Efficient Private Proximity Testing with GSM Location Sketches

Zi Lin, Denis Foo Kune, and Nicholas Hopper
Computer Science & Engineering, University of Minnesota,
Minneapolis, MN 55455
lin, foo, hopper@cs.umn.edu

Abstract. A protocol for private proximity testing allows two mobile users communicating through an untrusted third party to test whether they are in close physical proximity without revealing any additional information about their locations. At NDSS 2011, Narayanan and others introduced the use of unpredictable sets of "location tags" to secure these schemes against attacks based on guessing another user's location. Due to the need to perform privacy-preserving threshold set intersection, their scheme was not very efficient. We provably reduce threshold set intersection on location tags to equality testing using a de-duplication technique known as shingling. Due to the simplicity of private equality testing, our resulting scheme for location tag-based private proximity testing is several orders of magnitude more efficient than previous solutions. We also explore GSM cellular networks as a new source of location tags, and demonstrate empirically that our proposed location tag scheme has strong unpredictability and reproducibility.

1 Introduction

The ability to test for physical proximity to one's friends, co-workers, family, or acquaintances can be useful in a variety of settings. For example, proximity testing has been found to facilitate in-person collaboration and thus increase work productivity [17]. It also has potential for building social networks, since sharing proximity frequently over time indicates common activities and interests, an important factor in friendship [25]. Narayanan et al. [22] list a variety of further scenarios in which it might be useful.

Although RF-based Inter-personal awareness devices (IPAD) were developed in 1991 [17], proximity awareness did not gain much attention until the proliferation of smartphones and online social networking sites. Equipped with GPS receivers and/or base station triangulation, most smartphones are able to pinpoint their geographic coordinates. As a result, several social networking services have been built to use these features. These location-based services ask phone users to submit their presence in a given venue ("check-in") so that friends can interact based on the location proximity. While these services offer many benefits, they also carry significant risks: users must trust the service providers and their friends to handle this location data properly. Unfortunately, it is well-established that indiscriminate handling of location information can lead to a variety of undesirable outcomes, including threats to the physical safety and well-being of users. As a result, a variety of "privacy-preserving" proximity tests have been proposed, allowing users to compare their locations so that nothing about their locations is revealed if they are not nearby.

Perhaps the most efficient such construction is due to Narayanan et al. [22] who also point out that most previous approaches suffer a common vulnerability; a malicious user can substitute a different location from her own in an attempt to learn or confirm the location of other users. To cope with this problem, Narayanan et al. introduce the concept

of location tags. A location tag is a collection of characteristic features derived from the unique combination of time and location. In other words, an ephemeral key that can only be obtained at a given time and a given location. Proximity testing through location tags eliminates the threat of an online attacker who wants to learn the location of the other remote party by actively lying about her own location. Friends recording location tags will be able to test whether they are proximate by measuring the similarity of their tags privately. If they are close enough, i.e. the simiarlity is above a preset threshold, they will be notified, otherwise they learn nothing. However, private threshold set intersection is an expensive primitive that mobile phones may not be capable of executing in a timely fashion.

We address the similarity test in a novel way. We observe that since location tags are sets of high-entropy elements, they are either essentially disjoint or essentially identical. Thus an efficient test that has high probability of accepting near-identical sets and high probability of rejecting near-disjoint sets is sufficient. We adopt de-duplication techniques to reduce "nearly identical" testing to simple equality testing. We also seek to compute location tags from sources other than WiFi, which has a limited coverage and leaves blind voids between different access points.

The main contributions of this work are the following:

1. We reduce location tag based proximity testing to efficient private equality testing, using the shingling de-duplication technique. Via shingling, we generate a concise sketch for each set of location tags. Location proximity should lead to two nearly-identical sets of location tags. Nearly identical sets should yield the same sketch with high probability. Therefore, we are able to test proximity through equality. Private equality testing is more efficient than private threshold set intersection, and requires less tuning.
2. We explore the cellular network as a source of location tags. Compared to WiFi, cellular networks have much better coverage and are much more reliable. In particular, we are able to take the content from the broadcast "paging" channel of GSM cellular networks. Two phones listening to the same channel at the same time period should hear almost identical content. This source of location tags has not been proposed or investigated in the literature previously. We evaluate these location tags by building a prototype that records actual readings from cellular networks.

We organize the paper as follows: We briefly reviewed related literature in section 2. And A high-level description of our system is given in section 3. We elaborate how we capture location tags in the cellular infrastructure and explain in detail how we integrate shingling with location tag requirements in sections 4 and 5. Experiments and results are reported in section 6, followed by discussion in section 7. Our conclusions appear in section 8.

2 Related Work

Proximity Awareness Devices Wireless RF-based devices that detect physical proximity, called Inter-Personal Awareness Devices (IPAD), were introduced by Holmquist et al. in 1991 [17]. A prototype, Hummingbird, [17] was developed: wearable devices that

hummed when two of them were close enough, e.g. within 100 meters. The hummingbird provided continuous updates while complementing the usage of phones, pagers and computers since it did not require infrastructure support.

Location Privacy Atallah and Du studied secure multi-party geometry computations [6]. Although computationally expensive, these protocols allowed honest users to learn whether they are closer than a mutually agreed threshold. But a malicious user could have lied about his/her location in order to learn the other party's rough location. Based on their work, Zhong, Goldberg and Hengartner introduced three protocols for private proximity detection [27] that can either reveal the liar or cost him an exceptional amount of work.

