LocaWard: A Security and Privacy Aware Location-Based Rewarding System

Ming Li, *Student Member*, *IEEE*, Sergio Salinas, *Student Member*, *IEEE*, and Pan Li, *Member*, *IEEE*

Abstract—The proliferation of mobile devices has driven the mobile marketing to surge in the past few years. Emerging as a new type of mobile marketing, mobile location-based services (MLBSs) have attracted intense attention recently. Unfortunately, current MLBSs have a lot of limitations and raise many concerns, especially about system security and users' privacy. In this paper, we propose a new location-based rewarding system, called LocaWard, where mobile users can collect location-based tokens from token distributors, and then redeem their gathered tokens at token collectors for beneficial rewards. Tokens act as virtual currency. The token distributors and collectors can be any commercial entities or merchants that wish to attract customers through such a promotion system, such as stores, restaurants, and car rental companies. We develop a security and privacy aware location-based rewarding protocol for the LocaWard system, and prove the completeness and soundness of the protocol. Moreover, we show that the system is resilient to various attacks and mobile users' privacy can be well protected in the meantime. We finally implement the system and conduct extensive experiments to validate the system efficiency in terms of computation, communication, energy consumption, and storage costs.

Index Terms—Mobile location-based services, security, privacy

1 Introduction

With the rapid evolution of mobile devices, mobile location-based services (MLBSs) have emerged as a new type of mobile marketing. According to a 2010 report by Pew Research Center, on any given day, 1 percent of online Americans used MLBSs [1]. Juniper Research predicts that the revenues from MLBSs will surge to more than $12.7 billion by 2014 [2].

Currently, there are various kinds of MLBSs. One of them is location-based social networking, such as Facebook Places [3], where users share their locations with friends and find others who are nearby. Another type of MLBSs requires the users to provide current or historical location proof to fulfill some purposes [4], [5], [6]. For example, a hospital may allow doctors or nurses to access patients' documents only when they can prove that they are in a particular room of the hospital [4]. A person accused of committing a crime is very much interested in being able to prove to the police that he was somewhere else rather than at the crime scene while the crime was committed. Mobile commerce is another branch of MLBSs, for example, forwarding advertisements to customers when they are near a business spot [7]. These MLBSs do not consider rewarding services.

More recently, a new type of MLBSs called location-based check-in game, which is developed based on location-based social networking, lets users earn beneficial rewards if they visit certain places. In particular, some applications, including Foursquare [8], and Loopt Star [9], let users check in different locales (e.g., coffee shops, restaurants, shopping malls) to not only compete with friends in games, but also earn rewards, points, or discounts from retailers and organizations. The rewards and reward amounts can be different depending on time of day, how frequently the person has checked it in the past, and so on. However, these location-based check-in systems are limited in several aspects. *First of all*, customers can only receive and redeem rewards at the same brand stores or even the same store only. For instance, if a customer visits a Gap store twice, he/she can get a discount on the purchases at Gap stores (or the same Gap store) only, not at any other places like Starbucks. This greatly weakens the customers' motivations for visiting the locales. *Second*, from a service provider's perspective, security is not guaranteed in the existing systems. Since users can receive benefits for visiting some places, they have incentives to claim that they are at certain locations even though they are not. Most of those location-based check-in applications (e.g., Foursquare) use the GPS on a user's mobile device to verify the location claimed by the user. However, users may cheat on their locations by, for example, jailbreaking their mobile devices. This problem is in fact very common in most MLBSs and have not been satisfactorily solved by existing works [5], [6], [10], [11], [12], [13]. *Third,* from users' perspective, users' privacy including identity privacy and location privacy has been largely ignored in the current check-in systems. In particular, since the current systems use central servers to store all users' records, they can easily know which users have ever been to which places at what times

- *The authors are with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762-9571. E-mail: {ml845, sas573}@msstate.edu, li@ece.msstate.edu.*

Manuscript received 16 Sept. 2012; revised 16 Apr. 2013; accepted 11 May 2013; published online 24 May 2013.
Recommended for acceptance by X. Li, P. McDaniel, R. Poovendran, and G. Wang.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDSSI-2012-09-0947.
Digital Object Identifier no. 10.1109/TPDS.2013.152.

for what purposes. Previous works on users' identity privacy in wireless networks are not applicable to MLBS scenarios [14], [15]. Although there has been some research on location privacy regarding general location-based services, such as k-anonymity cloaking [16], [17], [18], [19], location obfuscation [20], [21], [22], [23], [24], pseudonym exchanges in mix zones [25], [26], [27], [28], they all have their limitations.[1]

In this paper, we propose a secure, privacy-preserving, and realistic mobile location-based rewarding system, called LocaWard, which strives to address the above concerns. The proposed system consists of a trusted third party (TTP), mobile users (MUs), token distributors (TDs), token collectors (TCs), and a central controller (CC). The TTP issues each MU with a real identity and a corresponding certificate. A legal MU is able to obtain a location-based token when it visits a commercial entity that participates in the system, i.e., a TD. The issued tokens at various TDs have the same format but possibly different indicated values. With all the collected tokens, an MU can redeem them for beneficial rewards not only at the same store or brand stores, but also at any other retailers or commercial entities, i.e., TCs, that have joined the system. The amount of received rewards depends on the value represented by the collected tokens. Besides, the CC stores token audition information sent by TDs and provides it to TCs when required.

Then, we design a security and privacy aware location-based rewarding protocol for the proposed LocaWard system. We assume that TDs, TCs, and the CC work in the semihonest mode, i.e., they faithfully and correctly execute the system protocol but are curious about MUs' privacy, including their personal information like real identities, token information, and location histories. Specifically, the protocol is composed of three parts: identity initiation, token distribution, and token redemption. In *identity initiation*, the TTP issues each MU with an identity and a corresponding certificate. Each MU keeps its identity private and generates a new pseudonym for each token request or redemption. The certificate is used for a user's identity authentication without revealing its real identity. In *token distribution*, a TD needs to verify if an MU requesting a token is a legal user in the system without knowing its real ID. After that, the TD issues the MU with an anonymous token which can be redeemed at any TC for rewards. Since the token contains some of the MU's private information, it is only kept by the MU but not any other network entities, including TCs and the CC. The TD then generates corresponding audition information for the token and sends it instead of the token itself to the CC for future token verification. In *token redemption*, a TC first verifies whether the current MU trying to redeem a token is a legal system user, without knowing its real ID. Then, the TC checks to see if the token to be redeemed is intact and has not been tampered since it was generated with the help of the CC, without knowing the content of the token. After that, the TC checks if the token does belong to the MU. If the MU passes all these verification phases, the TC verifies whether the value of the token claimed by the MU is true, and if so, distributes the corresponding rewards to him/her. Therefore, in our proposed system, no one else other than the TTP can know an MU's real identity. As the CC and TCs only have the knowledge of token audition information, they do not know the content of any token. Since a TD/TC is only aware of the location of the tokens it issued/accepted and there is no central server to store all the historical location information, no entity could figure out any specific MU's location history. Note that our design does not require any trustworthy server for generating/storing location proofs like in [3], [5], [8], [13], [29] or for protecting users' location privacy like in [16], [17], [18], [19], [25], [26], [27], [28]. Moreover, we have proved both the completeness and the soundness of the protocol, while most previous systems only focus on their completeness.

Furthermore, we analyze the security and privacy of the LocaWard system. We find that the system is resilient to various attacks such as multitoken request attack, duplicate token redemption attack, impersonation attack, token-tampering attack, and colluding attack. We also show that the MUs' privacy can be well protected. In addition, we build a testbed consisting of an Android smartphone and a laptop to implement our proposed system. We validate the efficiency of LocaWard in terms of computation, communication, energy consumption, and storage costs through extensive experiments.

2 System Models

2.1 System Architecture

In LocaWard, the system entities include a Trusted Third Party (TTP), Mobile Users (MUs), Token Distributors (TDs), Token Collectors (TCs), and a Central Controller (CC). Please refer to Fig. 1 in Appendix B, available in the online supplemental material, for the architecture of LocaWard. In what follows, we describe the functionalities and interactions of these system entities.

