Privacy-preserving Spatiotemporal Matching

Jingchao Sun, Rui Zhang, and Yanchao Zhang
School of Electrical, Computer, and Energy Engineering (ECEE)
Arizona State University, Tempe, Arizona, USA
{jcsun, ruizhang, yczhang}@asu.edu

***Abstract*—The explosive growth of mobile-connected and location-aware devices makes it possible to have a new way of establishing trust relationships, which we coin as spatiotemporal matching. In particular, a mobile user could very easily maintain his spatiotemporal profile recording his continuous whereabouts in time, and the level of his spatiotemporal profile matching that of the other user can be translated into the level of trust they two can have in each other. Since spatiotemporal profiles contain very sensitive personal information, privacy-preserving spatiotemporal matching is needed to ensure that as little information as possible about the spatiotemporal profile of either matching participant is disclosed beyond the matching result. We propose a cryptographic solution based on Private Set Intersection Cardinality and a more efficient non-cryptographic solution involving a novel use of the Bloom filter. We thoroughly analyze both solutions and compare their efficacy and efficiency via detailed simulation studies.**

I. Introduction

Mobile-connected devices are penetrating everyday life. According to a recent Cisco report [1], the number of mobile-connected devices such as smartphones, tablets, laptops, and eReaders will exceed the world population in 2012 and hit 10 billion in 2016. The majority of mobile-connected devices are expected to have multiple communication interfaces (cellular, WiFi, Bluetooth, etc.) whereby to conveniently communicate with nearby devices. In addition, they will be location-aware and can always acquire their own precise locations via pre-installed cellular/WiFi/GPS positioning software.

The explosive growth of mobile-connected and location-aware devices makes it possible to have a new way of establishing trust relationships, which we coin as *spatiotemporal matching*. In particular, a mobile user could very easily maintain his *spatiotemporal profile* recording his continuous whereabouts in time, and the level of his spatiotemporal profile matching that of the other user can be translated into the level of trust they two can have in each other. As an example, if Alice and Bob discover via spatiotemporal matching that they often go to the same coffee shop or take the same train in the same period, it is natural for Alice to trust Bob over another person whom she only met once before. Below are some exemplary applications of spatiotemporal matching selected from a potentially long list.

- (Participatory Sensing) Multiple mobile users often jointly perform a sensing task about urban environments in a participatory-sensing application [2]. It is highly advantageous for mobile users to collaborate with others with similar spatiotemporal profiles instead of random persons (e.g., tourists) especially when there are expected rewards commensurate with the quality of sensed data.
- (Ad Targeting) Advertisers may minimize operating costs by sending tailored ads to consumers often appearing at some locations and/or during certain time periods. This can be achieved by comparing the specified spatiotemporal profiles of advertisers with those of potential ad viewers.
- (Missed Connection) A missed connection is an occurrence where two or more people cannot exchange contact information or the information exchanged is lost. An example is two commuters who exchange glances daily when taking the same train. People involved could (re)discover each other by comparing their spatiotemporal profiles via free dedicated missed-connection sites.

Spatiotemporal matching also complements the traditional way of establishing trust relationships by letting involved parties verify others' cryptographic credentials (e.g., public-key certificates) which often only establishes personal identities.

There are two critical requirements for releasing the full potential of spatiotemporal matching. In particular, spatiotemporal profiles contain very sensitive personal information, and incautiously disclosing them to the public may cause severe consequences. For example, if an employer surreptitiously discovers an employee's frequent patronage of night clubs, the employee may get unfair treatment at the workplace; if a thief knows the routine of a target victim, he could break in when the victim will be away for a long time. It is thus crucial to have *privacy-preserving* spatiotemporal matching, which ensures that as little information as possible about the spatiotemporal profile of either matching participant is disclosed beyond the matching result. In addition, spatiotemporal matching may be directly performed on mobile devices and thus needs to be very *efficient* in both communication and computation.

There is no sound solution to privacy-preserving spatiotemporal matching. In particular, the work on privacy-preserving profile matching [3]–[5] targets non-spatiotemporal user profiles such as hobbies or interests, and it is unclear how to extend these schemes for spatiotemporal profiles. In addition, some solutions are available for privacy-preserving proximity test [6], [7] which is to test whether two users are in each other's physical proximity at a certain time point. In contrast, spatiotemporal matching in our definition refers to two users testing their physical proximity for an extended period of time. Directly applying these mechanisms [6], [7] to our problem will be highly inefficient and impractical.

This paper focuses on realizing efficient privacy-preserving spatiotemporal matching. Our essential idea is to let each user

978-1-4673-5946-7/13/$31.00 ©2013 IEEE

periodically record his own locations, each of which is then approximated by a geographic cell index among a large set of predefined ones. A user's spatiotemporal profile can then be defined as a set of 2-tuples with each comprising a time index and the corresponding cell index. Two users engaging in spatiotemporal matching can then check whether their spatiotemporal profiles have more than a threshold number of common 2-tuples in a privacy-preserving fashion. Our contributions in this paper can be summarized as follows.

- We are the first to formulate spatiotemporal matching as a new way of establishing trust relationships and identify critical privacy and efficiency challenges.
- We propose two protocols for privacy-preserving spatiotemporal matching. The first protocol is based on Private Set Intersection Cardinality (PSI-CA) [8] and can ensure nearly perfect spatiotemporal privacy at the cost of possibly large communication and computation overhead. The second protocol involves a novel use of the Bloom filter [9] and enables either party involved in spatiotemporal matching to estimate with tunable accuracy the number of common elements in their spatiotemporal profiles without disclosing too much private information to each other. This protocol incurs much lower communication and computation overhead than the first protocol at the cost of slightly reduced spatiotemporal privacy protection.
- We thoroughly analyze the correctness, accuracy, privacy provision, and overhead of both protocols, and we also compare them via detailed simulations driven by experimental data.

The rest of this paper is organized as follows. Section II presents the problem formulation. Section III illustrates two novel privacy-preserving spatiotemporal matching protocols. Section IV theoretically analyzes the performance of the proposed protocols. Section V evaluates our protocols by detailed numerical and experimental results. Section VI surveys the related work. Section VII concludes this paper.

II. Problem Formulation

A. Problem Statement

We consider a large geographic region such as the NYC metropolitan area with system users as either permanent residents or temporary visitors. Each user carries at least one mobile-connected device (mobile device for short hereafter) , which has a WiFi /Bluetooth interface and is able to acquire his position via on-device positioning software. Such assumptions on device capabilities are fairly justifiable on most current and future mobile devices. In addition, time is divided into equal-length *epoch*s, each represented by a globally unique epoch index of l_{epoch} bits. We also postulate that each mobile device, which may traverse different time zones, can always convert its local time into the corresponding epoch index.

Each user u's *spatiotemporal profile* is defined as a set of 2-tuples $(i, loc_{u,i})$, where i and $loc_{u,i}$ denote the epoch index and the corresponding location index, respectively. In our protocol, $loc_{u,i}$ comprises some physical locations closely approximating the user's whereabouts in epoch i. The detailed construction of spatiotemporal profiles is postponed to Section III.

