How to Protect Query and Report Privacy without Sacrificing Service Quality in Participatory Sensing

Meng Li[1], Fan Wu[2], Guihai Chen[2], Liehuang Zhu[1], Zijian Zhang[1]*
[1] Beijing Engineering Research Center of Massive Language Information Processing and Cloud Computing Application,
School of Computer Science and Technology, Beijing Institute of Technology, China
[2] Shanghai Key Laboratory of Scalable Computing and Systems,
Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
hfmengli@bit.edu.cn, {fwu, gchen}@cs.sjtu.edu.cn, {liehuangz, zhangzijian}@bit.edu.cn

***Abstract*—The ubiquity of mobile devices has brought forth the concept of participatory sensing, whereby people can collect and share data from ambient environment for the benefit of themselves or community. To encourage participation of all stakeholders and guarantee system functionality, a privacy-preserving participatory sensing system should be established to hide querier's and participant's sensitive information(e.g., interest, location and content). Meanwhile, it is also imperative for server to provide an accurate and quick service(match the "need" with "supply" and retrieve the desired result from collected data set) for queriers when queries and reports are encrypted to protect privacy. In this paper, we propose Query and Report privacy-preserving protocol(QueRe) in participatory sensing system aiming to protect the query privacy and report privacy without sacrificing the service quality. Security analysis and performance simulation show our method achieves superior performance in privacy protection and service quality. To the best of our knowledge, our work is first attempt for protecting query privacy and report privacy while considering server's service quality.**

I. Introduction

Now over one billion people carry smart phones. They not only serve as the key computing and communication mobile device(e.g., smart phones, tablets, and smart wristbands), but it also comes with a rich set of powerful embedded sensors(e.g., cameras, microphones, accelerometers). Coupled with ubiquity, developments in smart phone technology have paved the way for designing new paradigm for accomplishing large-scale sensing. Different from the typical wireless sensor networks(WSNs), the idea of participatory sensing(PS) [1] has enabled the emergence of personal and group sensing applications in which participants collect and share information in their environments.

Applications of PS system range from people centric scenarios to environment centric scenarios. In people centric scenarios, participants can monitor and document health-related issues, such as diet behaviors [2] and individual exposure and impact to air pollution [3], [4], physical activities [5] and sport experiences [6], [7]. In environmental centric scenarios, participants can crowdsource the data about noise pollution [8], urban air pollution [9] and bus arrival times [10].

It is undeniable that opportunities and benefits provided by participatory sensing can significantly revolutionize a large number of existing applications and ultimately impact our everyday lives. However, querier and participant may raise a question: if I initiate a query or upload my report, will or will not my privacy [11] be revealed to others(e.g, server)? From the participatory sensing server's perspective, correct and real-time sensory data are vital to any sensing task, and their absence or deficiency will endanger the success of participatory sensing systems. Therefore, an appealing query and report privacy protection mechanism is needed to increase the users' participation and ensure the durability of the participatory sensing system.

In participatory sensing, the privacy of participants include location, time, identity, preference and health condition which can be directly obtained by spatiotemporal information, pictures and sound samples or indirectly inferred from their frequent visits or same choices over multiple contributions to sensing tasks [12], [13]. Common techniques to protect privacy in participatory sensing are pseudonyms [14], [15], k-anonymity [14], [16]–[19], [22]–[24] and cryptography [15], [17], [19]–[21].

For querier, if her choice of an interest and location are not well protected, say, a history of requests is accumulated by server, then potentially her destination, activity pattern and even her identity will be inferred. Similarly for participant, if his choice of an interest and location are not well protected, his historical trajectory, active region and even his identity will be inferred. Therefore, only when privacy concerns(e.g., choice of an interest, location) are eliminated, should more queriers and participants join in the participatory sensing system and do their parts right.

[25] presented private information retrieval(PIR) and k-anonymity to protect query privacy [26]. However, they did not consider report privacy and make no effort to hide the users identity from the location based service. [21] adopted bilinear mapping to protect query privacy and report unlinkability. However, a secret z is shared among mobile nodes and queier's interest and participant' interest can be inferred if ciphertext-only attack is launched and two encrypted message are matched.

Meanwhile, we have not forgotten the motivation of PS system which is harness the power of crowds to provide useful information for information-sharing, decision-making, personalized recommendation, trend forecasting, etc. Querier

*Corresponding Author.

978-1-4673-8590-9/15/$31.00 ©2015 IEEE

and participant can choose to opt in or opt out leverage what kind of useful can be promised and acquired. Hence, service quality is another factor we can never overlook if we believe in the functionality of participatory sensing system.