Location privacy by anonymization has been studied extensively. Beresford and Stajano introduced the concept of "mix zones" [7] and Gruteser and Grunwald [14] introduced "cloaking" for k-anonymization. Hundreds of papers have since been published for location anonymization. However, anonymization by quantization or mixing may not provide the desired privacy for a variety of reasons [24]; in one example, the obfuscated location becomes more accurate when well populated.

Location tags, initially studied by Qiu et al. in [11, 23] as time-invariant location characteristics, were extended by Narayanan et al. to be a nonce associated with a unique location and time combination [22]. With proper location tags, location proximity can be reduced to measuring similarity between two sets of tags. Narayanan et al. suggested deriving tags from surrounding environment including WiFi traffic and Access Point identifiers, GSM signals, audio and even atmospheric gas composition.

Private Set Operations and Private Equality Testing (PET). In multi-party protocols for Private set operations, each participant has a set of elements as input and the parties wish to compute some operation on the sets while revealing nothing about the inputs beyond the result. Freedman et al. studied a private set intersection (PSI) protocol based on homomorphic encryption [13]. Kissner and Song presented a general protocol for multi-party set operations [21]. Protocols focusing on different aspects of private set intersection have been abundant [10, 15, 16, 19, 20]. When both participants have a singleton set, PSI reduces to private equality testing (PET).

A seminal work by Fagin, Naor and Winkler presented multiple PET protocols [12]. Inspired by PSI protocols, Narayanan et al. [22] described two efficient PET protocols, which we describe in the next section. In contrast to these works, we study how to reduce set *similarity* measurements – operations on non-singleton sets – to PET.

3 System Overview

In this section we first give a brief high-level overview of our approach. Details of the new building blocks are discussed in sections 4 and 5. The overall architecture is shown in Figure 1. It has three major ingredients, the first two of which are new in this work:

1. GSM paging channel dumping: mobile phones record all messages received on a special broadcast channel used by all phones in range of the same cellular tower, or in the same location area.

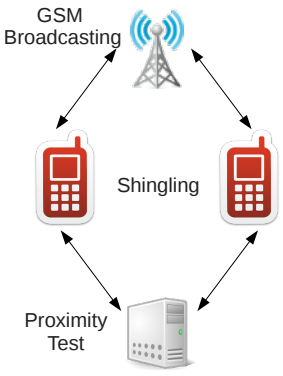

Fig. 1: The overall architecture of location tag system

2. Location sketch generation by shingling: phones use the shingling document de-duplication technique to produce a short string (a *sketch*) that represents the set of broadcast messages received, such that if two sets are similar, then they will have the sam ity; if they are not similar, then with high probability t
3. Private equality testing: Given a location sketch, phone can test for proximity using a private equality test on their sketches.

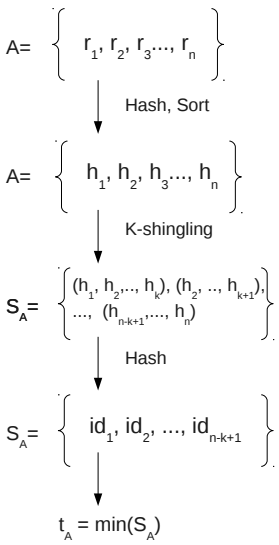

Fig. 2: The shingling process

3.1 GSM Location Tags

The previous work on location tags used wireless broadcast messages over IEEE 802.11 networks [22], but with a range limited to the access points belonging to a given WLAN. To provide wider coverage, we use the GSM network with base stations broadcasting at much higher power and covering a much wider area. Particularly, we use the set of messages received on the GSM paging channel as elements of a location tag set. When the cellular network initiates contact with a mobile station (phone), it issues a paging request on the broadcast paging channel of all base stations within a specific local calling area (of size at most 100km^2) referred to as a Location Area Code (LAC). The mobile station then answers back and is assigned radio resources specific to a particular base transceiver station (BTS) with an immediate assignment message, with a range of at most 1km^2. Each mobile station is assigned a Temporary Mobile Subscriber Identity (TMSI) or a unique International Mobile Subscriber Identity (IMSI). In our measurements, we observed that paging requests are mostly issued using TMSIs (90% of the total paging requests captured). Since the TMSIs are local to a given LAC and one mobile phone only belongs to one LAC at a time, a mobile phone will get a new random TMSI once it travels from one LAC to another. As a consequence, two phones on the same LAC will likely observe the same set of TMSIs on paging requests. On the contrary, two phones on different LACs will likely see disjoint sets of TMSIs. This makes the set of TMSIs seen on the paging channel a good candidate for location tags. In addition, the Immediate Assignment traffic is specific to each base tower within an LAC and can be combined with the TMSIs of paging requests to produce finer-grained location tags.

3.2 Location Sketches

To measure the similarity between two sets of location tags, we adopt a mechanism called "Shingling" from the area of text mining/fingerprinting, which originally was used to detect (almost-)identical documents. The intuition is that we derive a sketch from a document such that the similarity between sketches will be high with a high probability when two documents are close to identical. Rather than viewing documents as ordered lists, we consider sets by simply reducing each document to a canonical sorted collection of elements. In this paper, the shingling process is shown in Figure 2.