We use Alice and Bob as two exemplary mobile users throughout the paper. Let $\mathcal{P}_A = \{(i, loc_{A,i})\}_{i>0}^{\infty}$ and $\mathcal{P}_B = \{(i, loc_{B,i})\}_{i>0}^{\infty}$ denote the spatiotemporal profiles of Alice and Bob, respectively. We also let $\mathcal{P}_{A,\alpha\to\beta}$ and $\mathcal{P}_{B,\alpha\to\beta}$ denote their respective spatiotemporal profiles from epochs α to β. Assume that they want to compare their spatiotemporal profiles, which can be done in different ways according to concrete application scenarios. For example, in applications like Participatory Sensing [2], Alice and Bob would execute our protocol directly on their mobile devices when in each other's vicinity; in applications like Missed Connection, Alice and Bob could both synchronize their profiles to computers and conduct matching via free dedicated missed-connection sites; in applications like Ad Targeting, one of them serving as an advertiser uses a wired computer, and the other could use either a mobile device or wired computer, in which case spatiotemporal matching can be conducted via some third-party Internet ad broker. For simplicity, we ignore such underlying application details and simply assume the availability of a suitable communication channel between Alice and Bob.

Assume that Alice and Bob want to compare their spatiotemporal profiles from epochs α to β. A complete matching process involves each of them initiating an independent protocol instance. The number of encounters with Bob in Alice's eye in any epoch $i \in [\alpha, \beta]$ equals the number of common locations in their location indexes in epoch i, and the number of encounters with Bob from epochs α to β in her eye equals the sum of total encounters in every epoch from α to β. In the similar fashion, we can define the total number of encounters with Alice from Bob's viewpoint from epochs α to β. We proceed to introduce the following definition.

Definition 1. (**Spatiotemporal Match**) *After protocol execution, a* spatiotemporal match *between Alice and Bob from epochs α to β is said to occur if the total number of encounters with Bob exceeds τ_A from Alice's viewpoint, and the total number of encounters with Alice exceeds τ_B from Bob's viewpoint, where τ_A and τ_B are personal thresholds independently chosen by Alice and Bob, respectively.*

We assume that Alice and Bob both desire strong spatiotemporal privacy. Our focus is to devise an efficient protocol ensuring that as little information as possible about the spatiotemporal profile of either Alice or Bob is disclosed beyond the matching result. One may think about letting them directly exchange and compare their spatiotemporal profiles under pseudonyms instead of real names so that a known spatiotemporal profile cannot be directly linked to a real identity. Unfortunately, the knowledge of a pseudo-identity's spatiotemporal profile may be disastrous enough, e.g., leading to physical chasing to unveil the corresponding real identity. We thus need a sound solution regardless of the use of pseudonyms.

B. Adversary Model

We assume a honest-but-curious adversary model commonly adopted to study privacy-preserving profile matching [3]–[5]

or proximity test [6], [7]. With Alice and Bob as an example, they both honestly follow the spatiotemporal matching protocol while having great curiosity about the other's spatiotemporal profile. Although either or both of them may use honest or fake spatiotemporal profiles in matching, we will show that a randomly forged spatiotemporal profile will very unlikely result in a spatiotemporal match with the other party.

We do not consider *continuous fake-profile* attacks and *denial-of-service* (DoS) attacks due to space limitations. In the former, either matching participant keeps using fake spatiotemporal profiles possibly under different pseudonyms in order to accumulate more information about the other party's spatiotemporal profile as time goes by, while in the latter, an attacker just aims at depleting the resources of the other party in the same way. The only feasible countermeasure against both attacks in our opinion is for every party to rate-limit the total number of matching requests he/she will accept. Further investigations on these attacks is beyond the scope of this paper.

There might also be external eavesdroppers or physical chasers. The former overhear the messages incurred by a spatiotemporal matching instance and can be easily thwarted by letting the matching participants encrypt the protocol messages. The latter tail a victim user and thus can always have a spatiotemporal profile resembling that of the victim user. There is no sound technical solution to such chasing attacks.

III. Privacy-preserving Spatiotemporal Matching

In this section, we first introduce a basic protocol for privacy-preserving spatiotemporal matching based on Private Set Intersection Cardinality (PSI-CA) [8] and then present a much more efficient one based on estimation.

A. Basic Protocol based on PSI-CA

Our protocol explores the prevalent capability of mobile devices obtaining their physical locations based on hybrid GPS, WiFi, and cellular positioning techniques. Assume that each epoch is evenly divided into λ intervals, where $\lambda \geq 1$ is a global parameter. In general, each user passively records his location in the middle of each interval to tolerate synchronization errors among mobile devices. Recall that any user u's spatiotemporal profile is defined in Section II-A as a set of 2-tuples like $(i, loc_{u,i})$. We have $loc_{u,i} = \{p_{u,i}[j]\}_{j=1}^{\lambda}$, where $p_{u,i}[j]$ denotes user u's jth location in epoch i. Consider the exemplary users Alice and Bob with profiles $\mathcal{P}_A = \{i, \{p_{A,i}[j]\}_{j=1}^{\lambda}\}_{i=1}^{\infty}$ and $\mathcal{P}_B = \{i, \{p_{B,i}[j]\}_{j=1}^{\lambda}\}_{i=1}^{\infty}$, respectively. Now they attempt to compare their profiles from epochs α to β, i.e., $\{i, \{p_{A,i}[j]\}_{j=1}^{\lambda}\}_{i=\alpha}^{\beta}$ and $\{i, \{p_{B,i}[j]\}_{j=1}^{\lambda}\}_{i=\alpha}^{\beta}$, equivalent to the comparison of $\lambda(\beta - \alpha + 1)$ location pairs.

We further assume that each physical region of interest (like a metropolitan area) can be approximated by a square called a *level-1* cell. Then we divide the level-1 cell into four equally-sized squares called *level-2* cells, each of which is further divided into four equally-sized squares named as level-3 cells. This process continues until reaching level-θ cells, each having a side length no larger than a desired threshold, and how to determine the cell-division threshold will be discussed later. Note that there are totally 4^{j-1} level-j cells for $\forall j \in [1, \theta]$. Then we assign a unique cell index to the cell(s) on every level. In particular, the index of the level-1 cell is 0, and the indexes of the upper-left, lower-left, upper-right, and lower-right level-2 cells are 00, 01, 02, and 03, respectively. The same indexing rule can be applied to the cells on all levels. The region-division rules are public information and can be downloaded as needed. In practice, each user just needs to have the rules related to the regions he commonly stay in or travel to, so the related storage overhead is negligible.