Intuitively, a good service provided by server for querier is an accurate and quick service. In this work, we use symmetric encryption to match querier's "need" with participant's "supply", which means the matching process is run over ciphertexts. This raises a challenge for server if it is expected to locate the exact result querier requested in an accurate and quick way. Note that asymmetric encryption is adopted to protect query privacy and report privacy, since shared key among participants will violate the report privacy which we will discuss in section V.

In this work, we propose QueRe: query and report privacy-preserving protocol to protect query privacy for querier and report privacy for participants(reporters) without sacrificing the quality of service which is requested by querier. To summarize, the contributions of our work include:

1) Stronger query privacy and report privacy are added when compared with previous work.
2) An accurate and quick matching process over ciphertexts is proposed to guarantee service of high quality.

The rest of this paper is organized as follows. In section II, we discuss related works. Section III describes preliminaries. The detailed design of QueRe: query and report privacy-preserving protocol is presented in section IV. In section V and VI, we analyze the privacy and performance of QueRe. Finally, we conclude this paper in section VII.

II. Related Work

A few works have been focusing on query privacy using different techniques or in different applications.

[25] presented a technique for private information retrieval that allows a user to retrieve information from a database server without revealing what is actually being retrieved from the server. Their algorithm of using a variable-sized cloaking region divided into VHC cells resulted in location privacy. However, they did not consider report privacy and make no effort to hide the users identity from the location based service.

Bilinear mapping was adopted to protect query privacy and report unlinkability in [21]. Each querier registered herself with registration authority and obtained a signature to make a request and mobile nodes reported with a secret and public key. However, the secret z is shared among mobile nodes and queier's interest and participant' interest can be inferred if ciphertext-only attack is launched and two encrypted message are matched. Moreover, one additional hash function is needed to compute decryption k.

[27] proposed new metrics to measure users' query privacy taking into account user profiles. Furthermore, we design spatial generalization algorithms(for k-ABS, α-USI, β-EBA and γ-MIA) to compute regions satisfying users' privacy requirements expressed in these metrics. [28] defined and addressed both query and data privacy in the context of Urban Sensing. However, a secret key was shared between each sensor and OWN(owner of the network/querier), which is not a realistic assumption for participatory sensing.

Query dependency which can be derived from users' request history was studied in [29] and an approach was presented to compute the probability for a user to issue a query, by taking into account both users query dependency and observed requests. They also proposed new metrics incorporating query dependency for query privacy, and adapt spatial generalization algorithms in the literature to generate requests satisfying users' privacy requirements expressed in the new metrics.

[30] designed two mechanisms using mobile clouds to preserve data query privacy in mobile mashups. [31] proposed a new local data perturbation method called Aroma to protect data under differential privacy while provide query answers as accurate as possible. [32] addressed the problem of privacy preservation if the query returns the histogram of rankings and the framework of differential privacy is applied to rank aggregation.

To the best of our knowledge, our work is the first attempt to look at the protection of query/report privacy while guaranteeing server service quality in the participatory sensing context.

III. Preliminaries

In this section, we formalize: (i) the entities and operations involved in a privacy-preserving participatory sensing model, (ii) privacy assumption, and (iii) design objectives.

A. System Model

Figure 1 shows our system model, which is similar to the model in [21]. Entities in our model include:

Participants. Each participant carries a mobile device equipped with embedded sensors. They can report to server what he knows or cares about since he is looking forward to getting paid somehow by a commercial server or just helpful.

Querier For most sensing tasks, participants collaborate to achieve a common goal such that querier can request server to acquire their desired results and make more rational decisions.

Server. Server initiates sensing task by issuing a public list of interests and defining specific requirements(e.g., data type, data length, duration). Before querier can access the result of the sensing tasks, server will first match querier' "need" with participants' "supply" and send the encrypted result to querier.

Registration Authority(RA). The Registration Authority handles the application setup with server, as well as registration of queriers and participants. By generating and distributing cryptographic parameters, RA plays an important role in protecting query privacy and report privacy (which we will explain in section 3.2).

Network Provider(NP). Network provider manages the network used to collect and deliver query and sensor report (e.g., they maintain GSM and/or third/fourth generation, 3G/4G, networks). As the network provider already knows the participants' location, so involving this role does not increase the risks for the querier's or participant's privacy.