Trusted Third Party (TTP): A trusted third party which issues each MU with an identity and a certificate. The TTP is only responsible for issuing identities and not involved in any other activities in the system.

Mobile Users (MUs): The mobile devices which collect location-based tokens and redeem them for beneficial rewards. Each time that an MU visits a token distributor, it sends a request and receives a token through its WiFi interface. Whenever an MU meets a token collector, it can redeem its gathered tokens. After the token collector verifies that the tokens are redeemable, the MU will receive the corresponding rewards. The communications between MUs and token collectors can also be carried out via their WiFi interfaces.

Token Distributors (TDs): The commercial entities who issue redeemable tokens containing reward points to attract customers, such as stores, restaurants, and car rental companies. Each TD is equipped with a WiFi access point (AP) which can distribute location-based tokens. Besides, each TD also generates corresponding audition information and stores it in the CC for future token verification. TDs are connected to the CC through a backbone wired network, say the Internet.

[1] Due to limited space, we defer the extensive discussion of related works in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.152.

Token Collectors (TCs): The commercial entities who verify the MUs' token redemptions and reward the MUs with benefits, for example, monetary rewards, coupons, gift cards. TCs communicate with MUs via WiFi interfaces and are connected to the CC via the backbone network. Note that some TDs can serve as TCs at the same time.

Central Controller (CC): As commonly used in many mobile application systems [10], [30], we consider an online Center Controller run by an independent third party. It is responsible for storing audition information of a token and forwarding it to a TC when asked to.

2.2 Threat Model

We consider that some outsider adversaries in the network may intend to obtain MUs' private information by impersonating some legal MUs or eavesdropping on the wireless communications between the MUs and the TDs/TCs. It is beyond the scope of this paper to consider that the adversaries perform some active attacks like channel jamming, mobile worm attacks, denial-of-service (DoS) attacks, or sabotaging the protocols executed among TDs/TCs/CC, since there have been some defense schemes against them.

Besides, as there are benefits of having location-based tokens, MUs have the incentive to lie to TDs/TCs. First, they can generate excessive token requests to a TD and try to acquire multiple tokens during the same visit, or try to redeem the same token more than once at TCs. Second, an MU may eavesdrop on the communications between other MUs and TDs/TCs, steal their tokens and/or pseudonyms, and try to redeem the tokens. Third, an MU may change the content of a token to try to obtain more rewards. The same as before, we do not consider malicious MUs who try to disrupt the normal operation of the network.

Moreover, in the LocaWard system, TDs are trusted by TCs to provide valid location-based tokens following their preestablished contracts. TCs also trust TDs in the sense that TDs would not collude with MUs to undermine their common interest. In addition, TDs trust TCs to give beneficial rewards to the MUs who present valid tokens. We consider that TDs, TCs, and the CC work in the semihonest mode, i.e., they faithfully and correctly execute the system protocol, but are curious about MUs' privacy.

Furthermore, we consider there are preestablished security keys between TDs/TCs and the CC, and the communications among them are secure.

2.3 Design Goals

- **System security**: *completeness* and *soundness*. Completeness means that honest MUs can always successfully obtain tokens from TDs and redeem valid tokens at TCs. Soundness refers to that the probability that forged/tampered/stolen tokens can be redeemed is negligible.
- **Users' privacy**: Users' private information includes: first, MUs' personal information like real identities, second, token information including the value of a token, and third, location histories. Note that since TDs issue those tokens, they know some of MUs' token information. However, they cannot know MUs' real identities, or the information of the tokens issued by other TDs, or MUs' previous location histories. Besides, although TCs are responsible for verifying MUs' tokens to be redeemed, they cannot know MUs' real identities, or any of the detailed token information except the values of the tokens to be redeemed, or MUs' previous location histories. Any MU cannot know any other MUs' private information either.

3 Preliminaries

3.1 Bilinear Maps

Let $\mathbb{G}$ and $\mathbb{G}_T$ be a cyclic additive group and a cyclic multiplicative group, respectively, of the same prime order n. An admissible bilinear pairing $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a map with the following properties [31]:

1. *Bilinearity*: $e(P^a, Q^b) = e(P, Q)^{ab}$, for any $P, Q \in \mathbb{G}$, and $a, b \in \mathbb{Z}_n^*$, where $\mathbb{Z}_n^*$ denotes the multiplicative group of $\mathbb{Z}_n$, the integers modulo n. In particular, $\mathbb{Z}_n^* = \{z|1 \leq z \leq n-1\}$ since n is a prime.
2. *Nondegeneracy*: $\exists P, Q \in \mathbb{G}$ such that $e(P, Q) \neq 1$;
3. *Computability*: There is an efficient algorithm to compute $e(P, Q)\ \forall P, Q \in \mathbb{G}$.

Such an admissible bilinear pairing $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ can be implemented by the modified Weil/Tate pairings.

3.2 Shared Symmetric Key

We employ the secret-splitting principle [32] to develop shared symmetric keys between MUs and TDs/TCs instead of using asymmetric keys. The reason is that when MUs use public/private key pairs, adversaries may be able to infer MUs' identities by analyzing their public key patterns. In contrast, we use the following simple secret-splitting mechanism to generate a symmetric key between an MU and a TD or a TC: $K_{1,2} = N_1 \oplus N_2$, where N_1 and N_2 are sufficiently large random numbers generated by the MU and the TD/TC, respectively. In this paper, we use N_1's and N_2's that are 256-bit long. The computation complexity of this symmetric key generation scheme is low since XOR is a simple operation.

3.3 Complexity Assumptions

Strong RSA assumption. Let p and q be two distinct large prime numbers and $r = p \times q$ (r is semiprime). Given a randomly chosen $b \overset{R}{\longleftarrow} \mathbb{Z}_r^*$ and $a > 1$, it is computationally intractable to find $x \in \mathbb{Z}_r^*$ such that $b = x^a \bmod r$ without knowing the factorization of r.

Discrete logarithm problem (DLP) assumption. Let $\mathbb{G}$ be a cyclic group of order n, and g a generator of $\mathbb{G}$. Then, given $h \in \mathbb{G}$, it is computationally intractable to find $a \in \mathbb{Z}_n$ such that $h = g^a$.

4 A Security and Privacy Aware Location-Based Rewarding Protocol for LocaWard

In this section, we develop a security and privacy aware location-based rewarding protocol for the LocaWard system. The protocol mainly consists of three processes: *identity initiation*, *token distribution*, and *token redemption*. We also show that the protocol fulfills the aforementioned design goals.

4.1 Identity Initiation

Before an MU i enters the system, it is issued by a TTP with a real identity s_i and a certificate $cert_i$ of s_i. In particular, let n_T be the product of two primes p_T and q_T, where $p_T = 2kp'_T + 1$, $q_T = 2kq'_T + 1$, and k, p'_T and q'_T are distinct primes. Note that we set $n_T = p_T \times q_T$ to be 1,024-bit long [33]. The TTP randomly chooses an integer ϵ_i such that $1 < \epsilon_i < \phi(n_T)$ and $\gcd(\epsilon_i, \phi(n_T)) = 1$, where $\phi(n_T) = (p_T - 1)(q_T - 1)$ is Euler's totient function. Denoted by w a generator of sufficiently large subgroup $\mathbb{G}_{n_T}$ of $\mathbb{Z}^*_{n_T}$, which should be selected by the TTP to satisfy $w^m \equiv 1 \bmod n_T$ where m is the order of $w \bmod n_T$ and a factor of $\phi(n_T)$ [33]. Then, the TTP computes s_i as follows: $s_i = w^{\frac{1}{\epsilon_i}} \bmod n_T = w^{d_i} \bmod n_T$ where $\epsilon_i d_i \equiv 1 \bmod \phi(n_T)$[2]. Note that since ϵ_i is closely related to s_i, it serves as a test parameter of the MU's identity s_i. However, it is computationally intractable to calculate s_i given ϵ_i and w due to the strong RSA assumption.