To facilitate the comparison of location pairs, we propose an *adaptive quantization* technique which works by letting each user convert his locations into cell indexes. In particular, assume that Alice and Bob negotiate a common region of interest on which to conduct spatiotemporal matching. Since each region corresponds to a large geographic area, disclosing the regions of interest to each other may not be a serious concern in practice; otherwise, Alice and Bob can apply Private Set Intersection (PSI) [10] to negotiate the common region, which will be very efficient given the limited possible regions. In addition, they agree on a cell level $\xi \in [1, \theta]$ on which the quantization takes place, and the impact of ξ will be discussed shortly. Then Alice converts $\{i, \{p_{A,i}[j]\}_{j=1}^{\lambda}\}_{i=\alpha}^{\beta}$ into $\overline{\mathcal{P}}_{A,\alpha\to\beta} = \{\{i, j, \bar{p}_{A,i}[j]\}_{j=1}^{\lambda}\}_{i=\alpha}^{\beta}$, where $\bar{p}_{A,i}$ denotes the index of the level-ξ cell that contains $p_{A,i}$. If a certain location is not in the negotiated region, the corresponding cell index is set to some randomly chosen unlikely cell index indicating this abnormality. Similarly, Bob can convert his profile $\{i, \{p_{B,i}[j]\}_{j=1}^{\lambda}\}_{i=\alpha}^{\beta}$ into $\overline{\mathcal{P}}_{B,\alpha\to\beta} = \{\{i, j, \bar{p}_{B,i}[j]\}_{j=1}^{\lambda}\}_{i=\alpha}^{\beta}$. With adaptive quantization in place, the number of encounters between Alice and Bob equals the number of level-ξ cells they both came across in the same epoch interval, or equivalently the intersection cardinality $|\overline{\mathcal{P}}_{A,\alpha\to\beta} \bigcap \overline{\mathcal{P}}_{B,\alpha\to\beta}|$.

How could Alice and Bob discover $|\overline{\mathcal{P}}_{A,\alpha\to\beta} \bigcap \overline{\mathcal{P}}_{B,\alpha\to\beta}|$ in a privacy-preserving fashion? An intuitive idea is to apply a PSI-CA protocol such as [8]. If Alice initiates the protocol, the PSI-CA protocol allows her to know nothing about $\overline{\mathcal{P}}_{B,\alpha\to\beta}$ other than $m_A = |\overline{\mathcal{P}}_{A,\alpha\to\beta} \bigcap \overline{\mathcal{P}}_{B,\alpha\to\beta}|$. Likewise, if Bob initiates the protocol, he can learn nothing about $\overline{\mathcal{P}}_{A,\alpha\to\beta}$ beyond $m_B = |\overline{\mathcal{P}}_{A,\alpha\to\beta} \bigcap \overline{\mathcal{P}}_{B,\alpha\to\beta}|$. Finally, Alice checks if m_A is greater than her personal threshold τ_A, and Bob checks if m_B is greater than his personal threshold τ_B. If so, a successful spatiotemporal matching occurs.

Although this basic method allows Alice and Bob to determine the actual number of encounters and offers very high-level privacy protection for them, it has large communication and computation overhead when $\lambda(\beta - \alpha + 1)$ is large and is thus not suitable for resource-constrained mobile devices, as we will analyze in Section IV-B and evaluate in Section V. It is thus necessary to explore a more efficient alternative.

B. Advanced Protocol based on Bloom Filter

Our second protocol involves a novel use of the Bloom filter [9] and is motivated by the observation that an accurate estimation of the number of encounters may suffice in practice.

A Bloom filter [9] is a space-efficient probabilistic data structure for set-membership testing and many other applications [11], [12]. Assume that we want to use a w-bit Bloom

filter for a data set $\{s_i\}_{i=1}^{d}$, which has every bit initialized to bit-0. Let $\{h_a(\cdot)\}_{a=1}^{k}$ denote k different hash functions, each with output in $[1, w]$. Every element s_i is added into the Bloom filter by setting all bits at positions $\{h_a(s_i)\}_{a=1}^{k}$ to bit-1. To check the membership of an arbitrary element e in the given data set, we can simply verify whether all the bits at positions $\{h_a(e)\}_{a=1}^{k}$ have been set. If not, e is certainly not in the data set; otherwise, it is in the data set with some probability jointly determined by d, w, and k.

Our protocol involves each of Alice and Bob using a different set of hash functions to construct a Bloom filter based on her/his spatiotemporal profile. In particular, let $\mathcal{H}$ denote a large and public pool of hash functions with each indexed by a unique identifier. Assume that Alice and Bob are to find out $|\overline{\mathcal{P}}_{A,\alpha\to\beta} \bigcap \overline{\mathcal{P}}_{B,\alpha\to\beta}|$ as in the basic protocol. The following operations are done in sequence for Alice to obtain an estimated $\hat{m}_A$ about $|\overline{\mathcal{P}}_{A,\alpha\to\beta} \bigcap \overline{\mathcal{P}}_{B,\alpha\to\beta}|$.

1. Alice sends a matching request to Bob.
2. Bob randomly chooses k hash functions from $\mathcal{H}$ with indexes denoted by $\mathcal{H}_B$ and then inserts each element in his quantized profile $\overline{\mathcal{P}}_{B,\alpha\to\beta}$ into a w-bit Bloom filter (denoted by BF_B) with different $l < k$ functions randomly selected from $\mathcal{H}_B$ and $k - l$ random hash functions outside $\mathcal{H}$. Finally, Bob returns $\mathcal{H}_B$ and BF_B to Alice.
3. Alice constructs a w-bit Bloom filter (denoted by BF_A) based on the hash functions specified in $\mathcal{H}_B$ and her quantized profile $\overline{\mathcal{P}}_{A,\alpha\to\beta}$. Then she counts the number of common bit-0 positions in BF_A and BF_B (denoted by n_0) whereby to compute

$$\hat{m}_A = \frac{2k\lambda(\beta - \alpha + 1) - w(\ln w - \ln n_0)}{l}. \quad (1)$$

The correctness and accuracy of this estimation will be thoroughly analyzed in Section IV-C.

Likewise, Bob can estimate the number of encounters with Alice from epochs α to β as $\hat{m}_B$. Finally, they can jointly determine whether there is a successful spatiotemporal matching after independently comparing $\hat{m}_A$ (or $\hat{m}_B$) with the personal threshold τ_A (or τ_B).

We have some important remarks to make. First, since Alice and Bob use some common hash functions in $\mathcal{H}_B$ to construct their respective Bloom filter, the same pairs of epoch and cell indexes in their quantized spatiotemporal profiles (if any) are likely to set the same bit positions. So we can estimate the number of common pairs of epoch and indexes via the number of common bit-0 and/or bit-1 positions. Second, the reason for Bob using $k-l$ random hash functions unknown to Alice for each epoch-cell index pair is to prevent Alice from accurately estimating the cell indexes of Bob by simple Bloom set-membership tests. In particular, if Bob uses the same k hash functions in $\mathcal{H}_B$ to generate BF_B, Alice can easily test whether every possible epoch-cell index is in BF_B, which is equivalent to breaching Bob's spatiotemporal privacy. The choice of k and l will be detailed in Section IV-C. Finally, the construction of many different hash functions for implementing the Bloom filter is also very important. One common method is to seed a cryptographic hash function such as SHA-2 with the indexes of hash functions we want. There are also some more efficient realizations of many hash functions specifically for the Bloom filter [13], [14].