Operations in our model include:

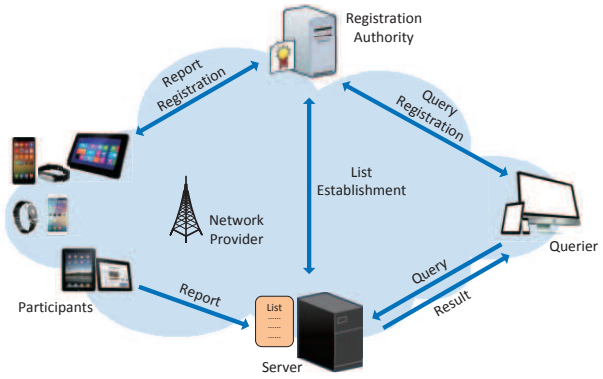

Fig. 1: Query and Report Privacy Protecting Protocol

Setup. RA and server co-establish and issue a list of interests(e.g., nearest gas station, cheapest mall, underground parking spot) and they respectively choose and generate their own cryptographic parameters(e.g., symmetric encryption algorithm, private key and public key). (RA and server should also acquire valid certificates from certificate authority.)

Participants Registration. When the list of interests are issued, participant registers his sensor-equipped device to the RA and acquires two keys corresponding to his interest and report time.

Report. Participant send his knowledge about certain interest to the server which stores the encrypted message. Also, no information about what information is collected or who/where the participant is should be revealed to the server.

Querier Registration. When the list of interests are issued, querier registers her device to the RA and acquires a pair of keys corresponding to her interest and query time(will be transformed into a time interval for computation).

Query. Queriers initiates an query to the server to obtain a specific type of result and awaits for the responses containing her desired result. No information about what information is needed or who/where the querier is should be revealed to the server.

Query Execution. Receiving query and report, server first decrypts the encrypted tag and matches reports with query subscriptions. As we designed, this should be done blindly, meaning server cannot know what information is needed/collected or who/where the querier/participant is.

List Searching. After collecting enough reports, server will have to first match a new query "need" with stored "supply" in the growing list of interest and then return the accurate report to the querier. Note that we require high matching accuracy and efficiency from the server with powerful computing capability.

List Maintaining. Remember that the pair of keys obtained by querier is only effective regarding one certain interest and time. Therefore, server needs to update the list of interest when new pair of keys are used to issue queries or when it is notified by the RA.

B. Privacy Assumption

In this paper, we adopt a realistic assumption: the server cannot be trusted and service fee is charged when querier initiates a query to server. Before entering the details of our privacy requirements, we observe that the main purpose of a participatory sensing application is to allow queriers to obtain desired result. More importantly, we aim to protect query privacy and report privacy at the same time. The definition of correctness, query privacy and report privacy are given below:

Definition 1. (Correctness) *Let Q_i be the querier of a query to the server. We say that correctness is guaranteed if upon subscribing to a query, Q_i obtains her desired result (assuming server has stored corresponding report).*

Definition 2. (Query Privacy) *Let Q_i be the querier of a query to the server. We say that query privacy is guaranteed if (i) any malicious adversary has just a negligible advantage over a random guess of the identity, location, time or query interest of Q_i, (ii) different queries originating by the same querier cannot be linked to the source and (iii) Q's interest cannot be inferred as the same as another querier's interest.*

Definition 3. (Report Privacy) *Let P_i be the owner of a report to the server. We say that report privacy is guaranteed if (i) any malicious adversary has just a negligible advantage over a random guess of the identity, location, time or report interest of P_i, (ii) different reports originating by the same participant cannot be linked to the source and (iii) P's interest cannot be inferred as the same as another participant's interest.*

Furthermore, we assume the server will not collude with other stakeholders and querier may be not acquainted with participants, yet they all can be honest-but-curious. In this paper, we do not consider the denial-of-service (DoS) attack in various protocol layers [33], [34] where the adversary prevents the querier from getting any result at all.

C. Design Objectives

In this paper, we mainly aim to protect query privacy and report privacy as defined in last subsection. Meanwhile, it is also imperative that server can still provide service of high quality when operating on encrypted queries and reports.

Besides the objective on privacy preservation, the design should also have:

1) high service accuracy: complete the matching process with a high accuracy of retrieving the result.
2) low service response time: server completes the matching process with an acceptable period of time.

IV. Query and Report Privacy-Preserving Protocol

In this section, we present the protocol QueRe: Query and Report privacy-preserving protocol that satisfies the above design objectives.