In addition, the TTP generates a corresponding signature on ϵ_i using its private key K^{pri}_T, i.e., $K^{pri}_T(\epsilon_i)$, which is used as a certificate denoted by $cert_i$. Finally, the TTP issues the MU with its real ID s_i and the corresponding certificate $cert_i$. Here, n_T, w, and the TTP's public key K^{pub}_T are public to the whole system, while the factorization of n_T is concealed. Thus, no MU can know $\phi(n_T)$ or obtain another ϵ'_i that satisfies the above conditions even given an ϵ_i.

4.2 Token Distribution

When an MU, who is interested in collecting location-based tokens, visits the site of a TD, it initiates a token request conversation with this TD. In order to protect its identity privacy, the MU randomly generates a pseudonym (PID) based on its real identity (ID) to contact the TD, instead of directly using its real ID. Note that an MU updates its PID in each token request to avoid being linked to its real ID [34]. While on the TD's side, it first needs to check MU's identity before allocating a token. Thus, the token distribution process consists of two phases: *MU's identity authentication* and *token distribution*. Note that the MU's privacy should be protected during the whole process.

4.2.1 MU's Identity Authentication at a TD

The purpose of authenticating an MU's identity before a TD distributes a location-based token is twofold. First, there might be misbehaving users who use fake IDs to generate PIDs in order to obtain more location-based tokens for more beneficial rewards. Second, identity authentication can efficiently defend against some of MUs' misbehavior, for example, impersonation attack and colluding attack, which we would discuss in details later in the paper. In general, in the identity authentication phase, a TD checks if an MU has a valid ID without knowing the MU's real ID. The detailed phase is described as follows:

- Let p_i and q_i be two distinct large prime numbers and $n_i = p_i \times q_i$ be a 768-bit public semiprime number to ensure the hardness of discrete logarithm problem [35]. The MU chooses a random number r_i such that $1 < r_i < \phi(n_i)$ and $\gcd(r_i, \phi(n_i)) = 1$. Let ρ_i be a generator of a sufficiently large subgroup of $\mathbb{Z}^*_{n_i}$. Then, the MU computes its PID as $pid_i = \rho_i^{s_i r_i} \bmod n_i$. Here, ρ_i, s_i, and r_i are this MU's secrets kept from the other entities in the system, and no adversary can figure out the MU's real ID s_i from pid_i. Note that the MU uses a new PID for each token request. After that, the MU generates a token request using TD's public key, i.e., $req_D = K^{pub}_{TD}(cert_i \| pid_i \| N_i)$, where N_i is a nonce of sufficiently long bit-length generated by the MU for constructing a shared secret session key between itself and the TD. In this paper, we set N_i to be 256-bit long. The session key for each token request is different. In addition, the TD broadcasts its public key to MUs visiting its site when they get associated with the TD's AP.
- Once the TD receives the MU's token request, it decrypts the request message and obtains $cert_i$ (and thus ϵ_i), pid_i, and N_i using its own private key K^{pri}_{TD}. Then, the TD chooses another nonce N_d of 256 bits to construct a shared session key $K_{i,TD}$ between the MU and itself, i.e., $K_{i,TD} = N_i \oplus N_d$, and sends N_d (encrypted by its private key $K^{pri}_{TD}(N_d)$) back to the MU. Note that even if an adversary can intercept N_d, it does not know N_i and hence cannot recover the symmetric key $K_{i,TD}$. Therefore, the data exchanged between the MU and the TD are secure.
- Upon receiving $K^{pri}_{TD}(N_d)$ from the TD, the MU retrieves N_d by applying the TD's public key. After that, the MU chooses a random number $r'_i \in \{1, \ldots, n_T - 1\}$, and sends $x = (r'_i)^{\epsilon_i} \bmod n_T$ to the TD. Note that since both the MU and the TD can now construct their shared session key $K_{i,TD}$, they apply it to encrypt and decrypt the messages in the rest of their conversations. We omit such procedures to simplify the descriptions here.
- The TD chooses a random number $r_{TD} \in \{1, \ldots, n_T - 1\}$ and sends it as a question to the MU.
- With two parameters r'_i and r_{TD}, and its private knowledge of s_i, the MU sends to the TD $y = r'_i \cdot (s_i)^{r_{TD}} \bmod n_T$ as an answer.
- The TD checks the following verification equation: $x \equiv y^{\epsilon_i} \cdot w^{-r_{TD}} \bmod n_T$. The identity authentication succeeds if this equation holds, and fails otherwise.

The whole identity authentication phase is also briefly shown by Fig. 2 in Appendix B, available in the online supplemental material. Besides, please refer to Appendix D, available in the online supplemental material, for detailed proofs of *completeness*, *soundness*, and *privacy reservation* of the identity authentication phase.

4.2.2 Location-Based Token Distribution

If the MU can successfully pass the identity authentication phase, the TD would continue processing the MU's token request. Before we delve into the details of token distribution, we would like to give the definition of a time window first. In particular, the time window used by a TD is the duration within which each MU can only receive one location-based token from this TD. The length of the time window can be regulated by the system and indicates

[2] Please refer to Appendix C, which is available in the online supplemental material, for the detailed derivation process.

the minimal time interval after which a TD can issue another token to the same MU, for example, 30 minutes or 1 hour. Note that the length of the time window does not affect the design of the token distribution phase, which is described as follows:

- The TD first checks if the token request is an excessive one by comparing ϵ_i (obtained from $cert_i$) with the existing records within a specified time window. Since each MU only has one ϵ_j corresponding to its identity as a test parameter, the TD can check the token request records, i.e., the MUs' ϵ_j's, in the current time window and see if there is already an existing record of ϵ_i. If so, the TD determines the current token request is invalid, and does not issue a new token to the MU.
- If the current token request is not an excessive one, the TD generates a new location-based token as follows: $\psi = pid_i \| v \| cert_i \| r_d$, where v stands for the value of the token, i.e., the reward points, $cert_i$ is the TTP's signature over ϵ_i, r_d is a 512-bit length nonce chosen by the TD to protect the privacy of token.[3] Here, we let v be of 32 bits. Since pid_i and $cert_i$ are 768-bit and 1,024-bit long, respectively, the total length of ψ is $768 + 32 + 1,024 + 512 =$ 2,336 bits. The TD then sends the token to the MU which is encrypted by the shared symmetric key $K_{i,TD}$.
- Finally, the TD generates audition information for this token. Since one of our objectives in system design is to keep the token's content as the MU's private information, the TD generates audition information and sends it instead of the token itself to the CC for future token verification at TCs. Specifically, let $\mathbb{G}_1$ and $\mathbb{G}_2$ be a cyclic additive group and a cyclic multiplicative group, respectively, of the same prime order n_d, and $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a bilinear map as introduced before. Let g be a generator of $\mathbb{G}_1$, and $H(\cdot)$ be a shared secret map-to-point BLS hash function among all the TDs and TCs: $\{0,1\}^* \rightarrow \mathbb{G}_1$, which maps strings uniformly to $\mathbb{G}_1$. It is infeasible to find out the original message given a hashed message. Then, the TD chooses a random number $\alpha \xleftarrow{R} \mathbb{Z}^*_{n_d}$, where n_d is a 512-bit long prime number in our paper, a random element $u \xleftarrow{R} \mathbb{G}_1$, and computes $h = g^\alpha \in \mathbb{G}_1$. We denote the token ψ in the following more general form: $\psi = I_1 \| I_2 \| I_3 \| I_4$, where I_k ($1 \leq k \leq K$, $K = 4$) stands for the kth item in the token. Thus, the TD computes the audition value σ_k for each item I_k as follows: $\sigma_k = (H(I_k) \cdot u^{I_k})^\alpha \in \mathbb{G}_1$. Denote the whole set of audition parameters for a token by $\Phi = \{\sigma_1, \ldots, \sigma_K, \ H(I_1), \ldots, H(I_K)\}$. Subsequently, the TD uploads the audition information $\Psi = pid_i \| \Phi \| g \| u \| h \| \alpha \| n_d$ and stores it in the CC. There are mainly two reasons for generating audition parameters in such a way. First, since audition parameters contain full information of an issued token, it can be used for checking the integrity of a token. Second, the random number masked or hashed parameters prevent the CC from knowing the real content of the token, which might be compromised by adversaries. Note that the transmissions between the TD and the CC can be secured by a preestablished shared symmetric key between them, say $K_{TD,CC}$. Moreover, the CC attaches a 1-bit flag for each token: "1" means the token has been redeemed and "0" otherwise. The flag is set to "0" when the token is received at the CC.