C. Discussion

There are some important design issues to discuss.

Impact of recording frequency: Each user records his location in the middle of each interval in each epoch of fixed length. The fewer intervals in each epoch, the lower the recording frequency, and the more likely for *matching false negatives* to occur, in which case a protocol initiator considers the responder a mismatch who actually encountered him multiple times and just did not record the encounter locations due to the low recording frequency. In contrast, the higher the recording frequency, the less likely for matching false negatives to occur, and the longer every location index in every epoch which will lead to larger computation and communication overhead.

Impact of quantization granularity: The granularity of spatiotemporal matching can be controlled by choosing a proper quantization level $\xi \in [1, \theta]$. A larger ξ can lead to finer-grained matching at the sacrifice of spatiotemporal privacy, while a smaller ξ can lead to better spatiotemporal privacy at the cost of coarser-grained matching.

Impact of imperfect quantization: Our quantization process may cause some ambiguity. For example, if the recorded locations of Alice and Bob in the same interval are near the upper-left and lower-right corners of the same level-ξ cell, they will be quantized to the same level-ξ index and thus translated into one encounter. In contrast, if the two locations are in adjacent level-ξ cells and close to each other along the cell boundary, they, however, will be quantized to different level-ξ indexes and translated into a non-encounter. To resolve such ambiguity, we plan to adopt more advanced and complex quantization technique such as those proposed in [6] in our future work.

Impact of long-time encounters: If two users stay in each other's proximity for a long time with their recorded locations always quantized to the same cell indexes, they will discover multiple encounters via our protocols, though some people may consider that they just have one encounter lasting for a long time. We argue that the matching result from our protocol execution somehow reflects the true spatiotemporal relationship between them and is desired in practice. In particular, given two other users who encountered him both once but for different amount of time, a user may naturally have more trust in the one with longer encounter time.

IV. Performance Analysis

In this section, we conduct theoretical analysis of the basic and advanced protocols.

A. Performance Metrics

We use the following metrics to evaluate our protocols.

Accuracy: The following standard (ϵ, δ) guarantee is used to measure the accuracy of the protocol output,

$$Pr[(1-\epsilon)m \leq \hat{m} \leq (1+\epsilon)m] > 1-\delta, \tag{2}$$

where m is the actual number of common elements (or encounters), and $\hat{m}$ is the estimation of m via our protocol.

Privacy: We quantify spatiotemporal privacy by the Shannon entropy, a commonly used measure of uncertainty. For example, recall that Bob's quantized spatiotemporal profile from epochs α to β is $\overline{\mathcal{P}}_{B,\alpha\to\beta} = \{\{i, j, \bar{p}_{B,i}[j]\}_{j=1}^{\lambda}\}_{i=\alpha}^{\beta}$, where $\bar{p}_{B,i}$ denotes a level-ξ cell index. The only information Alice knows about $\overline{\mathcal{P}}_{B,\alpha\to\beta}$ before protocol execution include the parameters α, β, and λ. Since there are totally $N = 4^{\xi-1}$ level-ξ cell indexes, each of them is equally likely to be $\bar{p}_{B,i}[j]$ from Alice's viewpoint. There are thus totally $N^{\lambda(\beta-\alpha+1)}$ candidate quantized profiles for $\overline{\mathcal{P}}_{B,\alpha\to\beta}$ with equal probability in Alice's eye. So the maximum spatiotemporal privacy of Bob with regard to Alice (i.e., the maximum uncertainty of his spatiotemporal profile to Alice) can be evaluated in bits as

$$\mathbf{E}^* = \log_2 N^{\lambda(\beta-\alpha+1)} = 2\lambda(\beta-\alpha+1)(\xi-1). \tag{3}$$

After the execution of either protocol, Alice can know more information about the probability of each candidate profile being Bob's profile whereby to reduce the entropy or uncertainty, which we will analyze later in this section. The maximum spatiotemporal privacy of Alice with regard to Bob is the same as above.

Overhead: We will analyze the communication and computation overhead of both protocols.

B. Analysis of Basic Protocol based on PSI-CA

We focus on the case that Alice initiates one protocol run with Bob, and the other case can be likewise analyzed.

Accuracy Analysis: Since the underlying PSI-CA protocol such as [8] outputs the exact number of common elements of two quantized spatiotemporal profiles, our basic protocol can achieve 100% accuracy (i.e., $\hat{m} = m$).

Privacy Analysis: After the protocol execution, Alice only knows m, the number of common elements in her profile $\overline{\mathcal{P}}_{A,\alpha\to\beta}$ and Bob's profile $\overline{\mathcal{P}}_{B,\alpha\to\beta}$. To determine Bob's profile, Alice first needs to guess the m intervals in which she and Bob encountered, and there are $\binom{\lambda(\beta-\alpha+1)}{m}$ possibilities with $N = 4^{\xi-1}$ for the level-ξ quantization. In addition, for each of the rest $\lambda(\beta-\alpha+1)-m$ intervals, there are $N-1$ possible cell indexes (excluding Alice's own). Therefore, the entropy of Bob's profile to Alice is given by

$$\mathbf{E}_1 = \log_2 \binom{\lambda(\beta-\alpha+1)}{m} + (\lambda(\beta-\alpha+1)-m)\log_2(N-1). \tag{4}$$

Overhead Analysis: The computation and communication overhead depend on the underlying PSI-CA protocol. Consider the one in [8] as example. If Alice initiates the protocol execution, she needs up to $2(\lambda(\beta-\alpha+1)+1)$ modular exponentiations and $\lambda(\beta-\alpha+1)$ modular multiplications, and Bob needs up to $2\lambda(\beta-\alpha+1)$ modular exponentiations and $\lambda(\beta-\alpha+1)$ modular multiplications. The overall communication overhead is about $(3\lambda(\beta-\alpha+1)+2)1024$ bits to ensure 1024-bit security which is considered necessary and sufficient in practice.

C. Analysis of Advanced Protocol based on Estimation

Accuracy Analysis: The accuracy of the advanced protocol is guaranteed by the following theorem.