A. A Simple Case

A querier Q wishes to access the participatory sensing network for information regarding her interest and time (e.g.,

TABLE I: Key Notations

Notation	Definition
Q	querier Q
P	participant P
t	time interval
int	certain interest
$pubk_{server}$	public key of server
$prik_{server}$	private key of server
$pubk_{int,t}$	public key of a int within t
$prik_{int,t}$	private key of int within t
Tag_Q	tag computed by Q to identify interest
Tag_P	tag computed by P to identify interest
Enc	asymmetric encryption algorithm
Dec	asymmetric decryption algorithm
M	integer number reducing the size of the ciphertext
m	bit length of M
n	maximum number of results per interest
N	maximum number of interests that server can store
o	number of optional results for each querier($1 < k < n$)
b	bit length of ciphertext after AES encryption

"Nearest Nearest French Restaurant, Time Square, NYC, 1900-2100, Mar 10") in the public list. She first registers with RA and retrieves a pair of keys $< pubK_{int,t}, priK_{int,t} >$. Then she can compute an encrypted tag $AESTag_Q$ on her tag Tag_Q identifying what she needs:

$$Tag_Q = int||t, \tag{1}$$

$$AESTag_Q = AES_{AES_{prik_{int,t}}(Tag_Q)}(Tag_Q) \bmod M \tag{2}$$

and send her query $Enc_{pubk_{server}}(Tag_Q||r)$ to server, where $M(= 2^m)$ simply reduces the size of the AES tag for quicker matching, and it barely affects data retrieving or security and r is a random number.

Intuitively, a random function which is not invertible can be used like SHA-1. However, SHA-1 is not as fast as AES because AES is implemented in hardware on modern processors.

A voluntary participant P hoping to help others seeking answers to interest int first registers with RA and retrieves two keys $< pubk_{int,t}, AES_{prik_{int,t}}(int||t) >$. Then he computes his encrypted tag $AESTag_P$ identifying what he offers and send his report $< Enc_1, Enc_2 >$ to server where

$$Tag_P = int_Q||t_P, \tag{3}$$

$$AESTag_P = AES_{AES_{prik_{int,t}(Tag_P)}(Tag_P)} \bmod M, \tag{4}$$

$$Enc_1 = Enc_{pubk_{server}}(AESTag_P||r_1), \tag{5}$$

$$Enc_2 = Enc_{pubk_{int,t}}(data||t_P||r_2). \tag{6}$$

When server receives a query and a report, it first decrypts the query and the first part of report:

$$Tag_Q^{'} = Dec_{prik_{server}}(Enc_{pubk_{server}}(AESTag_Q||r)), \tag{7}$$

$$Tag_P^{'} = Dec_{prik_{server}}(Enc_{pubk_{server}}(AESTag_P||r_1)). \tag{8}$$

If $Match(AESTag_Q, AESTag_P) = 1$(after removing the two random numbers in the rear of the concatenations), then server sends $Enc_{pubk_{int,t}}(data||t||r_2)$ to Q, or $Enc_{pubk_{int,t}}(\text{"}Not\ reported\ yet\text{"}||int||t)$ is returned.

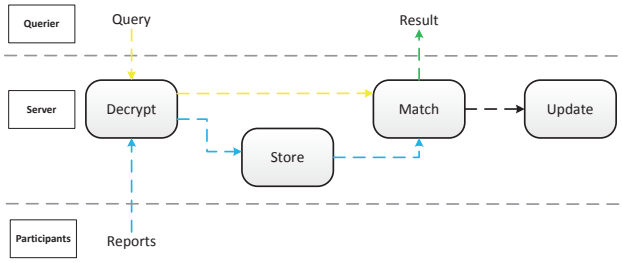

Fig. 2: Work Flow of QueRe

We require that whenever server receives a report, it will store the real tag after decryption and encrypted result for further matching. At last, Q can obtain her desired answer by decrypting $Dec_{prik_{int,t}}(Enc_{pubk_{int,t}}(data||t||r_2))$ and removing random number r_2 in the rear of the concatenation.

B. A General Case

When server has collected enough tags and encrypted results, it can answer a new query (possibly containing tags encrypted with a previous $prik_{int||t}$ not long ago) by searching the stored and growing list of real tags and encrypted result. Note that we induce an integer M to reduce the size of the ciphertext such that matching process is accelerated and bandwidth overhead is reduced.

Instead of running full search on the list of tags and results, Q can obtain her desired result through 1-out-of-N oblivious transfer (OT) [35]–[37], only if tags and corresponding results are sorted and the prior knowledge of order is shared between RA and Q. The general idea of 1-out-of-N OT is receiver only obtains one secret from sender who has N secrets and sender does not know which secret is transferred to receiver while receiver has no information about other secrets when the interaction is over.