We define the following concepts adopted from document shingling:

Definition 1 *k-shingle: a k-tuple which consists of k consecutive elements of a set D, which is presented as a list of sorted elements. We define the k-shingling of a set D to be the set of all unique k-shingles of D, $S_D = \{s_1, s_2, \ldots, s_n\}$*

For example, the 3-shingling of set { step, on, no, pets, } is the set {(no, on, pets), (on, pets, step) }.[1] It is not hard to see that nearly identical sets will generate nearly identical shingling. Furthermore, each unique shingle can be indexed by a numerical unique id (UID). By shingling we convert a set D into S_D, a set of uids. We reduce similarity testing on sets to similarity testing on shinglings.

The "resemblance" between sets A and B is defined by $r(A, B) = \frac{|S_A \cap S_B|}{|S_A \cup S_B|}$ Note that the resemblance definition is different than classical Jaccard index [18] of A and B, which is $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$. It is, however, the Jaccard index on shingling sets S_A and S_B. With a random permutation π: $\{0,1\}^n \rightarrow \{0,1\}^n$, it is not hard to see that

$$\Pr[min\{\pi(S_A)\} = min\{\pi(S_B)\}] = r(A, B) \ ,$$

allowing us to reduce the similarity test to an equality test.

UID generation and random permutation together can be accomplished by using cryptographic hash functions h: $\{0,1\}^* \rightarrow \{0,1\}^n$ (In [9], Rabin fingerprinting is applied). We can therefore save the permutation step, since $min\{\pi(S_A)\}$ and $min\{S_A\}$ will have the same distribution.

Theorem 1. *When Jaccard index between A and B is approximately 1, $min\{S_A\}$ and $min\{S_B\}$ are identical with high probability.*

Proof. With the union bound, $\Pr[min\{S_A\} = min\{S_B\}] = r(A, B) = \frac{|S_A \cap S_B|}{|S_A \cup S_B|} \geq 1 - k(1 - J(A, B)) \approx 1$

Now we define $min\{S_A\}$ as the sketch of S_A, denoted as t_A in Figure 2. If users A and B have similar sets of location tags, then with high probability $t_A = t_B$, and if their location tag sets are distinct, then with all but negligible probability, $t_A \neq t_B$. Thus similarity testing can be achieved by equality testing on sketches.

3.3 Private Equality Test

To test proximity, we test equality of sketches. Surely being aware of privacy issues, users doesn't want to reveal their sketches nor will they trust a third-party to carry out the operation for them. Fortunately it is possible for multiple parties to test equality privately. For example, we can utilize any of the existing solutions to Yao's millionaire problem [26]. Private equality tests (PET) have been extensively studied in the literature.

In [22], Narayanan et al. discussed two protocols for PET, that are particularly well-suited to the mobile phone with online social network setting:

– **Synchronous PET.** Based on additive homomorphic ElGamal encryption, Alice sends an encryption $E(a)$ of her input a to Bob. In turn Bob derives $E(s(a - b))$ from $E(a)$, with random s and his input b only. If $a = b$, Alice will decrypt 0 from $E(s(a-b))$. Otherwise Alice gets a non-zero random number after decryption. The protocol requires both parties to be online.

[1] Note that we sort the set to get the sequence (no, on, pets, step)

– **Asynchronous PET with Oblivious Server.** Closely related to multi-party secret-sharing, this protocol assumes no party is colluding with any other. It allows tag submission and actual tag equality testing to happen at different times. This protocol enjoys better efficiency over the synchronous one at the cost of the additional security assumption of a non-colluding server. Essentially, the protocol assumes Alice and Bob share two keys $k_1, k_2 \in_R \mathbb{Z}_p$ and Bob and the server share a key $r \in_R \mathbb{Z}_p$. Bob first sends the server $m_b = r(b + k_1) + k_2$. Later, when Alice wants to test equality, she sends the server $m_a = a + k_1$, and the server responds with $m_s = r \cdot m_a - m_b = r(b - a) - k_2$. Alice computes $m_s + k_2$, which will be 0 if $a = b$ and a random element of $\mathbb{Z}_p$ otherwise.[2]

Originally in [22] PET protocols were used to test whether Alice and Bob were within the same location on a map divided into overlapping hexagonal grids, assuming Alice and Bob are honest. Location tags (if unpredictable and reproducible) would enhance the security because neither Alice nor Bob could gain anything by lying about their location. However, since wireless nodes in close proximity will generally see similar but not identical traffic (due to noise and physical location relative to the base station), Alice and Bob were forced to use a much more expensive threshold private set intersection protocol to determine if their location tags are similar.

The advantage of PET over private set intersection protocols is efficiency. When the sets under consideration are as large as several hundreds (or thousands) of elements, which is the case with location tags, PET outperforms PSI by several orders of magnitude. With location sketches, we achieve the "best of both worlds," allowing the use of a simple PET with unpredictable location sketches.

4 Cellular Networks

The use of cellular phones is already pervasive with 5 billion users worldwide in 2010 [2]. A side effect of the protocols currently in use is that base stations are constantly broadcasting the unique or pseudorandom identifiers of mobile stations they are trying to contact. In this work, we focus on the GSM network with over 3.5 billion worldwide subscribers in 2009 [1], but the techniques are applicable to other cellular networks with broadcast paging channels similar to GSM. Since the paging traffic depends on the mobile stations (phones) being served in a geographic area, the paging channel will be different for phones in different LACs, but similar in the same LAC. Since an LAC can cover areas of up to 100km^2, it is useful to have another test to determine proximity with higher granularity. The Immediate Assignment message traffic is specific to a BTS since it is an allocation of radio resources from that tower. Devices camped on different towers will observe different Immediate Assignment message traffic. In an urban environment, we have observed that the typical range of a BTS is around 1km^2. We use those broadcast messages at the LAC and base station level to increase the area covered by location tags.