Theorem 1. *Given the number of common bit-0 positions n_0 in the w-bit Bloom filters BF_A and BF_B constructed in the advanced protocol, Alice can estimate $|\overline{\mathcal{P}}_{A,\alpha\to\beta} \bigcap \overline{\mathcal{P}}_{B,\alpha\to\beta}|$ as*

$$\hat{m} = \frac{2nk - w(\ln w - \ln n_0)}{l}, \tag{5}$$

where $n = \lambda(\beta-\alpha+1)$ is the number of elements in both $\overline{\mathcal{P}}_{A,\alpha\to\beta}$ and $\overline{\mathcal{P}}_{B,\alpha\to\beta}$. Assuming that $\epsilon m \geq 1$, $\hat{m}$ is an (ϵ, δ) estimation of m if

$$\delta \geq \frac{w(e^{\frac{2nk}{w}} - (1+\frac{2nk}{w}))}{l^2\epsilon^2 m^2}. \tag{6}$$

Proof: For each bit position of either Bloom filter, the probability that it is set to bit-1 by a common element with l common hash functions is given by

$$p = 1 - (1-\frac{1}{w})^{ml} \approx 1 - e^{-\frac{ml}{w}}. \tag{7}$$

The probability that it is set to bit-1 in all the other cases is given by

$$q = 1 - (1-\frac{1}{w})^{nk-ml} \approx 1 - e^{-\frac{nk-ml}{w}}. \tag{8}$$

Therefore, the probability that a position is bit-0 in both BF_A and BF_B (i.e., common bit-0 position) is given by

$$P_0 = (1-p)(1-q)^2 = e^{-\frac{ml}{w}} e^{-\frac{2(nk-ml)}{w}}. \tag{9}$$

Since Alice can count the number of common bit-0 positions n_0 in BF_A and BF_B, the following equation can be established

$$P_0 = e^{-\frac{ml}{w}} e^{-\frac{2(nk-ml)}{w}} = \frac{n_0}{w}. \tag{10}$$

Solving this equation, we have

$$\hat{m} = \frac{2nk - w(\ln w - \ln n_0)}{l}. \tag{11}$$

Next, we derive the variance. We cast the problem into RFID tag estimation and refer to the results in [15]. The RFID system with t tags divides a time period into f slots and let each RFID tag randomly select one of f slots to respond. One slot may be responded by zero, one, or multiple tags. The expected number of zero-response slots is nearly $fe^{-t/f}$. Knowing the number of zero-response slots, the system administrator can estimate the number of present RFID tags. Our estimation method based on the Bloom filter is similar to RFID tag estimation if we consider common bit-1 positions and common bit-0 positions as multiple-response and zero-response slots in the RFID system, respectively. The expected number of common bit-0 positions of BF_A and BF_B is nearly $we^{-(2nk-ml)/w}$. Knowing the number of common bit-0 positions, we can estimate the intersection size m.

Let $\rho = \frac{2nk-ml}{w}$. According to Theorem 1 in [15], we have $n_0 \sim \mathcal{N}(\mu, \sigma^2)$, where

$$\mu = w(1 - \frac{1}{w})^{2nk-ml} = we^{-\rho} , \tag{12}$$

$$\sigma^2 = we^{-\rho}(1 - (1+\rho)e^{-\rho}) . \tag{13}$$

We can view μ as a function of the true number of common elements, denoted by $\mu(m)$. Since $\mu(m)$ is monotonic continuous functions of m, it has a unique inverse, denoted by $g()$, i.e., $g(\mu(m)) = m$. Let $2nk - ml \to \infty$ and $w \to \infty$, while maintaining $\frac{2nk-ml}{w} = \rho$. Since $g(\mu(m)) = m$, differentiating this equation with respect to m, we get $g'(\mu(m))\mu'(m) = 1$. it follows that $g'(\mu(m)) = \frac{1}{\mu'(m)}$. According to Theorem 6 in [15], the variance of common bit-0 estimation of m is given by

$$\delta_0 = \sigma^2(m)[g'(\mu(m))]^2 = \frac{\sigma^2(m)}{[\mu'(m)]^2} . \tag{14}$$

Since $\mu = we^{-\frac{2nk-ml}{w}}$ and $\sigma^2 = we^{-\rho}(1-(1+\rho)e^{-\rho})$. Differentiating $\mu(m)$ with respect to m, we can obtain $\frac{d\mu(m)}{dm} = le^{-\rho}$. Therefore we have

$$\delta_0 = \frac{we^{-\rho}(1-(1+\rho)e^{-\rho})}{l^2e^{-2\rho}} = \frac{w(e^{\rho}-(1+\rho))}{l^2}. \tag{15}$$

In addition, since $\frac{d\delta_0}{d\rho} = \frac{w}{l^2}(e^{\rho}-1) > 0$, we know that δ_0 is monotonic increasing with ρ. Since $0 \le m \le n$, we have $\frac{n(2k-l)}{w} \le \rho \le \frac{2nk}{w}$. Therefore when $\rho = \frac{2nk}{w}$, we have

$$\delta_{0\max} = \frac{w(e^{\frac{2nk}{w}} - (1+\frac{2nk}{w}))}{l^2} . \tag{16}$$

We thus have $\hat{m} \sim \mathcal{N}(m, \delta_0)$. According to the Chebyshev's inequality, we have

$$Pr(|\hat{m} - m| \le \epsilon m) \ge 1 - \frac{\delta_0}{\epsilon^2 m^2} \ge 1 - \delta . \tag{17}$$

Therefore, $\hat{m}$ is an (ϵ, δ) estimation of m if

$$\begin{aligned} \delta &\ge \frac{\delta_{0\max}}{\epsilon^2 m^2} \\ &= \frac{w(e^{\frac{2nk}{w}} - (1+\frac{2nk}{w}))}{l^2\epsilon^2 m^2}. \end{aligned} \tag{18}$$

■

Privacy Analysis: The privacy analysis of advanced protocol is given by the following theorem.

Theorem 2. *Let* BF_B *denote a* w*-bit Bloom filter Bob constructs on his level-*ξ *quantized profile* $\overline{\mathcal{P}}_{B,\alpha\to\beta} = \{\{i, j, \bar{p}_{B,i}[j]\}_{j=1}^{\lambda}\}_{i=\alpha}^{\beta}$ *using* l *functions from* $\mathcal{H}_B$ *and* $k-l$ *functions unknown to Alice. After transmitting* BF_B *and* $\mathcal{H}_B$ *to Alice, his remaining privacy of* $\overline{\mathcal{P}}_{B,\alpha\to\beta}$ *against Alice is*

$$\mathbf{E} = \lambda(\alpha+\beta-1)\mathbf{E}[i,j] , \tag{19}$$

where

$$\begin{aligned} \mathbf{E}[i,j] &= \sum_{x=1}^{N}\binom{N}{x}P^x(1-P)^{N-x}\log_2 x, \\ P &= \sum_{i=l}^{k}\binom{k}{i}p^i(1-p)^{k-i} , \\ p &= 1 - e^{-\frac{\lambda(\alpha+\beta-1)k}{w}} . \end{aligned} \tag{20}$$

Proof: Bob's privacy disclosure is caused by transmitting BF_B and the indexes $\mathcal{H}_B$ of k hash functions to Alice. In particular, Alice can exploit BF_B and the knowledge that Bob inserts every element in $\overline{\mathcal{P}}_{B,\alpha\to\beta}$ using l random hash functions from $\mathcal{H}_B$ and $k-l$ unknown hash functions to deduce some information about $\overline{\mathcal{P}}_{B,\alpha\to\beta}$. Consider an arbitrary element $\langle i, j, \bar{p}_{A,i}[j]\rangle$ as an example. For each of the N possible cell indexes, say cID, Alice can test whether it is a viable candidate for the unknown $\bar{p}_{A,i}[j]$ by using all the k hash functions in $\mathcal{H}_B$ to compute the k corresponding positions for the resulting element $\langle i, j, cID\rangle$. If there are at least l out of k corresponding positions set to bit-1 in BF_B, we have $cID = \bar{p}_{A,i}[j]$ with probability P; otherwise, we must have $cID \ne \bar{p}_{A,i}[j]$.