If a match is found and multiple encrypted results are attached, server can choose to return just one result(which may not solve querier's problem) or all of them(which definitely incurs great communication overhead). Here, we let the querier have $o(o = 5)$ optional results at most.

As the reports from participants grow with time and different participants joining in the system, server will certainly have to maintain the public list of interest and the encrypted answers in database. A simple solution is to kset a upper bound to the number answers with same tag and automatically delete the least "fresh" answers.

Figure 2 shows the general work flow of QueRe.

V. Privacy Analysis

We now consider privacy properties of QueRe.

Correctness. Since each participant has $AES_{prik_{int,t}}(int||t)$, the understanding of private key of

a querier, from RA, he can use it to generate a tag as same as the querier's. Then server can find the matching tags through decryption and random number removal. Remember that integer M is induced to reduce the size of the ciphertext and we will see how this affects correctness in section VI.

Query Privacy. Definition 2 is indistinguishability-based: at a high level, given two tags and an encryption of one of these two sets of tags, no polynomial-time adversary can tell with chance better than half, which of the two sets of tags were encrypted. In other words, the adversary gains no side information from the encryption scheme.

Definition 4. (Query Security) *Consider a scheme with algorithms* (Setup, Enc, Match) *and associated message space M. Let* Adv *be a p.p.t. stateful adversary with oracle access to QueRe. Consider the following experiment.*

$\underline{\text{Exp}_{\text{Adv}}(1^k)}$:
1: $k_0, k_1, K_0, ..., K_{l-1} \leftarrow$ Setup(1^k)
2: $QTag_0, QTag_1 \leftarrow$ Adv(1^k)
3: $b \leftarrow \{0,1\}$, a random bit.
4: $c_0 \leftarrow$ Enc$(k_0, QTag_0)$ and $c_1 \leftarrow$ En$(k_1, QTag_1)$
5: Adv$(1^k) \leftarrow c_b$
6: $r_i \leftarrow$ Enc$(K_i, RTag_i),\ i \in 0, ..., l-1$
7: Adv$(1^k) \leftarrow (r_0, ..., r_{l-1})$
8: $b' \leftarrow Adv(1^k)$
9: Let 1 be the output of Match fuction.
If $b' = b$, output "Success", else output "Fail".

We say that the scheme is secure if for all p.p.t. stateful adversaries Adv, *and for all sufficiently large k*:

$$Pr[Exp_{Adv}(1^k) = \text{“}Success\text{”}] \leq 1/2 + negl(k)$$

In this security definition, the adversary Adv chooses two queries $QTag_0$ and $QTag_1$, receives an encryption of one of these at random (the bit b controls which query will be encrypted) and then tries to guess b by outputting b'. The adversary succeeds if his guess b' equals b. and we want to ensure that the attacker dos not learn anything from the scheme other than knowing matching has been found.

For preciseness, we provide the construction of the scheme here too. For simplicity, we consider security after the query and report are first decrypted into AES tags in server, because this coincides with the assumption that server cannot be trusted.

CONSTRUCTION Π
The setup algorithm Setup(1^k): Generate AES keys.
The encryption algorithm AES$(\cdot,\cdot)$:
1: For each $i \in \{0,1\}$, do:
1.1: $key_i = AES(k_i, QTag_i)$.
1.2: Compute $c_i = AES_{key_i}(QTag_i)\ mod$ M.
3: Output $c_i, i \in \{0,1\}$.

The security of our scheme relies on the standard cryptographic assumption that AES is pseudorandom permutation.

Theorem 1. *Assuming that AES is a pseudorandom permutation, our construction* Π *is a query secure scheme.*

Proof: We now prove security. We prove security through a hybrid. The hybrid replaces the AES encryption of tags with deterministic random values, based on the pseudorandom security property of AES. This results in an experiment in which the distribution of encryptions of $QTag_0$ and $QTag_1$ are statistically equal and thus indistinguishable, proving our theorem. Hybrid. The AES algorithm is replaced with random values. Concretely:

1: For each $i \in \{0,1\}$, generate a random value $rand_i$ in the ciphertext space of AES_k, with the only restriction that it preserves equality.

2: Compute $c_i = rand_i$ mod M.

3: Output $c_i, i \in \{0,1\}$.