4.1 Infrastructure overview

For the purposes of this work, we can view a typical cellular network as being composed of a number of towers (BTS) belonging to an LAC and connected to a core network.

[2] [22] includes additional information to pseudorandomly generate and rotate the single-use shared keys r, k_1, k_2.

That central network contains a location register (HLR) that keeps track of each mobile station's last known location. The cellular network is then connected to the regular phone Public Switched Telephone Network (PSTN) system [5].

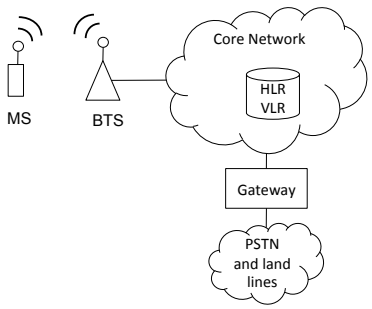

Fig. 3: Overview of a cellular network connected to the PSTN

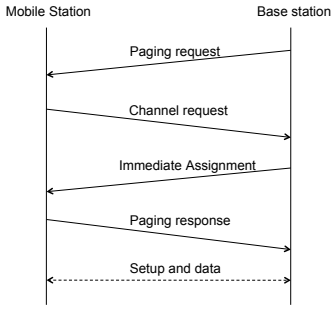

Fig. 4: Sequence diagram for the air interface between the MS and the BTS

Most of the messages between a BTS and a mobile station including voice and data transmissions, are done with frequencies and codes unique to that BTS-mobile station pair [4]. However, there are dedicated broadcast channels that all mobile stations have to listen to. In particular, the broadcast paging channel downlink is used to notify a mobile station that it needs to contact the BTS [4]. Mobile stations tune or camp on a particular frequency for their service providers and are able to hear all the pages being issued in the LAC as well as Immediate Assignment messages allocating radio resources for a particular BTS. Each paging request message contains the unique identifier of the intended destination. The set of identifiers are unique per geographic region due to the set of mobile stations in that region. Similarly, the Immediate Assignment traffic, especially the time at which those messages are broadcasted is unique by BTS as observed by mobile stations camped on those base stations.

4.2 Incoming Call protocol

The logical flow for the radio interface in a GSM network during an incoming call works as follows [4]. The BTS attempts to find the mobile station over the broadcast paging channel downlink by issuing a paging request with the mobile station's identifier [4], which should be the TMSI, but can also be the IMSI. Upon receiving the paging request and matching the identifier, the mobile station will contact the BTS over the random access channel uplink which is separate from the downlink channel. The BTS will then indicate the frequency and code for the mobile station with an immediate assignment message, possibly over the same paging channel downlink. The mobile station then responds with a paging reply over another random access uplink. The rest of the protocol allows the mobile station and the BTS to negotiate the security level and other setup parameters, before data (text or voice) can be transmitted. The initial protocol proceeds as shown in Figure 4.

4.3 Paging request messages

The paging channel carries different messages including System Information, Immediate Assignment and Paging Requests. The paging requests can be of 3 types. By far the

most common paging requests are of type 1 that allow a single or two mobile identities to be paged per message [4] (clause 9.1.22). Those paging requests are issued for every call or text message being sent to a mobile station within range of the BTS. Due to its frequent use and unique traffic pattern depending on the mobile stations served in the area, the paging channel offers a good medium that is unique in time and location. Paging requests are broadcast to every mobile station within a location area. The location area consists of several nearby BTSes. In contrast, immediate assignments are issued locally by an individual BTS. In other words, two mobile stations belonging to the same location area will hear almost the same paging requests. If they are not covered by the same BTS, they will hear completely different immediate assignments.

4.4 Location tag using cellular networks

In our evaluation in section 6, we observe that the traffic on the paging channel is a perfect candidate for LAC-level proximity testing: devices in the same LAC see very similar sets of TMSIs while devices in different (neighboring) LACs see disjoint sets. Similarly, devices camped on the same BTS hear the same Immediate Assignment traffic, but devices camped on different BTSes do not. We thus built our location tag algorithm to compare the conditions monitored by two different mobile stations.

We note that it is possible for an attacker to use high-powered directional antennae to eavesdrop on IA and PCCH traffic over longer distances. Clear lines of sight to the towers will likely be required. However, this does not guarantee that the victim's phone will be camped on that same tower, forcing the attacker to record all possible towers in the area. Moreover, interference from nearby towers using the same frequency is possible, reducing the ability to effectively eavesdrop on the target tower. In any case, this enforces that the attacker must have a device that is in physical proximity to the victim.

5 Location Sketch Privacy

Generally speaking, location sketch privacy is indeed location tag privacy. As Narayanan et al. point out, location tags should meet two key requirements as follows:

- **Reproducibility.** Two measurements taken in the same place and same time produces two almost-identical tags t_1 and t_2.
- **Unpredictability.** Without presence at a certain location and time, a malicious party should not be able to produce the tag for that location and time. Note we require location tags to be varying with time otherwise an adversary can pre-compute all location tags and a brute-force attack can reveal the victim's location trajectory.