Now we estimate P. After inserting all the $\lambda(\alpha+\beta-1)$ elements in $\overline{\mathcal{P}}_{B,\alpha\to\beta}$ into BF_B, the expected number of bit-1 positions is $w(1-(1-\frac{1}{w})^{\lambda(\alpha+\beta-1)k})$. For a random hash function applied to cID, the probability of the corresponding bit position having been set to bit-1 is

$$p = 1 - (1-\frac{1}{w})^{\lambda(\alpha+\beta-1)k} \approx 1 - e^{-\frac{\lambda(\alpha+\beta-1)k}{w}} . \tag{21}$$

The probability that at least l corresponding bit positions corresponding to cID have been set to bit-1 is then given by

$$P = \sum_{i=l}^{k}\binom{k}{i}p^i(1-p)^{k-i} . \tag{22}$$

Let $X_{i,j}$ denote the number of valid candidate cell indexes for $\bar{p}_{A,i}[j]$. The remaining entropy for interval i in epoch j is then $\log_2 X_{i,j}$. Since $X_{i,j}$ is randomly distributed in $[1, N]$ ($N = 4^{\xi-1}$), we have the mean remaining entropy for interval i in epoch j as

$$\begin{aligned} \mathbf{E}[i,j] &= \sum_{x=1}^{N} Pr(X_{i,j} = x)\log_2 x \\ &= \sum_{x=1}^{N}\binom{N}{x}P^x(1-P)^{N-x}\log_2 x . \end{aligned} \tag{23}$$

Assuming that the $\lambda(\beta-\alpha+1)$ intervals are independent from each other, the total remaining entropy is given by

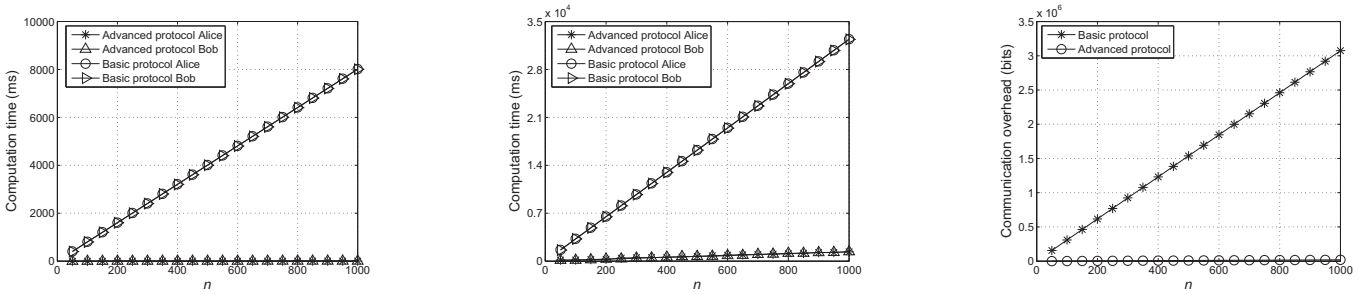
$$\mathbf{E} = \sum_{i=\alpha}^{\beta}\sum_{j=1}^{\lambda}\mathbf{E}[i,j] = \lambda(\alpha+\beta-1)\mathbf{E}[i,j] . \tag{24}$$

■

Overhead Analysis: In contrast to the basic protocol, the advanced protocol does not depend on expensive cryptographic operations and involves Alice and Bob each performing

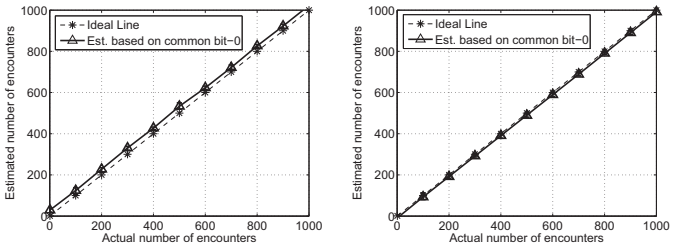
TABLE I: Comparison of Privacy-preserving Spatiotemporal Matching Protocols

Protocol	**Result Type**	**Alice's Comp.**	**Bob's Comp.**	**Comm.**
Basic Protocol	Accurate	$2(n+1)$ exp, n mul	$2n$ exp, n mul	$(3n+2)1024$ bits
Advanced Protocol	Estimated	n hash	n hash	w bits



(a) Computation overhead on Dell PC (b) Computation overhead on LG P-970 (c) Communication overhead

Fig. 1: Computation and communication overhead.



(a) Effect of n (b) Effect of $\hat{n}$

Fig. 2: The estimation accuracy of the advanced protocol.

$k\lambda(\beta-\alpha+1)$ hash operations which are very efficient. The communication overhead comes from the transmission of one Bloom filter and is of w bits.

V. Performance Evaluation

In this section, we evaluate our protocols using simulations. Table I summarizes the theoretical performance of the basic protocol based on PSI-CA [8] and the advanced protocol based on the Bloom filter, where exp and mul denote 160-bit exponentiation operation and 1024-bit multiplication operation, respectively, hash denotes one hash operation, $n = \lambda(\beta-\alpha+1)$, and w is the Bloom-filter length. It is clear that the non-cryptographic advanced protocol theoretically incurs significantly lower computation and communication overhead than the cryptographic basic protocol, which will be backed up by our simulation results.

A. Simulation Settings

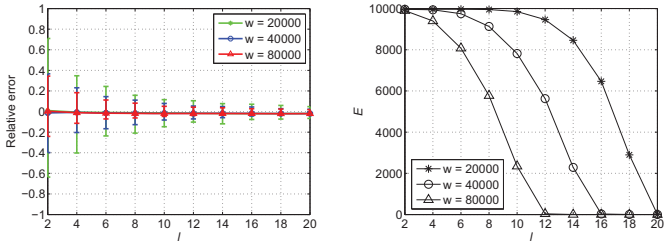
We first evaluate the time taken for exp and mul on a Dell PC with 2.67 GHz CPU, 9 GB RAM, and Windows 7 64-bit Professional, and also on a LG P-970 smartphone with 1 GHz Cortex-A8 processor, 512 MB RAM, and Android v2.2. It takes 4 ms for exp and 0.0076 ms for mul on the PC on average, as well as 15.83 ms for exp and 0.73 ms for mul on LG P-970 on average. These timing data are the basis of our simulations.

Other simulation settings are as follows. We assume that the quantization is done on the level $\xi = 6$, i.e., $N = 4^{\xi-1} = 1024$. In addition, the evaluation program is written in Java, and every data point represents the average of 1000 runs. As discussed, a complete spatiotemporal matching involves Alice and Bob each initiating one protocol execution, but we only show the results for one protocol execution for simplicity. In addition, we set δ to 0.02, and ϵ is the relative error shown in each figure.