Lemma 1. *LEMMA 2. Assuming AES is a pseudorandom permutation, for all p.p.t. stateful adversaries Adv, for all sufficiently large k:*
$Pr[Exp_{Adv}(1^k) = \text{“}Success\text{”}] \leq Pr[Exp_{Adv,Hybrid}(1^k) = \text{“}Success\text{”}] + negl(k)$.

Proof: The proof follows directly from the pseudorandom property of AES, which means that AESk is computationally indistinguishable from a random oracle. ■

■

Hence, no adversaries from the server end can tell anything between two AES tags. Note that, additional encryption using a random number has been put on AES tags such that no adversaries impersonating participants can learn anything from the communication between queriers and server. Since inputs vary with different queries, (ii) and (iii) in Definition 2 can be guaranteed.

Report Privacy. Definition 3 is also indistinguishability-based, we do not explicitly prove it here, since it is similar to the proof procedure above.

VI. PERFORMANCE ANALYSIS

When evaluating QueRe, we aimed to answer questions. First, what are the setup time for server to be ready for the first query? Second, what are the performance overheads of QueRe at server? Third, how is the accuracy of QueRe?

Our prototype of QueRe runs on two server with 2.40 GHz Intel i5-2430M CPU and 4 GB RAM. All of our experiments were performed on this testbed. For the asymmetric encryption, we use RSA and the bit length b of ciphertext after AES encryption is 1024. In each experiments behind the figures below, we took an average of a hundred runs.

A. Setup Time

We first look at the setup time for server before it answers any queries. Here, by setup time, we mean the time consumed in the collection and decryption of participants' reports. Figure 3 shows our average setup time for server to collect N reports and decrypt them to obtain AES tags(then store these tags in database). From the results we can see that setup time increases linearly with N. When this phase is complete, server can hopefully provide information to queriers and inevitably

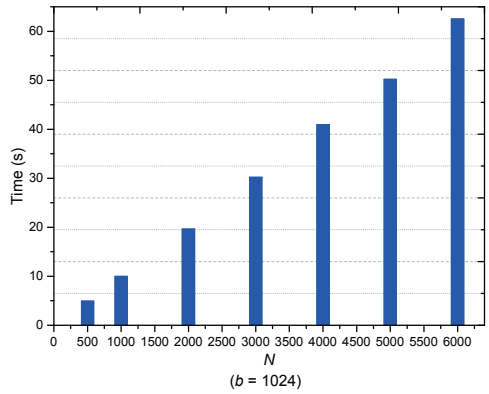

Fig. 3: Average Setup Time

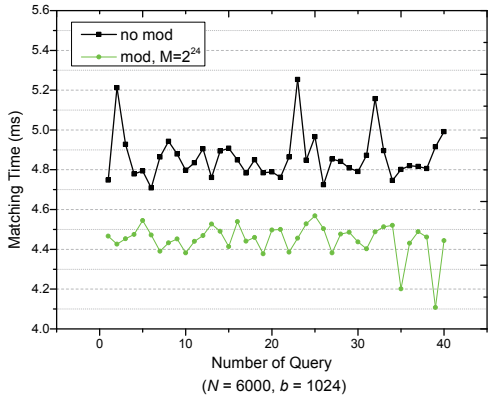

Fig. 5: Matching Time Comparison When $N = 6000$

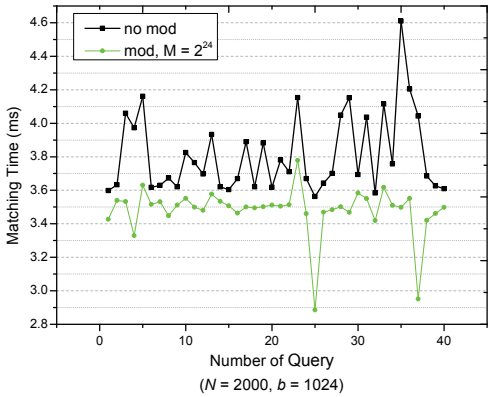

Fig. 4: Matching Time Comparison When $N = 2000$

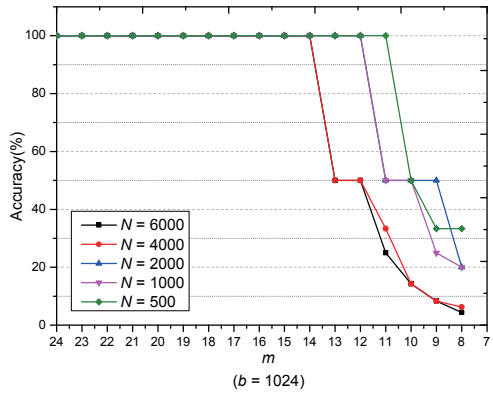

Fig. 6: Accuracy of QueRe with Different Mod

expand its database for a wide selection of service. In real scenarios, server can receive reports and begin matching at the same time in order to save time.