The cellular mobile network broadcasts paging request messages through its base stations to alert the target mobile phone in the case of an incoming phone calls or text messages. In addition, each tower broadcasts the allocation messages to mobile stations requesting radio resources. Thus, two phones within the same location area should hear near-identical paging requests. If those phones are listening on the same tower, they will also hear near-identical immediate assignment message traffic. Heuristically, like the WiFi channel in [22], the GSM paging channel is a rich source of location tags.

Narayanan et al. reduce the similarity test to the private threshold set intersection (PTSI) problem [22]. In PTSI, Alice and Bob will execute a protocol that returns '1'

(as success) when the set A and B have an intersection C of size $> t'$, and returns '0' (as failure) when C is of size $< t$. Note that we require $t' > t$. When $|C| \in [t, t']$, we expect the probability of success to gradually decrease from 100% to 0%. Here we apply shingling technique to accomplish the similarity test.

5.1 Shingling and Unpredictability

In the seminal work by Broder et al. [9], the shingling technique was introduced to give a binary answer on whether two documents (as web page content) are nearly identical. Decomposing a document into a set of k-shingles is called k-shingling. Since we are interested in comparing two sets of paging requests, the order of the requests makes no difference. In one time epoch, we first hash each recorded paging request to a numerical ID, sort them and then apply shingling to them.

One issue on unpredictability is the entropy of location tags. Among paging requests, we use TMSIs which are 32-bit IDs locally randomly assigned to a mobile phone. These IDs display some redundancy but still retain 24 bits of entropy. Thus with k-shingles, the location sketch presumably has $24k$-bits of entropy. With large enough k, the adversary cannot brute-force test the tag and the proximity before the victim moves to the next location. However, with larger k, reproducibility is decreased. With the same level of differences between two tag sets, a larger k will create more unique shingles and decrease the resemblance between shinglings. A proper choice of k will thus be needed to balance the tradeoff.

As described above, shingling the paging channel provides a certain level of time sensitivity because intuitively it's rare for a k-shingle to appear twice at different time epochs. However, exceptions do occur; some subscribers receive frequent calls or text messages and may produce many paging requests for the same TMSI over time. To cope with such issues, we append timestamps to each paging request so that the same paging request at different times will be tagged with a different UID. Because different phones will record paging requests in slightly different times, we restrict the timestamps to 10-second granularity to strike a balance between unpredictability and reproducibility.

5.2 Shingling and False Positive

A false positive of location promixity happens when two phones, located at two different LACs, derive the same location sketch. Two types of cases may have led to false positives: a hash collision by two different shingle, or sharing the same shingle which produces the location sketch. Notice those cases are necessary, rather than sufficient, conditions of false positives.

We are assured that either condition will not happen in practice, neither does false positive. With random inputs, we can compute the probability of collisions with birthday attack. With m k-shingle and SHA-1 funciton, the probability of having one hash collision is roughly $p = m^2 \cdot 2^{-161}$. In practical the collision probability should be very small. Meanwhile, the probability of producing a duplicate k-shingle across LACs is also low. Without timestamp, the probability of having one duplicate k-shingle is 2^{-24k}. It is such a rare event, let alone the case in which this k-shingle eventually yields the location sketch for both LACs. The timestamps make duplicate shingles even less probable. In conclusion, it is essentially impossible to yield false positive with TMSI shingling.

5.3 Reproducibility Boosting

Reproducibility only captures the desired outcome when two measurements are nearly the same but it doesn't define how two measurements differ when they are taken at different times. With shingling, the probability of testing positive for proximity is p when the fraction p of two shinglings are common elements. Also, when two measurements share a fraction f of common readings, with $f \approx 1$, the fraction of shingles in common is at least $1 - k(1 - f)$ in the worst case, by the union bound. For instance, even when two phones see 90% common values, they would produce the same 3-shingle sketch at least 70% of the time. This reduction in accuracy is undesirable, since it would require two mobile devices to be closely synchronized in time and view of the network.

However, we can quickly boost the probability by repeating the protocol multiple times. After m repetitions, the probability of getting at least one positive increases to $1-k^m(1-f)^m$. The boosting effect is shown in Figure 5. Such boosting only requires a small value of m to significantly weaken the required synchronization between phones. Therefore, we modify our protocol so that users compute m sketches for each epoch (partitioned evenly into m sub-epochs) and execute m individual PETs. If there is at least one positive, they conclude they are co-located.