B. Simulation Results

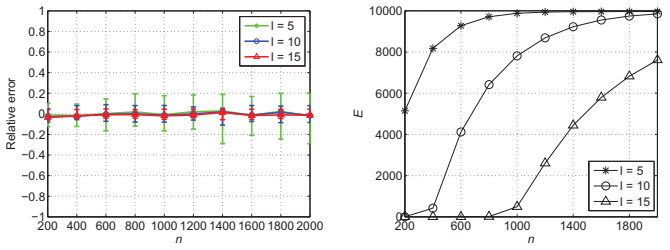
We first compare the computation and communication overhead of our protocols. For the advanced protocol, we set $k = 10$, $l = 10$, and $w = 20n$, and vary n from 50 to 1000. Fig. 1(a) and Fig. 1(b) compare the computation overhead of the two protocols on Dell PC and LG P-970, respectively. It is obvious that the advanced protocol incurs much lower computation overhead than that of the basic protocol due to its reliance on efficient hash functions instead of expensive PSI-CA operations. Fig. 1(c) shows the communication overhead of the two protocols. As expected, the communication overhead of the basic protocol is proportional to n, while the advanced protocol incurs very low overhead which equals to the Bloom-filter length that can be well controlled according to different accuracy and privacy requirements.

Fig. 2(a) compares the estimated number of encounters $\hat{m}$ with the actual number of encounters m, when $k = 20$, $l = 16$, $n = 1000$, and $w = 40000$. We can see that the estimator in Eq. (1) is always biased. The reason is that traditional analysis about the w-bit Bloom filter assumes that every bit position is set to bit-1 for any of n elements with equal probability $1/w$. In practice, however, the probability that one position is set to bit-1 is not independent of other positions: when one position is set to bit-1, it slightly reduces the probability that other positions are set to bit-1 [11], [16], [17]. Therefore, the actual number of bit-1 positions n_1 in the Bloom filter is a little smaller than that obtained via theoretical analysis, and the actual number of bit-0 positions n_0 in the Bloom filter is a little larger than that obtained via theoretical analysis. Since



(a) Relative error (b) Privacy

Fig. 3: The impact of l, the number of common hash functions.



(a) Relative error (b) Privacy

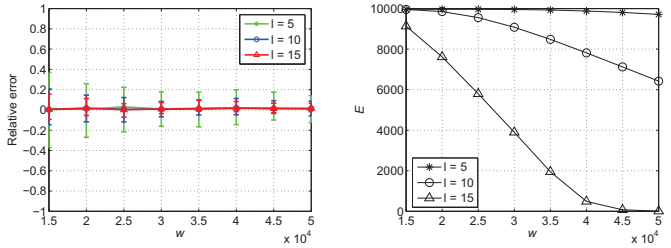
Fig. 4: The impact of n, the cardinality of spatiotemporal profiles.

$\hat{m} = \frac{2kn - w(\ln w - \ln n_0)}{l}$, we can expect $\hat{m}$ to be larger than the true value m.

We resolve the biased estimation by letting $\hat{m} = \frac{2k\hat{n} - w(\ln w - \ln n_0)}{l}$, where $\hat{n} = \frac{\ln(n_{A0}/w)}{k\ln(1-1/w)}$, n_{A0} is the number of bit-0 positions in BF_A. Fig. 2(b) shows that this new estimator is almost unbiased and matches well with m. The reason is that using estimated number of elements $\hat{n}$ instead of the real number of elements $n = \lambda(\beta - \alpha + 1)$ takes into account the above difference between observed and theoretical numbers of bit-0 and bit-1 positions. So we will use this modified estimator hereafter whose effectiveness will be further evidenced.

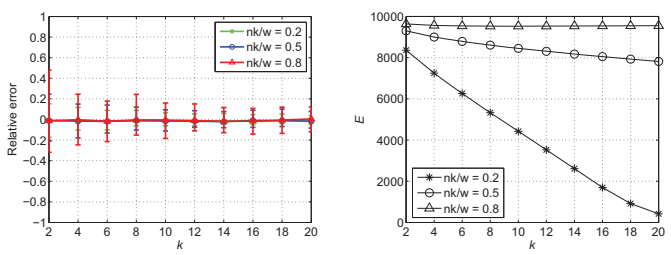
Fig. 3 shows the impact of l (the number of common hash functions Bob chooses to insert each of his elements) on the performance of advanced protocol, when $n = 1000$, $m = 500$, and $k = 20$. We can see from Fig. 3(a) that the more common hash functions (i.e., larger l), the smaller the variance of the relative error $|\hat{m}_A - m|/m$ (i.e., the more accurate the estimation). The reason is that the more common hash functions, the more common bit-0 positions in BF_A and BF_B, leading to fewer possible Bloom filters for Alice and Bob, and the smaller estimation error variance, because the estimation error mainly comes from the uncertainty of BF_A and BF_B. In addition, the more common hash functions Alice and Bob share, the lower the probability that a random location index having corresponding bits set to bit-1 by at least l out of k hash functions, and thus the lower remaining entropy left for Bob's location profile after Alice testing all possible location indexes. It is thus of no surprise to see that Bob's remaining privacy of $\overline{\mathcal{P}}_{B,\alpha\to\beta}$ against Alice decreases with both l and w.

Fig. 4 shows the impact of n (the number of location indexes of each user) on the performance of advanced protocol, when $k = 20$, $w = 40000$, and $m = n/2$. We can see that as n increases, the relative error becomes larger. The reason is that when the Bloom-filter length w is fixed, the more



(a) Relative error (b) Privacy

Fig. 5: The impact of w, the length of the Bloom filter.



(a) Relative error (b) Privacy

Fig. 6: The impact of k, the total number of hash functions.

elements inserted, the fewer common bit-0 positions in BF_A and BF_B, the more possible Bloom filters for Alice and Bob, which leads to higher estimation variance. In contrast, Bob's remaining privacy increases as n increases because the fewer bits-0 positions in BF_A, the higher the probability of a random location index having corresponding bits set to bit-1 by at least l out of k known hash functions, and the higher remaining entropy for Bob's location profile from Alice's point of view after testing all possible location indexes.

Fig. 5 shows the impact of w (the Bloom-filter length) on the performance of advanced protocol, when $k = 20$, $n = 1000$, and $m = 500$. We can see that the relative error decreases as w increases. This is because when the number of elements n is fixed, increase in w leads to more common bit-0 positions. The more common bit-0 positions, the fewer possible Bloom filters for Alice and Bob, and thus the smaller estimation error variance. In addition, Bob's remaining privacy against Alice decreases as w increases. The reason is that the longer the Bloom filter, the lower the probability that a random location index having corresponding bits set to bit-1 by at least l out of k known hash functions, and thus the lower remaining entropy left for Bob's location profile after Alice testing all possible location indexes.