B. Matching Time

After the setup is finished, we see how quickly server responses to a query. Now the server has collected enough reports and obtained the corresponding AES tags. A query has been initiated to server. Server decrypts the query using its private key and begin matching with the stored tags. We conducted in two scenarios where $N = 2000$ and $N = 6000$, both of which are performed with and without mod operation. As depicted in figure 4, the matching time has dropped about 10% when the nod operation is adopted and it is worthwhile to note that the time difference between experiments without mod operation and ones with mod operation will become even bigger if much larger N is used.

C. Accuracy

To reduce the matching time of "need" with "supply", we used an integer number M to shorten the length of AES tags and then begin lookup. The advantage has already been shown in the subsection above, however, doing this will raise another question: how accurate if the AES tags are modified? By "accuracy", we mean the proportion of the exact report(assuming only one matching report has been stored in server) to answer the query in all the reports which are matched and returned. For example, in figure 6, we can see the accuracy maintains 100% until $M = 2^{13}$ which is quite an acceptable number for network transmission.

VII. Conclusion

When taking part in participatory sensing, queriers and participants instinctively want to hide their sensitive information, making the protection of query privacy and report privacy essential for functionality in participatory sensing system. The transformation on original interest and report will lead to obstacles for server in providing high-quality service. In this paper, we proposed QueRe to overcome the gap between those two problems. Security anaysis has prived that QueRe can protect query privacy and report privacy regarding our strict definitions. Performance evaluation results have shown that

with an acceptable length of modified tags our scheme can provide accurate and quick service to the queriers.

VIII. Acknowledgements

This work was supported by National Natural Science Foundation of China "NSFC"(Grant No.61272512, 61300177).

References

[1] J. A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing," in *Proc. of ACM WSW Workshops*, pp. 1–5, 2006.

[2] L. Nachman, A. Baxi, S. Bhattacharya, V. Darera, P. Deshpande, N. Kodalapura, V. Mageshkumar, S. Rath, J. Shahabdeen, and R. Acharya, "Jog falls: a pervasive healthcare platform for diabetes management," *IEEE Pervasive Comput*, vol. 6030, pp. 94–111, 2010.

[3] M. Mun, S. Reddy, K. Shilton, N. Yau, J. A. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "PEIR,the personal environmental impact report, as a platform for participatory sensing systems research," in *Proc. of ACM MobiSys 2009*, pp. 55–68, 2009.

[4] P. Zappi, E. Bales, J. H. Park, W. Griswold, and T. Š Rosing, "The Citisense air quality monitoring mobile sensor node," in *Proc. of ACM/IEEE IPSN 2012*, pp. 1–5, 2012.

[5] E. P. Stuntebeck, J. S. Davis II, G. D. Abowd, and M. Blount, "HealthSense: classification of health-related sensor data through user-assisted machine learning," in *Proc. of HotMobile 2008*, pp. 1–5, 2008.

[6] S. B. Eisenman, and A. T. Campbell, "SkiScape sensing," in *Proc. of ACM SenSys*, pp. 401–402, 2006.

[7] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T Campbell, "BikeNet: a mobile sensing system for cyclist experience mapping," *ACMTrans. Sens. Netw*, vol. 6, iss. 1, pp. 1–39, 2009.

[8] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, "SoundSense: scalable sound sensing for people-centric applications on mobile phones," in *Proc. of ATM MobiSys*, pp. 165–178, 2009.

[9] W. Sun, Q. Li, and C.-K. Tham, "Wireless deployed and participatory sensing system for environmental monitoring," in *Proc. of IEEE SECON*, pp. 158–160, 2014.

[10] P. Zhou, Y. Zheng, and M. Li, "How long to wait?: predicting bus arrival time with mobile phone based participatory sensing, in *Proc. of ACM MobiSys*, pp. 379–392, 2012.

[11] S. D. Warren and L. D. Brandeis, "The right to privacy." in *Harvard Law Review*, vol. 4, pp. 193–220, 1890.

[12] K. Shilton, "Four billion little brothers? Privacy, mobile phones, and ubiquitous data collection," in *ACM Commun*, vol. 52, iss. 11, pp. 48–53. 2009.

[13] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers, in *SIGKDD. Explor. Newsl*, vol. 12, iss. 2, pp. 74–82, 2011.