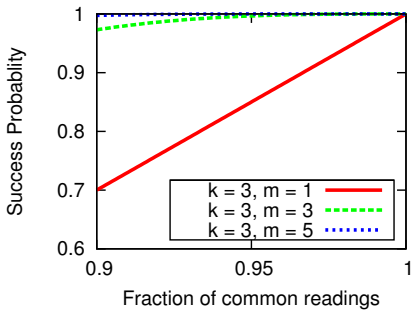

Fig. 5: The boosting effect on m executions with k-shingling

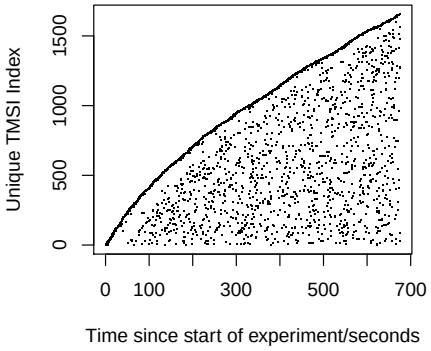

Fig. 6: TMSI requests recorded in an 11-minute period

6 Evaluation

6.1 Experiment Data Collection

To test our design, we used modified Motorola C118 cell phones connected to laptops and logging the paging channel traffic at varying distances. On our system, we ran the open source baseband project OsmocomBB [3]. We used the custom firmware from that project to reflash the phone, thereby acting as the layer 1 of the GSM communication stack. Using a serial link, the phone relays each message heard to a laptop running the upper layer 2 and 3 of the GSM stack. By default, upon startup, this system would scan all available frequencies, and select the one with the strongest signal. We configured our two cell phone-laptop systems to listen on the same GSM mobile network operator, but let the phone choose the appropriate frequencies based on the RSSI level. Once the system was started and selected the appropriate frequencies, it would start to listen on the broadcast paging channel (PCCH). A normal phone would ignore all paging

requests and immediate assignment messages that were not intended for it. In our case, we made a small modification to the layer 3 of the system to log all those messages heard on the PCCH.

For our experiments, we selected the largest GSM cellular network service provider in our area and we listened to the paging channel in several geographic locations. We had two devices listening simultaneously at different distances apart, namely 1m, 100m and 7km. For the experiment involving the largest distance apart, we observed that the devices were reporting cells with a different Location Area Code (LAC) for the chosen operator. Devices located close together report the same LAC and if very close, they choose the same frequency. The paging request logs were recovered from the laptops and filtered for the TMSIs. Figure 6 shows a plot of unique TMSI we heard over a period of 700 seconds. A unique TMSI heard multiple times will appear as multiple dots horizontally on the plot. We have three datasets: Room (1m), Campus(100m) and and Far-away (7km). Each of them consists of two traffic logs from two systems. Each log keeps a 10-minute record of the paging channel traffic, including the paging requests and immediate assignment messages.

6.2 Shingling Experiment

At the shingling stage we experiment with multiple set of parameters. We choose the epoch time to be 1 minute long and use SHA-1 as the hash function. The shingling parameter k is chosen to be 3, 4 and 5. To study the reproducibility, we observe how the clock offset between two mobile phones affects the resemblance. For each trial, we take a measurement of phone 1 at a random time $t1$ and one of phone 2 starting at $t1 + \delta$ as input to our system. We repeat the trial until all possible $t1$ values are exhausted. and record the empirical probability of proximity detection success, at offset δ. With different δ, we plot the probability of success versus the clock offset. The plots for the Campus dataset and Room dataset with different values of k are shown in Figure 7. The plots for the Room dataset display a very similar behavior and the plot for the Far-away dataset is uninteresting as expected; the probability is always zero no matter what the offset is.

As Figure 7 confirms, larger k values tend to decrease reproducibility because larger k values are more sensitive to the set difference. Additionally, each phone observes approximately 4 paging requests per second. As a result, reproducibility quickly drops as the clock offset increases. It quantitatively matches with the analysis in Section 5. We choose $k = 3$ as it produces the best reproducibility while keeping the entropy at an acceptable level (72 bits).

For reproducibility boosting, we use $m = 5$. Here the epoch time is 5 minutes and we divide it into 5 sub-epochs, each 1 minute long. Recall that with boosting, proximity detection returns true when at least one PET returns true. As a result, we again plot the probability of success versus the clock offset on both dataset Room and dataset Campus in Figure 8. With much better success rate, we only need to loosely synchronize all phones. As the distance between two phones increases, the success rate is more sensitive to the clock offset.

6.3 Fine Proximity Test

Proximity testing based only on paging requests is limited to determining whether two phones are located in the same location area (LAC), which can be as large as 100 km^2.

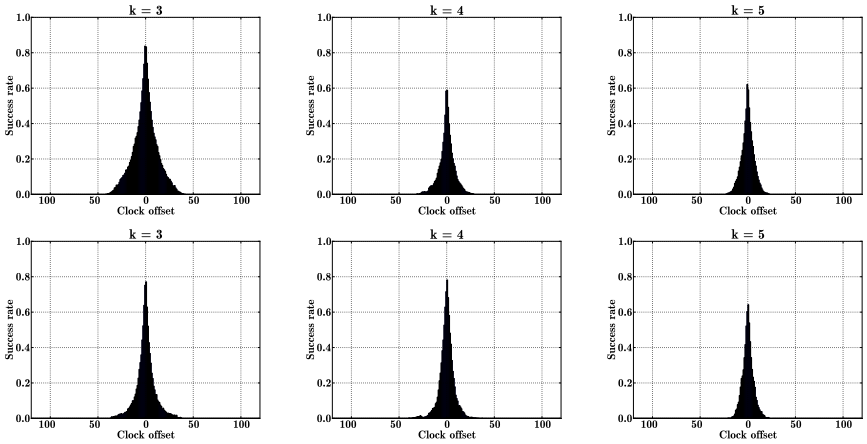

Fig. 7: First row is the plots on dataset Room, and the second row is those on dataset Campus. We omit the plots on dataset Far-away because they are simply blank ones. As k increases, resemblances drops faster and faster as the offset drifts from zero. Also even when the offset is smaller than 1 second. the resemblance is no more than 85% for either dataset.