The above operations are briefly summarized by Fig. 3 in Appendix B in the online supplemental material.

4.3 Token Redemption

Whenever an MU i encounters a TC, it can redeem its collected location-based tokens by initiating a token redemption conversation with this TC. As we discussed above, the MU does not want to reveal its real identity or token information to the TC. On the other hand, the TC needs to make sure that this MU is a legal user, and that the token provided by the MU is intact and valid, and indeed belongs to itself, i.e., not stolen from someone else. Thus, the token redemption process is divided into four phases: *MU's identity authentication*, *token audition*, *token property validation*, and *reward distribution*. In the whole process, the user's privacy should be protected.

4.3.1 MU's Identity Authentication at a TC

The TC first checks the MU's identity to make sure it is a legal user, for example, instead of an outsider adversary who tries to redeem a token stolen from some legal user.

The phase of identity authentication at the TC is quite similar to the one at the TD, except a few changes as follows: First, the token redemption request from the MU is generated as $K^{pub}_{TC}(cert_i \| pid_i \| N'_i)$, where pid_i is the pseudonym used for obtaining the token, N'_i is another nonce chosen by the MU for establishing a shared session key with the TC, and K^{pub}_{TC} is the public key of the TC. Note that the same as before, the MU can obtain K^{pub}_{TC} when it enters the TC's zone. Besides, the transmissions between the MU and the TC can be secured by the established shared symmetric key between them, denoted by $K_{i,TC}$. Second, both the MU and the TC choose new random numbers $r''_i, r_{TC} \in \{1, \ldots n_T - 1\}$, respectively, in the authentication phase.

Since the identity authentication phases at the TC and at the TD are the same, the completeness, soundness, and privacy reservation properties also hold at the TC.

4.3.2 Token Audition

The purposes of token audition are: first, to make sure that the token submitted by the MU is valid and not generated by other entities than a TD, and second, to guarantee that the token is intact and has not been tampered since it was generated. On the other hand, since the token carries some private information of the MU, its content should be hidden from the TC. The basic idea is to let the TC only use the corresponding audition information Ψ retrieved from the CC to verify the token without knowing its content. In what follows, we describe the detailed token audition phase.

[3] When the token also carriers the MU's other information besides pid_i, v and $cert_i$, the nonce r_d can be omitted. We explain it in detail in Appendix E in the online supplemental material.

- Once the TC determines that this MU holds a valid ID which belongs to itself, it then uses this MU's PID, i.e., pid_i contained in the token redemption request, to query for the corresponding token audition information Ψ from the CC. Note that their transmissions can be secured by a preestablished symmetric key between them, say $K_{TC,CC}$. The CC then searches its storage space for the records corresponding to pid_i. It first checks if the redemption flag for this token has been set to 1. If so, the CC sends the TC a message "This token has been redeemed," and aborts the entire process. Otherwise, the CC sends back the corresponding Ψ.
- The TC generates a challenge message for token audition. It chooses a set of random numbers $\{\nu_k | 1 \leq k \leq K\}$, where $\nu_k \overset{R}{\longleftarrow} \mathbb{Z}^*_{n_d}$ and sends $\nu_1 \| \ldots \| \nu_K$ to the MU.
- Upon receiving this challenge message, the MU responds with a corresponding proof. Specifically, MU computes a linear combination of all the items in ψ with different weights specified in $\{\nu_k\}, i.e., \mu = \sum_{1 \leq k \leq K} \nu_k I_k \bmod n_d$, and sends μ back to the TC.
- The TC also calculates the following quantity $\sigma = \prod_{1 \leq k \leq K} \sigma_k^{\nu_k} \in \mathbb{G}_1$. With the response from the MU, the TC checks the following verification equation: $e(\sigma, g) = e((\prod_{k=1}^{K} H^{\nu_k}(I_k)) \cdot u^{\mu}, h)$. The TC determines that the MU's token is not forged or tampered if this equation holds, and aborts the redemption process otherwise.

The token audition phase is briefly summarized by Fig. 4 in Appendix B in the online supplemental material. Please refer to Appendix E, available in the online supplemental material, for detailed proofs of *completeness*, *soundness*, and *privacy reservation* of the token audition phase.

4.3.3 Token Property Validation

Even though the MU is a legal user who has a valid ID issued by the TTP, it might steal some legal users' tokens for redemption and can pass the previous two phases. Thus, the TC still needs to verify if the token belongs to this MU, which we call the token property validation phase. In particular, recall that the TC can retrieve the MU's certificate $cert_i$ in the identity authentication phase. Now the TC computes $(H(cert_i) \cdot u^{cert_i})^{\alpha}$ and see if it is equal to σ_3 in the token audition parameter set Φ. If so, then the token does belong to the MU. Otherwise, the token is not the MU's. The completeness, soundness, and privacy reservation of this phase can be easily proved, which are omitted here.

4.3.4 Reward Distribution

After the MU trying to redeem a token passes the above three verification phases, the TC determines that this MU is qualified to redeem the token. Since in the previous verification phases, the MU's private information, including the value of this token v, is hidden from the TC, the MU needs to tell the TC v explicitly so that it can verify. Specifically, the TC calculates $(H(v) \cdot u^{v})^{\alpha}$ to see if this value is equal to σ_2 in the token audition parameter set Φ. If the two values are the same, the TC determines that the MU reports the correct v, and sends the corresponding rewards, for example, cash back or equivalent gift cards, to the MU. Otherwise, the TC aborts the redemption process. After the entire redemption process is finished, the TC informs the CC so that the CC labels the redemption flag on this token as 1 (or just deletes this token permanently).

5 Security and Privacy Analysis

5.1 Security Analysis

We first analyze the security of the system, considering that all the misbehaving MUs have valid identities issued by the TTP. Note that those MUs who do not have valid identities can be detected at the identity authentication phase.

Multi-token request attack. When visiting a TD, a well-behaved MU should obtain only one location-based token during each predefined time window, while a misbehaving MU may generate excessive token requests either by the same PID or by different PIDs, and try to get more than one tokens. Recall that an MU is required to send $cert_i$ during the identity authentication phase at the TD, and $cert_i$ is unique for every MU. By checking the existing request records in the time window for duplicate PIDs or $cert_i$'s (or ϵ_i's), the TD can easily detect any multitoken request attack.

Duplicate token redemption attack. In the duplicate token redemption attack, a misbehaving MU may try to redeem the same token multiple times. This kind of misbehavior can be easily defended against in our LocaWard system. In particular, as mentioned before the redemption flag of a token kept at the CC would be set to 1 (or the token can be deleted by the CC permanently), after the token is redeemed for the first time. Then, when the same token is redeemed again (i.e., indexed by the same pid_i), the TC would check with the CC and can easily find out that this is a duplicate redemption.

Impersonation attack. An impersonation attack is that a misbehaving MU manages to steal another MU's PID, or $cert_i$, or both to obtain tokens, or to steal another MU's tokens, in order to obtain more rewards from a TC. Our scheme is also efficient in defending against such misbehavior. In particular, in the first case, i.e., when a misbehaving MU uses another MU's PID and/or $cert_i$, to request for tokens, it cannot pass the identity authentication phase at the TD, since it does not know that MU's real identity s_i. We have proven it in the soundness of the identity authentication phase. In the second case, i.e., when an MU tries to redeem a stolen token, it cannot pass the identity authentication phase at the TC for the same reason. Besides, even if an MU could forge a redemption token request with its own $cert_i$ and the pid_i in a stolen token, it would fail the token property validation phase.

