (0.9s on a laptop). We then measure ZK-CTR's client and SPOTR$_V$ computation and communication overhead. Figure 2(b) shows the overheads of the three costs as a function of the number of ZK-CTR rounds, when the Benaloh key size is 1024 bit long. For 30 rounds, when a cheating client's probability of success is 2^{-30}, the total overhead is 3.6s. Finally, Figure 2(c) shows the SPOTR$_V$ storage overhead, only a fraction of the (single round client-to-SPOTR$_V$) communication overhead. For one dimension, with 20 subranges, the overhead is 5KB.

5. RELATED WORK

Golle et al. [9] proposed techniques allowing pollsters to collect user data while ensuring the privacy of the users. The privacy is proved at "runtime": if the pollster leaks private data, it will be exposed probabilistically. Our work also allow entities to collect private user data, however, the collectors are never allowed direct access to private user data.

Toubiana et. al [10] proposed Adnostic, a privacy preserving ad targeting architecture. Users have a profile that allows the private matching of relevant ads. While PROFIL$_R$ can be used to privately provide location centric targeted ads, its main goal is different - to compute location (venue) centric profiles that preserve the privacy of contributing users.

6. REFERENCES

[1] Yelp. `http://www.yelp.com`.

[2] Foursquare. `https://foursquare.com/`.

[3] Balachander Krishnamurthy and Craig E. Wills. On the leakage of personally identifiable information via online social networks. *Computer Communication Review*, 40(1):112–117, 2010.

[4] Emily Steel and Geoffrey Fowler. Facebook in privacy breach. `http://online.wsj.com/article/SB10001424052702304772804575558484075236968.html?mod=WSJ_hps_LEADNewsCollection`.

[5] M. Humbert, T. Studer, M. Grossglauser, and J-P Hubaux. Nowhere to Hide: Navigating around Privacy in Online Social Networks. In *The Proceedings of 18th ESORICS*, 2013.

[6] 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 MASS'12*, 2012.

[7] Josh Benaloh. Dense probabilistic encryption. In *Proceedings of the Workshop on Selected Areas of Cryptography*, pages 120–128, 1994.

[8] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[9] P. Golle, F. McSherry, and I. Mironov. Data collection with self-enforcing privacy. In *ACM CCS 2006*, pages 69–78. ACM.

[10] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas. Adnostic: Privacy preserving targeted advertising. In *NDSS*, 2010.