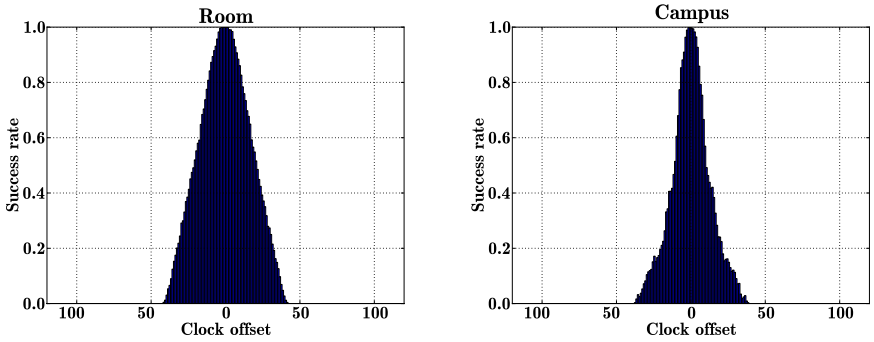

Fig. 8: With boosting parameter $m = 5$, shingling parameter $k = 3$ and 5-minute epoch time, the probability of success is almost 1 within 3-second offset for both datasets.

On the other hand, each base station covers a relatively contant area with a radius between 100m and 1000m, allowing us to determine proximity more precisely. As a proof of concept, we utilize IA messages as location tags. We choose $k = 3$ and $m = 5$, and repeat the shingling experiments. The result is shown in Figure 9. It is clear that with tower-specific location tags we can achieve finer-grained proximity testing. However, we want to point out that the unpredictability of IA messages is not as strong, because of non-random channel assignment and some other issues. We will discuss this further in Section 7.

7 Discussion

7.1 Shingling with PTSI

Broder et al. in [8, 9] use shingling and threshold set intersection together to identify nearly identical documents. For each document, the smallest k elements are chosen. If two documents have resemblance p, the (lower-bound) probability that they share t common top-k elements can be calculated. With proper choices of (k, t) the probability function acts very similar to a high-pass filter, i.e. only highly resemble documents

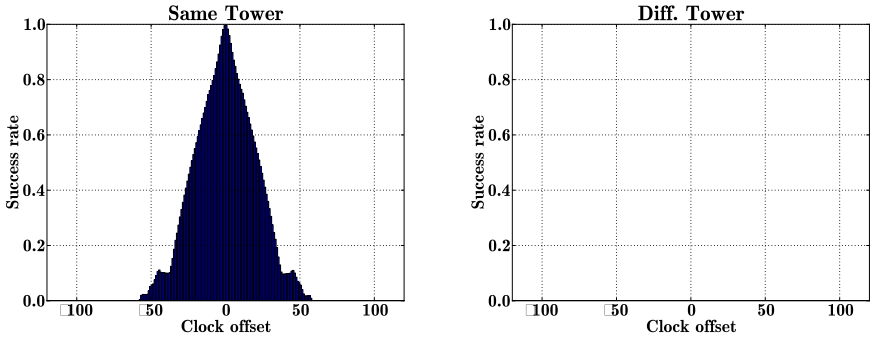

Fig. 9: The figure on the left shows two phones in the same room have large probability of testing positive for proximity while two phones in the same campus under different base stations will not be considered to be close at all, as shown by the figure on the right.

should have an intersection of size $> t$ with a high probability. In practice k is usually as small as 6, and t is as small as 2. In this way, it is possible to only keep the k shingle for each document as a sketch and execute a private set intersection protocol to figure out if two documents are near-identical.

With the same idea, shingling can be adopted to improve the efficiency of private threshold set intersection on location tag proximity. Now we shrink each location tag set to a k-element sketch set so the communication and computation cost can be greatly improved as well.

7.2 Unpredictability of IA messages

We have shown that proximity granularity can be controlled by different types of paging content. Specifically we have also shown that we can consider messages that are local to individual base stations, such like Immediate Assignment (IA) messages, to be used as location tags for finer granularity. We also note that utilizing IA messages as location tags has several issues. Unlike paging requests, IA messages has limited randomness. Base stations assign channels in a somewhat predictable way. Each available channel has a distinct set of parameters allowing passive attackers to enumerate the possible channels. When there is little traffic we will frequently see same IA messages due to the fact that same channel is reused. A preliminary computation shows the channel choices sent by the BTS on a university campus have entropy ranging from 8 bits to 16 bits. We will continue to investigate the security of IA messages and other resources for tags.

7.3 Practical considerations

The system we used for our measurement used a modified GSM layer 3 that passively observes broadcast paging requests on the PCCH. Commercially available GSM phones also hear the same traffic, but their implementation of the layer 3 simply drops any paging request that is not intended for that particular device. For our system to work, we only need a modification in the layer 3 implementation to add the ability to record paging requests on the PCCH traffic for short periods of time. While small, this change will probably require an update of the baseband firmware on the phones in order to support our protocol. We note that there are no changes required at layers 1 and 2 of the GSM protocol stack.

Baseband firmware updates have been deployed on iPhone and Android smartphones as bug fixes are rolled out. Indeed, a number of baseband updates from Apple were intended to limit the ability to jailbreak the iPhones. With this update infrastructure in place, it would be possible to package support for our protocol in one such baseband firmware update. However, baseband updates for feature phones are far less common, making a migration of the baseband firmware to support our protocol less likely.