Token-tampering attack. In a token-tampering attack, a misbehaving MU tries to forge a fake token, or to change certain content, for example, value, of a token to get beneficial rewards. In the first case, since a forged token is not obtained from a TD, there will not be any related records at the CC. Thus, when the misbehaving MU tries to redeem the forged token at a TC, the TC can find out that there is no corresponding audition information for this token at the CC during the token audition phase and will abort the redemption process. In the second case, if MU tampers any content of the token, this token cannot pass the token audition, as shown in the soundness of the token audition phase.

Colluding attack. Colluding attack here refers to the collusion misbehavior among MUs in the system trying to obtain illegal profit. In particular, in colluding attacks, some remote misbehaving MUs, who are not in the area of a TD, may try to obtain location-based tokens through some colluding MUs who are currently visiting the TD, for example, use them as "agents." Note that it is reasonable to assume that MUs do not share their identities with each other as many previous works do [13], [36], due to the following reasons. First, an MU can impersonate another MU with that MU's real identity and hence obtain tokens and redeem tokens for benefits. Second, in the LocaWard system, tokens act as virtual currency. Then, identity is like the password to an MU's bank account which MUs are unlikely to reveal to others. Third, similar to that in [13], [37], [38], we can embed MUs' identities in their mobile devices, for example, by using trusted computing components (TCCs) like trusted platform modules (TPMs) to store and bind their identities to the developed protocol. Thus, MUs cannot change their identities or they do not even know their own identities. MUs have to give away their mobile devices to others to launch colluding attacks. There are strong incentives for MUs not to give away their mobile devices, such as being reachable continuously and protecting personal information stored on the device. Therefore, our system can detect colluding attacks by performing identity authentication and checking if an MU does have the correct knowledge of the private identity s_i corresponding to the certificate $cert_i$ contained in the token request. Besides, since we assume TDs, TCs, and the CC work in the semihonest mode, we do not consider they collude with each other, MUs, or outside adversaries to attack the system.

5.2 Privacy Analysis

We then discuss the privacy of MUs protected against other system entities, including, TDs, TCs, and the CC.

A TD issues location-based tokens to MUs. So it has full knowledge of the tokens it has issued and knows the locations of those MUs when they visit it. However, since MUs only use their PIDs to request tokens, a TD cannot know the MUs' real IDs, even though it can tell if two requests are from the same MU by comparing the $cert_i$'s (or ϵ_i's) it has received. A TD cannot know the content of the tokens that MUs obtained from other TDs, or their location histories either.

A TC collects location-based tokens and rewards the MUs with benefits. As described before, a TC only knows the value of the tokens it has accepted, but nothing else. It does not know the locations where these tokens were obtained, or the real IDs of the MUs redeeming the tokens. Besides, the same as TDs, a TC only knows the current location of some MUs when they visit it to redeem tokens and cannot know their location histories.

In the case that TDs/TCs collude with other TDs/TCs, they basically exchange data regarding pid_i's with the same $cert_i$ and try to identify the MU. Since TDs/TCs do not know the real identities of any MUs in our system, even if some TDs/TCs collude with other TDs/TCs, they can only know the places, i.e., some TDs/TCs, which a certain anonymous MU has visited. Notice that in traditional location-based services, a user's queries may contain sensitive personal information, for example, habits, diseases, jobs, and more importantly, locations which could probably be his/her working place or home address. Based on such history information, a location service server may be able to identify users with the help of some side information like their addresses [39], [40]. However, in our system, MUs do not submit any personal data to TDs/TCs. The locations recorded by the TDs/TCs are their own locations, which are all common public places. Therefore, the TDs/TCs will not be able to identify any MU even if they collude with each other.

The CC stores the audition information of the tokens issued by TDs. However, such information only contains MUs' PIDs instead of their real IDs, and the tokens' audition parameters from which the CC cannot infer the content of the tokens.

In summary, LocaWard is resilient to various kinds of attacks and can protect MUs' privacy well.

6 Implementation

In this section, we evaluate the computation, communication, energy, and storage costs of the proposed LocaWard system on our testbed, which consists of a laptop and an Android smartphone as shown in Fig. 5 in Appendix F, available in the online supplemental material. In particular, the laptop has a 2.5 GHz CPU and 4 GB RAM, while the smartphone is a Samsung Nexus S with 1 GHz ARM Cortex A8 processor and 512 MB RAM. We implement a TD, a TC, and the CC on the laptop platform, and a MU on the smartphone platform, respectively. The two platforms communicate with each other via the WiFi access point in our engineering building using IEEE 802.11b, and their conversations are carried out via TCP connections. Thus, the communication cost in practice can be even lower than the results presented later on, considering that TDs/TCs may be connected to their access points via cables and the WLAN in our building already hosts a lot of users. Besides, we use the Java Pairing Based Cryptography Library (jPBC) [41], a Java port of the PBC library [42], to conduct pairing-based computations, and use the "Type A" elliptic curve generator in the library with the default parameters (160-bit long group order r and 512-bit long base field q) to offer a security level of 80 bits. To secure the communications, we use RSA-1024 for asymmetric encryption/decryption, and AES-CBC-256 for symmetric encryption/decryption involved in our system, respectively. Note that since identity initiation can be done offline, we focus on the efficiency of the other two processes in the system, i.e., token distribution and token redemption. All the experimental results represent the average of 50 trials.

6.1 Efficiency of Token Distribution

Single token request: We first study the scenario where there is only one token request in the system. The computation time is analyzed as follows. As shown in Fig. 2 in Appendix B, available in the online supplemental material, *in the identity authentication phase*, an MU's computation includes: generating pid_i, encrypting the token request req_D, and decrypting $K_{TD}^{pri}(N_d)$ both with the TD's public key, constructing the session key $K_{i,TD}$, calculating $x = (r_i')^{\epsilon_i} \bmod n_T$, and encrypting x, decrypting r_{TD}, and computing and encrypting $y = r_i' \cdot (s_i)^{r_{TD}} \bmod n_T$. Thus, its computation complexity contains: $3 \times Exp$, $2 \times Mul$, $5 \times Enc/Dec$, and $1 \times XOR$.[4] The average computation time for completing all the above operations at an MU is 2.73 ms. Fig. 6 in Appendix F, available in the online supplemental material, also shows the step-by-step identity authentication done in one trial at the MU. Besides, the TD's computation includes: decrypting the token request req_D, verifying ϵ_i by applying K_T^{pub} to $cert_i$, constructing the session key $K_{i,TD}$ and using its private key to encrypt N_d, decrypting x, encrypting r_{TD}, decrypting y, and finally checking the verification equation. Therefore, the TD's computation complexity is: $2 \times Exp$, $1 \times Mul$, $6 \times Enc/Dec$, and $1 \times XOR$, all of which takes 2.52 ms. The total computation time in the identity authentication phase is thus on average $2.73 + 2.52 = 5.25$ ms.

In the token distribution phase shown in Fig. 2 in Appendix B, available in the online supplemental material, the TD needs to encrypt the generated token, and send it to the MU. It also calculates the audition parameters, i.e., $H(I_k)$ and $\sigma_k = (H(I_k) \cdot u^{I_k})^\alpha$ for each element $I_k (1 \leq k \leq K, K = 4)$ in the token, and $h = g^\alpha$. Since the audition parameters are only needed when the MU redeems this token, the computations involved in calculating the audition parameters do not need to be conducted in real time. Therefore, the TD's realtime computation complexity is: $1 \times Enc/Dec$, resulting in the average computation time of 0.69 ms, while its non-realtime computation complexity contains $(2K + 1) \times Exp$, $K \times Hash$, and $K \times Mul$ and takes 104.29 ms. Besides, in this phase, the MU decrypts the received data to obtain the token from the TD, which takes 0.45 ms. Thus, the total real-time computation cost in the token distribution phase is on average $0.45 + 0.69 = 1.14$ ms.

We can see that the total computation time needed to process one token request is $5.25 + 1.14 = 6.39$ ms. We also notice that the total computation time at MU, which is $2.73 + 0.45 = 3.18$ ms, takes $3.18/6.39 = 49.8$ percent of the entire computation time. The detailed computation cost in the token distribution process can also be found in Table 2 in Appendix F.1, which is available in the online supplemental material.