Fig. 6 shows the impact of k (the total number of hash functions for Bloom filter construction) on the performance of advanced protocol, when $n = 1000$, $m = 500$, and the ratios l/k and nk/w are both fixed. It is obvious that the relative error decreases as k increases. The reason is that when k increases, l and w also increase proportionally with fixed l/k and nk/w. Recall that the variance of the $\hat{m}$ is inversely proportional to w/l^2 for fixed ρ (cf. Eq. (15)). As l increases, the variance of estimation error decreases. In addition, Bob's remaining privacy against Alice decreases as k increases. The reason is that the probability that at least l bit positions have been set decreases as k increases, which leads to lower remaining

entropy.

From the above figures, a general conclusion we can draw is that there is an inherent tradeoff between matching accuracy and spatiotemporal privacy: the more accuracy Alice wants, the lower spatiotemporal privacy Bob can enjoy, and vice versa.

VI. Related Work

In this section, we briefly discuss some work in several areas which is most germane to our work in this paper.

There is some work on encounter-based matching [18], [19]. Manweiler *et al.* [18] discussed the privacy concerns for some missed-connection sites, which allows anonymous users to rediscover strangers that they ever encountered. In their follow-on work [19], they proposed to let mobile users exchange spatiotemporal credentials when encountering each other and later attempt to discover each other via a third-party server which acts as a rendezvous point for users. In contrast, our protocols focus on a more general problem and are completely distributed without requiring mobile users to interact with each other or a third-party server.

As discussed, the protocols proposed in [3], [4], [20] aim at coarse-grained matching of user's non-spatiotemporal personal profiles such as hobbies or interests. It is unclear how to extend these schemes for privacy-preserving spatiotemporal matching.

Private proximity testing aims at testing the physical proximity of two users at some discrete time points in a privacy-preserving fashion. In [6], private proximity test is reduced to private equality test based on some location tags often sent by third parties, and the sketches of GSM location tags [7] are for efficient private proximity test. In contrast, our protocols evaluate the proximity of two users for any desired continuous time period. Moreover, our most efficient protocol does not involve expensive cryptographic operations unlike [6], [7].

There is a series of work on Private Set Intersection (PSI) [21], [22] or Private Set Intersection Cardinality (PSI-CA) [10], [23], whereby two mutually mistrusting parties, each holding a private data set, jointly compute the intersection [21], [22] or the intersection cardinality [10], [23] of the two sets without leaking any additional information to either party. PSI, PSI-CA, or their variants have been the cryptographic foundation of many private matching schemes such as [3], [4], [6], [7], [20]. Our first protocol is also based on PSI-CA, but our second protocol is non-cryptographic.

VII. Conclusion

In this paper, we motivated and formulated privacy-preserving spatiotemporal matching problem as a new way of establishing trust relationships. We also presented a novel cryptographic solution based on PSI-CA and a novel non-cryptographic solution based on a novel use of the Bloom filter. Detailed performance analysis and evaluation confirmed the high efficacy and efficiency of our solutions.

Acknowledgement

This work was supported in part by the US National Science Foundation under grants CNS-1117462 and CNS-0844972 (CAREER). We would also like to thank anonymous reviewers for their constructive comments and helpful advice.

References

[1] "Cisco visual networking index global mobile data traffic forecast update 2011-2016."

[2] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "PriSense: privacy-preserving data aggregation in people-centric urban sensing systems," in *INFOCOM'10*, San Diego, CA, Mar. 2010.

[3] M. Arb, M. Bader, M. Kuhn, and R. Wattenhofer, "VENETA: Serverless friend-of-friend detection in mobile social networking," in *WIMOB'08*, Avignon, France, Oct. 2008, pp. 184–189.

[4] M. Li, N. Cao, S. Yu, and W. Lou, "FindU: Privacy-preserving personal profile matching in mobile social networks," in *INFOCOM'11*, Shanghai, China, Apr. 2011.

[5] R. Zhang, Y. Zhang, J. Sun, and G. Yan, "Fine-grained private matching for proximity-based mobile social networking," in *IEEE INFOCOM'12*, Orlando, FL, Mar. 2012.

[6] A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, and D. Boneh, "Location privacy via private proximity testing," in *NDSS'11*, San Diego, CA, Feb. 2011.

[7] Z. Lin, D. Kune, and N. Hopper, "Efficient private proximity testing with GSM location sketches," in *FC'12*, Bonaire, Feb. 2012.

[8] E. Cristofaro, P. Gasti, and G. Tsudik, "Fast and private computation of cardinality of set intersection and union," Tech. Rep., 2011, http://eprint.iacr.org/.

[9] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Comm. ACM*, vol. 13, no. 7, pp. 422–426, July 1970.

[10] E. Cristofaro and G. Tsudik, "Practical private set intersection protocols with linear complexity," in *FC'10*, vol. 6052, Tenerife, Canary Islands, Spain, Jan. 2010, pp. 143–159.

[11] A. Broder and M. Mitzenmacher, "Network applications of bloom filters: A survey," pp. 636–646, 2002.

[12] Y. Zhao and J. Wu, "B-SUB: A practical bloom-filter-based publish-subscribe system for human networks," in *ICDCS '10*, 2010.

[13] A. Kirsch and M. Mitzenmacher, "Less hashing, same performance: Building a better Bloom filter," in *ESA'06*, Zurich, Switzerland, Sept. 2006.

[14] P. Dillinger and P. Manolios, "Bloom filters in probabilistic verification," in *FMCAD'04*, Austin, TX, USA, Nov. 2004.

[15] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *ACM MOBICOM'06*, Los Angeles, CA, Sep. 2006, pp. 322–333.

[16] K. Christensen, A. Roginsky, and M. Jimeno, "A new analysis of the false positive rate of a bloom filter," *Inf. Process. Lett.*, 2010.

[17] P. Bose, H. Guo, E. Kranakis, A. Maheshwari, P. Morin, J. Morrison, M. Smid, and Y. Tang, "On the false-positive rate of bloom filters," *Inf. Process. Lett.*, 2008.

[18] J. Manweiler, R. Scudellari, Z. Cancio, and L. Cox, "We saw each other on the subway: secure, anonymous proximity-based missed connections," in *HotMobile'09*, Santa Cruz, California, Feb. 2009.

[19] J. Manweiler, R. Scudellari, and L. Cox, "SMILE: encounter-based trust for mobile social services," in *CCS'09*, Chicago, Illinois, USA, Nov. 2009, pp. 246–255.

[20] R. Lu, X. Lin, X. Liang, and X. Shen, "A secure handshake scheme with symptoms-matching for mhealthcare social network," *Mobile Networks and Applications*, pp. 1–12, 2010.

[21] L. Kissner and D. Song, "Privacy-preserving set operations," in *CRYPTO'05*, Santa Barbara, CA, Aug. 2005, pp. 241–257.

[22] Q. Ye, H. Wang, and J. Pieprzyk, "Distributed private matching and set operations," in *ISPEC'08*, vol. 4991, Sydney, Australia, Apr. 2008, pp. 347–360.

[23] M. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *EUROCRYPT'04*, Interlaken, Switzerland, May 2004, pp. 1–19.