8 Conclusion

Location proximity testing has become an important social networking application. Meanwhile, concern over location privacy has also been growing. To address these concerns, Narayanan et al. [22] recently proposed Private proximity testing by location tags. We build on their work by successfully capturing location tags based on the GSM cellular network, which covers a larger area with greater reliability. Moreover, we describe a novel use of de-duplication shingling to test location tag similarity by private equality testing, a simple and efficient cryptographic primitive. We have developed prototypes that demonstrate the effectiveness of our approach. We thus describe the first privacy-preserving proximity system that builds on cellular network location tags. The use of shingling to reduce threshold set intersection to equality testing may be of interest to other private set operation protocols and is an open question for future research.

References

1. Gsm world – market data summary. `http://www.gsmworld.com/newsroom/market-data/market_data_summary.htm`, 2009.
2. United nations international telecommunication union sees 5 billion mobile subscriptions globally in 2010. `http://www.itu.int/net/pressoffice/press\_releases/2010/06.aspx`, 2010.
3. The osmocombb project – open source gsm baseband software implementation. `http://bb.osmocom.org/`, 2011.
4. 3GPP. Mobile radio interface layer 3 specification. TS 04.08, 3rd Generation Partnership Project (3GPP), January 2004.
5. 3GPP. Network architecture. TS 23.002, 3rd Generation Partnership Project (3GPP), March 2011.
6. M. J. Atallah and W. Du. Secure multi-party computational geometry. In *Proceedings of the 7th International Workshop on Algorithms and Data Structures*, WADS '01, pages 165–179, London, UK, 2001. Springer-Verlag.
7. A. R. Beresford and F. Stajano. Location privacy in pervasive computing. volume 2, pages 46–55, Piscataway, NJ, USA, January 2003. IEEE Educational Activities Department.
8. A. Broder. Identifying and filtering near-duplicate documents. In R. Giancarlo and D. Sankoff, editors, *Combinatorial Pattern Matching*, volume 1848 of *Lecture Notes in Computer Science*, pages 1–10. Springer Berlin / Heidelberg, 2000. 10.1007/3-540-45123-4_1.
9. A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. *Computer Networks and ISDN Systems*, 29(8-13):1157 – 1166, 1997. Papers from the Sixth International World Wide Web Conference.
10. E. De Cristofaro and G. Tsudik. Practical private set intersection protocols with linear complexity. *Financial Cryptography and Data Security*, pages 143–159, 2010.

11. D. Di Qiu, S. Lo, and P. Enge. Robust location tag generation from noisy location data for security applications. *The Institute of Navigation International Technical Meeting*, 2009.
12. R. Fagin, M. Naor, and P. Winkler. Comparing information without leaking it. *Commun. ACM*, pages 77–85, 1996.
13. M. J. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *Theory and Application of Cryptographic Techniques*, pages 1–19, 2004.
14. M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, MobiSys '03, pages 31–42, New York, NY, USA, 2003. ACM.
15. C. Hazay and Y. Lindell. Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries. In *Proceedings of the 5th conference on Theory of cryptography*, TCC'08, pages 155–175, Berlin, Heidelberg, 2008. Springer-Verlag.
16. C. Hazay and K. Nissim. Efficient set operations in the presence of malicious adversaries. In P. Nguyen and D. Pointcheval, editors, *Public Key Cryptography PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 312–331. Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-13013-7_19.
17. L. E. Holmquist, J. Falk, and J. Wigstrm. Supporting group collaboration with interpersonal awareness devices. *Personal and Ubiquitous Computing*, 3, 1991.
18. P. Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.
19. S. Jarecki and X. Liu. Efficient oblivious pseudorandom function with applications to adaptive ot and secure computation of set intersection. In O. Reingold, editor, *Theory of Cryptography*, volume 5444 of *Lecture Notes in Computer Science*, pages 577–594. Springer Berlin / Heidelberg, 2009. 10.1007/978-3-642-00457-5_34.
20. S. Jarecki and X. Liu. Fast secure computation of set intersection. In J. Garay and R. De Prisco, editors, *Security and Cryptography for Networks*, volume 6280 of *Lecture Notes in Computer Science*, pages 418–435. Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-15317-4_26.
21. L. Kissner and D. Song. Privacy-preserving set operations. In V. Shoup, editor, *Advances in Cryptology CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 241–257. Springer Berlin / Heidelberg, 2005. 10.1007/11535218_15.
22. A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, and D. Boneh. Location privacy via private proximity testing. In *Network and Distributed System Security Symposium*. Internet Society, Feb 2011.
23. D. Qiu, S. Lo, P. Enge, D. Boneh, and B. Peterson. Geoencryption using loran. *The Institute of Navigation International Technical Meeting*, 2007.
24. R. Shokri, C. Troncoso, C. Diaz, J. Freudiger, and J.-P. Hubaux. Unraveling an old cloak: k-anonymity for location privacy. In *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society*, WPES '10, pages 115–118, New York, NY, USA, 2010. ACM.
25. M. Terry, E. D. Mynatt, K. Ryall, and D. Leigh. Social net: using patterns of physical proximity over time to infer shared interests. In *CHI '02 extended abstracts on Human factors in computing systems*, CHI EA '02, pages 816–817, New York, NY, USA, 2002. ACM.
26. A. C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, SFCS '82, pages 160–164, Washington, DC, USA, 1982. IEEE Computer Society.
27. G. Zhong, I. Goldberg, and U. Hengartner. Louis, lester and pierre: three protocols for location privacy. In *Proceedings of the 7th international conference on Privacy enhancing technologies*, PET'07, pages 62–76, Berlin, Heidelberg, 2007. Springer-Verlag.