We then evaluate the communication cost in terms of both the transmitted payload data size and communication time. Since the TD communicates with the CC via a wired connection, the transmission time is negligible compared with the wireless transmission time. Therefore, we only show the communication cost between the TD and the MU. Besides, the measured communication time also includes the java program running time involved in data transmission. Specifically, *in the identity authentication phase*, the MU transmits 512 bytes payload data ($1,024 \times 2 = 2,048$ bits for req_D,[5] 1,024 bits for x, and 1,024 bits for y[6]), which incurs 168.39 ms communication time, while the TD transmits 256 bytes payload data (1,024 bits for $K_{TD}^{pri}(N_d)$, 1,024 bits for $K_{i,TD}(r_{TD})$), which results in 111.48 ms communication time. The total communication time in the identity authentication phase is $168.39 + 111.48 = 279.87$ ms. *In the token distribution phase*, only the TD sends a token of $256 \times 10 = 2,560$ bits to the MU, which takes 59.32 ms. Therefore, the overall communication time in the token distribution process is $279.87 + 59.32 = 339.19$ ms. The detailed communication cost in the token distribution process can also be found in Table 3 in Appendix F.1, available in the online supplemental material.

Moreover, based on the above results, we can see that the LocaWard system has very low latency, i.e., $6.39 + 339.19 = 345.58$ ms, for an MU to obtain a token. In addition, we find that the total computation time is much less than the communication time.

Furthermore, we evaluate the energy consumption of the token distribution process as follows: Since only the MU is energy-constrained in the system, we focus on the energy consumption of the MU in our experiment. In particular, the MU's energy consumption *in the identity authentication phase* and that *in the token distribution phase* are 0.84 and 0.15 mAh, respectively. The total energy consumption is 0.99 mAh, while the battery capacity of the MU, i.e., Samsung Nexus S, is 1,500 mAh. The detailed energy consumption cost in the token distribution process can also be found in Table 4 in Appendix F.1, available in the online supplemental material. Note that the battery of a mobile device is usually over 1,000 mAH. For example, iPhone 5 has a battery of 1,440 mAh, HTC One has a battery of 2,300 mAh, and Samsung Galaxy S4 has a battery of 2,600 mAh [43]. We can easily see that the proposed scheme is very efficient in energy consumption.

Multiple token requests: Next, we consider multiple MUs with multiple token requests in the system. Due to limited space, please refer to Appendix F.2 in the online supplemental material for detailed experiment results.

6.2 Efficiency of Token Redemption

Single token request: We first study the scenario where there is only one token request in the system. The computation time is analyzed as follows: As we discussed before, *the identity authentication phase* at the TC is quite similar to that at the TD. The computation time at the MU and at the TC are 2.8 and 2.95 ms, respectively, which are close to those at the TD. The total average computation time in the identity authentication phase is 5.75 ms. Here, we focus on the efficiency of the other three phases. As shown in Fig. 4 in Appendix B, available in the online supplemental material, *in the token audition phase*, an MU only needs to decrypt $\nu_1 \| \ldots \| \nu_K$ to obtain ν_k's, generate a linear combination of the elements in its token, i.e., $\mu = \sum_{k=1}^{K} \nu_k I_k \bmod n_d$, and encrypt μ. The corresponding computation complexity is $K \times Mul$, $2 \times Enc/Dec$, and $(K-1) \times Add$, where $K = 4$. Since there are not many time-consuming operations at the MU, it can conduct the computations very efficiently, particularly in 0.67 ms. On the other hand, the TC needs to encrypt pid, decrypt the audition information, encrypt $\nu_1 \| \ldots \| \nu_K$, decrypt μ, and calculate $\sigma = \prod_{k=1}^{K} \sigma_k^{\nu_k} \in \mathbb{G}_1$ and check the verification equation. The corresponding computation complexity is $2 \times Pairing$, $(2K+1) \times Exp$, $(2K+1) \times Mul$ and $4 \times Enc/Dec$,

[4] The notations for some operations we perform in the proposed scheme can be found in Table 1 in Appendix F, available in the online supplemental material.

[5] With RSA-1,024, the plaintext, consisting of 1,024 bits for $cert_i$, 768 bits for pid_i, and 256 bits for N_i, results in two 1,024-bit ciphertext segments.

[6] AES-CBC-256 segments both x and y into 4 blocks, each with 256 bits, before encryption. As the length of output is equal to that of the input, the length of ciphertext of x and of y are still 1,024 bits.

which takes 218.75 ms since paring is the most time-consuming one among all those involved operations. Thus, the total average computation time in the token audition phase is $0.67 + 218.75 = 219.42$ ms. *In the token property validation and reward distribution phases*, the TC only needs to compute $(H(cert_i) \cdot u^{cert_i})^\alpha$ and $(H(v) \cdot u^v)^\alpha$ to see if they are equal to σ_3 and σ_2 in the token audition parameter set Φ, respectively. Therefore, the TC's computation complexity is $2 \times Exp$ and $1 \times Mul$, in each phase, resulting in the computation time of 22.5 and 20.62 ms, respectively. Besides, the MU needs to send v to the TC *in the reward distribution phase*, which incurs computation cost of 0.47 ms for $1 \times Enc$. Therefore, the total average computation time in the token property validation phase and that in the reward distribution phase are 22.5 ms and $20.62 + 0.47 =$ 21.09 ms, respectively. The overall computation time in the token redemption process is $5.75 + 219.42 + 22.5 + `21.09 = 268.76$ ms, out of which $2.8 + 0.67 + 0.47 = 3.94$ ms, i.e., only about 1.5 percent, is attributed to the MU. The detailed computation cost in the token redemption process can also be found in Table 5 in Appendix F.3, available in the online supplemental material.

We also illustrate the communication cost in the token redemption process. Similar to that in the token distribution process, we do not consider the communication between the TC and the CC in our experiment. Specifically, *in the token property validation phase*, since the MU and the TC do not exchange any data, the corresponding communication cost is 0. In the rest three phases, the MU transmits 608 bytes payload data to the TC, including 512 bytes for *identity authentication*, 64 bytes for token audition, and 32 bytes for *reward distribution*, which results in the communication time of $166.24 + 56.13 + 55.97 = 278.34$ ms. The TC transmits 512 bytes payload data to the MU, including 256 bytes for *identity authentication* and 256 bytes for *token audition*, which results in the communication time of $112.75 + 58.45 = 171.2$ ms. Note that although the sizes of transmitted payload data from the TC to the MU in the identity authentication phase and in the token audition phase are the same, their corresponding communication times are not. The reason is that the TC transmits to the MU twice (for encrypted N_d and r_{TD}, respectively) in the identity authentication phase, while it only transmits once (for encrypted $\nu_1 \| \ldots \| \nu_K$) in the token audition phase. We can see that the total communication times in the identity authentication phase, in the token audition phase, in the token property validation phase, and in the reward distribution phase are $166.24 + 112.75 = 278.99$ ms, $56.13 + 58.45 = 114.58$ ms, 0 ms, and 55.97 ms, respectively. The overall communication time in the token redemption process is thus 449.54 ms. The detailed communication cost in the token redemption process can also be found in Table 6 in Appendix F.3, available in the online supplemental material.

We further evaluate the energy consumption of the MU in the token redemption process. The MU's energy consumption *in the identity authentication phase* and that *in the token audition phase* are 0.89 and 0.34 mAh, respectively. Its energy consumption *in the token property validation phase* is 0 since the MU is not involved in any computation or communication. The MU consumes 0.25 mAh *in the reward distribution phase*. Thus, the total energy consumption is 1.48 mAh, very low as well. The detailed energy consumption cost in the token redemption process can be found in Table 7 in Appendix F.3, available in the online supplemental material.

Multiple token requests: Next, we consider another scenario where multiple MUs redeem tokens. Due to limited space, please refer to Appendix F.4 in the online supplemental material for detailed experiment results.