[14] L. Kazemi and C. Shahabi, "A privacy-aware framework for participatory sensing," in *ACM SIGKDD Explor. Newsl* vol. 13, iss. 1, pp. 43–51, 2011.

[15] X. L. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "ARTSense: anonymous reputation and trust in participatory sensing." in *Proc. of IEEE INFOCOM*,pp. 2517–2525, 2013.

[16] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: architecture and algorithms," in *IEEE TOC*, vol. 7, iss. 1, pp. 1–18, 2008.

[17] K. Vu, R. Zheng, and J. Gao, "Efficient algorithms for k-anonymous location privacy in participatory sensing," in *Proc. of IEEE INFOCOM 2012*, pp. 2399–2407, 2012.

[18] I. Boutsis and V. Kalogeraki, "Privacy preservation for participatory sensing data," in *Proc. of PerCom*, pp. 103–113, 2013.

[19] F. D. Qiu, F. Wu, and G. H. Chen, "SLICER: a slicing-based k-anonymous privacy preserving scheme for participatory sensing," in *Proc. of IEEE MASS*, pp. 113–121, 2013.

[20] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "PriSense: privacy-preserving data aggregation in people-centric urban sensing systems," in *Proc. of IEEE INFOCOM*, pp. 1–9, 2010.

[21] E. De. Cristofaro and C. Soriente, "Short paper: PEPSI: Privacy-Enhanced Participatory Sensing Infrastructure," in *Proc. of ACM WiSec*. pp. 23–28, 2011.

[22] C. Dwork, "Differential privacy," in Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (Eds.) ICALP 2006. LNCS, vol. 4052, pp. 1C12. Springer, Heidelberg, 2006.

[23] Q. Li and G. Cao, "Efficient privacy-preserving stream aggregation in mobile sensing with low aggregation error," in: E. De. Cristofaro, M. Wright,(Eds.), *Privacy Enhancing Technologies, Lecture LNCS*, vol. 7981, pp. 60–81, 2013.

[24] H. To, G. Ghinita, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," in *Proc. of VLDB Endowment 2014*, vol. 7, iss. 10, pp. 919–930, 2014.

[25] F. Olumofin, P. K. Tysowski, I. Goldberg, and U. Hengartner, "Achieving efficient query privacy for location based services", in *Privacy Enhancing Technologies, LNCS*, vol. 6205, pp. 93–110, 2010.

[26] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proc. of ACM MobiSys*, pp.31–42, 2003.

[27] X. H. Chen and J. Pang, "Measuring Query Privacy in Location-Based Services," in *Proc. of ACM CODASPY*, pp. 49–60, 2012.

[28] E. De. Cristofaro and R. Di. Pietro, "Preserving Query Privacy in Urban Sensing Systems", in *Distributed Computing and Networking, LNCS*, vol. 7129, pp 218–233, 2012.

[29] X. H. Chen, J. Pang, "Exploring dependency for query privacy protection in location-based services," in *Proc. of ACM CODASPY*, pp. 37–47, 2013.

[30] R. Owens and W. C. Wang, "Preserving data query privacy in mobile mashups through mobile cloud computing," in *Proc. of IEEE ICCCN*, 2013.

[31] "Aroma: a new data protection method with differential privacy and accurate query answering," in *Proc. of ACM CIKM*, pp. 1569–1578, 2014.

[32] S .Shang, T. Wang, P. Cuff, and S. Kulkarni, "The application of differential privacy for rank aggregation: Privacy and accuracy," in *Proc. of IEEE Fusion*, pp. 1–7, 2014.

[33] A. Wood and J. Stankovic, "Denial of service in sensor networks," in *IEEE Computer*, vol. 35, iss. 10, pp. 54–62, 2003.

[34] J. McCune, E. Shi, A. Perrig, and M. Reiter, "Detection of denial-ofmessage attacks on sensor network broadcasts," in *Proc. of IEEE S&P*, pp. 64–78, 2005.

[35] M. Naor and B. Pinkas, "Oblivious Transfer with Adaptive Queries," in *Proc. of IACR CRYPTO*, pp. 573–590, 1999.

[36] W. G. Tzeng, "Efficient 1-out-of-n oblivious transfer schemes with universally usable parameters" in *IEEE TOC*, vol. 53, iss. 2, pp. 232–240, 2004.

[37] G. Asharov, Y. Lindell, T. Schneider, and M. Zohner, "More Efficient Oblivious Transfer and Extensions for Faster Secure Computation," in *Proc. of ACM CCS*, pp. 535–548, 2013.