7 Conclusion

In this paper, we have proposed a secure, privacy-preserving, and realistic location-based rewarding system, LocaWard. We have designed a security and privacy aware protocol for the LocaWard system and proven its completeness and soundness. We find that the system is resilient to many types of attacks and mobile users' privacy can be well protected as well. We have also evaluated the system efficiency by extensive real experiments and show that the computation, communication, energy, and storage costs are low. Moreover, although the proposed security and privacy aware location-based rewarding protocol is for our LocaWard system, the techniques herein can be generalized to address security and privacy problems in general location-based services and other areas like cloud computing.

Acknowledgments

This work was supported by the US National Science Foundation under grants CNS-1149786, ECCS-1128768, and CNS-1147851

References

[1] http://pewinternet.org/~/media/Files/Reports/2010/PIP-Location%20based%20services.pdf, 2010.

[2] Juniper Research, Mobile Location Based Services Applications, Forecasts and Opportunities 2010-2014, https://www.juniperre-search.com/reports/mobile_location_based_services, 2010.

[3] http://www.facebook.com/about/location.

[4] W. Luo and U. Hengartner, "Proving Your Location Without Giving up Your Privacy," *Proc. 11th Workshop Mobile Computing Systems Applications,* Feb. 2010.

[5] S. Saroiu and A. Wolman, "Enabling New Mobile Applications with Location Proofs," *Proc. 10th Workshop Mobile Computing Systems Applications,* Feb. 2009.

[6] V. Lenders, E. Koukoumidis, P. Zhang, and M. Martonosi, "Location-Based Trust for Mobile User-Generated Content: Applications, Challenges and Implementations," *Proc. Ninth Workshop Mobile Computing Systems Applications (HotMobile '08),* Feb. 2008.

[7] S. Loreto, T. Mecklin, M. Opsenica, and H.-M. Rissanen, "Service Broker Architecture: Location Business Case and Mashups," *IEEE Comm. Magazine,* vol. 47, no. 4, pp. 97-103, Apr. 2009.

[8] https://foursquare.com/.

[9] http://www.loopt.com/about/tag/loopt-star/.

[10] Z. Zhu and G. Cao, "Towards Privacy Preserving and Collusion Resistance in Location Proof Updating System," *IEEE Trans. Mobile Computing,* vol. 12, no. 1, pp. 51-64, Nov. 2011.

[11] B. Waters and E. Felton, "Secure, Private Proofs of Location," Technical Report TR-667-03, Dept. of Computer Science, Princeton Univ., Jan. 2003.

[12] N. Sastry, U. Shankar, and D. Wagner, "Secure Verification of Location Claims," *Proc. Second ACM Workshop Wireless Security (WiSe '03),* Sept. 2003.

[13] W. Luo and U. Hengartner, "Veriplace: A Privacy-Aware Location Proof Architecture," *Proc. 18th SIGSPATIAL Int'l Conf. Advances Geographic Information Systems (GIS '10),* Nov. 2010.

[14] K. Ren and W. Lou, "A Sophisticated Privacy-Enhanced Yet Accountable Security Framework for Metropolitan Wireless Mesh Networks," *Proc. 28th Int'l Conf. Distributed Computing Systems (ICDCS '08)*, June 2008.

[15] C. Ardagna, S. Jajodia, P. Samarati, and A. Stavrou, "Providing Mobile Users' Anonymity in Hybrid Networks," *Proc. 15th European Symp. Research Computer (ESORICS)*, Sept. 2010.

[16] M. Gruteser and D. Grunwald, "Anonymous Usage of Location-Based Services through Spatial and Temporal Cloaking," *Proc. First Int'l Conf. Mobile Systems, Applications Services (Mobisys '03)*, May 2003.

[17] B. Gedik and L. Liu, "Protecting Location Privacy with Personalized K-Anonymity: Architecture and Algorithms," *IEEE Trans. Mobile Computing*, vol. 7, no. 1, pp. 1-18, Jan. 2008.

[18] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing Location-Based Identity Inference in Anonymous Spatial Queries," *IEEE Trans. Knowledge Data Eng.*, vol. 19, no. 12, pp. 1719-1733, Dec. 2007.

[19] B. Gedik and L. Liu, "Location Privacy in Mobile Systems: A Personalized Anonymization Model," *Proc. IEEE 25th Int'l Conf. Distributed Computing Systems (ICDCS)*, June 2005.

[20] H. Kido, Y. Yanagisawa, and T. Satoh, "An Anonymous Communication Technique Using Dummies for Location-Based Services," *Proc. IEEE 25th Int'l Conf. Distributed Computing Systems (ICDCS)*, July 2006.

[21] H. Lu, C.S. Jensen, and M.L. Yiu, "Pad: Privacy-Area Aware, Dummy-Based Location Privacy in Mobile Services," *Proc. ACM Seventh ACM Int'l Workshop Data Eng. Wireless Mobile Access (MobiDE)*, June 2008.

[22] M. Duckham and L. Kulik, "A Formal Model of Obfuscation and Negotiation for Location Privacy," *Proc. Int'l Conf. Pervasive Computing*, May 2005.

[23] C.A. Ardagna, M. Cremonini, S.D.C. di Vimercati, and P. Samarati, "An Obfuscation-Based Approach for Protecting Location Privacy," *IEEE Trans. Dependable Secure Computing*, vol. 8, no. 1, pp. 13-27, Jan. 2011.

[24] A. Pingley, W. Yu, N. Zhang, X. Fu, and W. Zhao, "Cap: A Context-Aware Privacy Protection System for Location-Based Services," *Proc. IEEE 29th Int'l Conf. Distributed Computing Systems (ICDCS '09)*, June 2009.

[25] A. Beresford and F. Stajano, "Location Privacy in Pervasive Computing," *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 46-55, Jan.-Mar. 2003.

[26] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Preserving Privacy in Gps Traces via Uncertainty-Aware Path Cloaking," *Proc. 14th ACM Conf. Computer Comm. Security (CCS '07)*, Jan. 2007.

[27] X. Liu, H. Zhao, M. Pan, H. Yue, X. Li, and Y. Fang, "Traffic-Aware Multiple Mix Zone Placement for Protecting Location Privacy," *Proc. IEEE INFOCOM*, Mar. 2012.

[28] J. Meyerowitz and R.R. Choudhury, "Hiding Stars with Fireworks: Location Privacy through Camouflage," *Proc. ACM MobiCom*, Sept. 2009.

[29] http://www.yelp.com/, 2012.

[30] R.A. Popa, A.J. Blumberg, H. Balakrishnan, and F.H. Li, "Privacy and Accountability for Location-Based Aggregate Statistics," *Proc. 18th ACM Conf. Computer Comm. Security*, Oct. 2011.

[31] D. Boneh and M. Franklin, "Identity Based Encryption from the Weil Pairing," *SIAM J. Computing*, vol. 32, no. 3, pp. 586-615, 2003.

[32] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C.* Wiley, 1996.

[33] D.H. Nyang and J.S. Song, "Knowledge-Proof Based Versatile Smart Card Verification Protocol," *ACM SIGCOMM Computer Comm. Rev.*, vol. 30, no. 3, pp. 39-44, July 2000.

[34] C. Bettini, X.S. Wang, and S. Jajodia, "Protecting Privacy against Location-Based Personal Identification," *Proc. Second VDLB Int'l Conf. Secure Data Management (SDM '05)*, Aug. 2005.

[35] K.S. McCurley, "The Discrete Logarithm Problem," *Proc. Symp. Applied Math.*, vol. 42, pp. 49-74, 1990.

[36] W. Luo and U. Hengartner, "Proving your Location without Giving up Your Privacy," *Proc. ACM HotMobile*, Feb. 2010.

[37] M.S. Kirkpatrick and E. Bertino, "Enforcing Spatial Constraints for Mobile RBAC Systems," *Proc. 15th ACM Symp. Access Control Models Technologies (SACMAT '10)*, June 2010.

[38] M.S. Kirkpatrick, M.L. Damiani, and E. Bertino, "Prox-RBAC: A Proximity-Based Spatially Aware RBAC," *Proc. 19th ACM SIGSPATIAL Int'l Conf. Advances Geographic Information Systems (GIS '11)*, Nov. 2011.

[39] T. Xu and Y. Cai, "Exploring Historical Location Data for Anonymity Preservation in Location-Based Services," *Proc. IEEE INFOCOM*, Apr. 2008.

[40] M. Terrovitis and N. Mamoulis, "Privacy Preservation in the Publication of Trajectories," *Proc. Ninth Int'l Conf. Mobile Data Management*, Apr. 2008.

[41] "JPBC: Java Pairing Based Cryptography," http://gas.dia.unisa.it/projects/jpbc.

[42] "Pbc: Pairing Based Cryptography," http://crypto.stanford.edu/pbc/.

[43] http://iphonehelp.in/2013/03/29/iphone-5s-1450-mah-batteryis-gives-as-much-battery-life-as-2300-mah-battery-of-android-phones/.

Ming Li received the BE degree in electrical engineering from Sun Yat-sen University, China, in 2007, and the ME degree in electrical engineering from Beijing University of Posts and Communications, China, in 2010, respectively. She is currently working toward the PhD degree at the Department of Electrical and Computer Engineering, Mississippi State University. Her research interests include cross-layer optimization, and security and privacy in cognitive radio networks, smart grids, and cloud computing. She is a student member of the IEEE.

Sergio Salinas received the BS degree in telecommunications engineering from Jackson State University, Jackson, in 2010. He is currently working toward the PhD degree at the Department of Electrical and Computer Engineering, Mississippi State University. His research interests include cyber-physical systems, cloud computing, and online social networks. He is a student member of the IEEE.

Pan Li received the BE degree in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 2005, and the PhD degree in electrical and computer engineering from the University of Florida, Gainesville, in 2009, respectively. He is currently an Assistant Professor in the Department of Electrical and Computer Engineering, Mississippi State University. His research interests include capacity and connectivity investigation, cross-layer optimization and design, and security and privacy in wireless networks, complex networks, cyber-physical systems, mobile computing, and cloud computing. He has been serving as an editor for*IEEE Journal on Selected Areas in Communications—Cognitive Radio Series* and *IEEE Communications Surveys and Tutorials*, a *Feature Editor for IEEE Wireless Communications*, and a guest editor for *IEEE Wireless Communications SI on User Cooperation in Wireless Networks*. He received the US National Science foundation (NSF) CAREER Award in 2012 and is a member of the IEEE and the ACM.

Supplemental Material

LocaWard: A Security and Privacy Aware Location-Based Rewarding System

Ming Li, *Student Member, IEEE,* Sergio Salinas, *Student Member, IEEE,* and Pan Li, *Member, IEEE*

Abstract—This document provides the supplemental material for our main paper titled "LocaWard: A Security and Privacy Aware Location-Based Rewarding System". Appendix A provides a detailed literature review of existing works on mobile location-based services (MLBSs). Appendix B presents by figures the main operations involved in the identity authentication, token distribution, and token audition phases of the developed security and privacy aware location-based rewarding protocol. Appendix C and Appendix D give the detailed proof of security and privacy of the identity authentication phase and that of security and privacy of the token audition phase, respectively. Appendix E shows more experiment results, for both the single token request and the multiple token request scenarios. Note that some descriptions and references are similar to those in the main paper to make this supplemental material self-contained.

Index Terms—Mobile location-based services; security; privacy.

Appendix A
Detailed Related Work

Although there have been several kinds of mobile location-based systems (MLBSs), including location-based social networking [1], [2], historical location proof services for some purposes [3]–[5], mobile commerce [6], and location-based check-in game [1], [7], they cannot fully guarantee system security and user privacy.

First, users can lie about their locations in order to receive more benefits. This problem is very common in most MLBSs but has not been satisfactorily solved by existing works. Specifically, although cellular base stations may provide unforgeable real-time location information, the accuracy is not good enough and such location history information may not be available for use. For example, 2G/3G systems have localization accuracy of around 100 meters [8]. Sun et al. [9] utilize signal patterns to better position users. They consider the multi-path signal patterns as the "fingerprints" of mobile devices, and estimate their locations by comparing the received signals at a base station with those stored in the database. Anisetti et al. [10] explore geographic information and can achieve location accuracy of 65 meters with 95% correct rate. One possible solution to provide high-accuracy location information is to have Bluetooth enabled mobile users in range generate location proofs for each other [11]. Unfortunately, this scheme has a strict constraint on the user density, i.e., when there are not enough neighbors within a distance of several meters, a user will not be able to collect location proofs. Lenders et al. [5] propose a solution that lets users generate unforgeable geo-tags such as photos or videos as their location proofs. However, it may introduce heavy communication overhead. [12], [13] rely on the round trip time of electromagnetic signals or light signals to measure the distance between two parties, which is based on high precision measurements and requires dedicated hardware. Recently, several systems [4], [14] propose to let users obtain their location proofs through WiFi infrastructures. Such systems require a secure trusted central server to manage and store the location proofs, which may not be practical in real applications. In the proposed LocaWard system, each MU needs to pass the identity authentication phase in order to get a token from a TD. We have proved the completeness and soundness of our identity authentication scheme, which ensures that an MU has to be within a TD's communication range in order to obtain a token. In addition, most of the previous schemes cannot defend against the collusion among users, while our system can.

Second, users' privacy, including their personal data (e.g., identities and activities) and location information, can be easily compromised. Specifically, since the current systems use central servers to store all users' records, they can easily know which users have ever been to which places at what times for what purposes. This puts users' privacy at risk. Unfortunately, users' privacy has been largely neglected in current system design. A few works propose schemes to achieve communication anonymity and data privacy in wireless networks [15]–[21], but are not applicable to MLBS scenarios. Although there have been some works discussing users' location privacy, they all have their limitations. k-anonymity cloaking schemes [22]–[27] propose to hide a user's real location by incorporating its neighbors' location infor-

- *This work was supported by the U.S. National Science Foundation under grants CNS-1149786 (CAREER Award), ECCS-1128768, and CNS-1147851.*
- *M. Li, S. Salinas, and Pan Li are with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762. E-mail: {ml845@, sas573@, li@ece.}msstate.edu.*

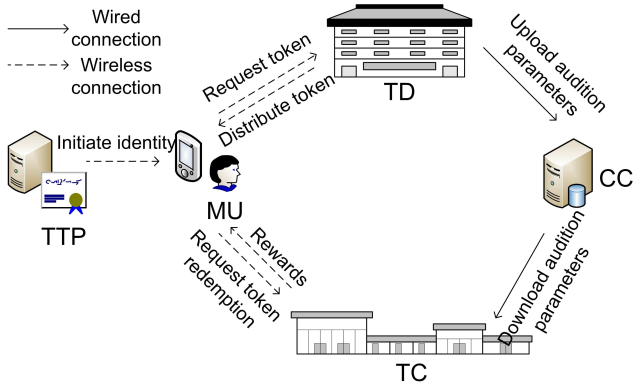


Fig. 1. The architecture of LocaWard system.

mation. However, they require a secure trusted central server, need the cooperation of at least k neighboring users, and may incur significant communication overhead. Another approach is location obfuscation [28]–[32]. By adding noise to a user's real location, adversary cannot infer the user's genuine location from the user's reported location information. Obviously, this method is at the cost of service accuracy. Besides, although using pseudonyms can protect users' privacy in one-time MLBSs, the server can continuously record users' pseudonym-location information. By piecing together all recorded historical location information and particular side-information (e.g., working/living addresses) [33], [34], the adversary might be able to infer users' identities. Some works [35]–[38] propose to let users exchange their pseudonyms when they meet in mix zones so as to break the moving pattern for each user, but may not provide real-time services. Notice that most of them assume a trustworthy and secure central server. In LocaWard, we can protect users' privacy against the system entities, including TDs, TCs, and the CC, without using any trustworthy central server.

Appendix B
Further Illustration of the Proposed Location-Based Rewarding Protocol

In this section, we first present the architecture of LocaWard in Fig. 1, in which the system entities include a Trusted Third Party (TTP), Mobile Users (MUs), Token Distributors (TDs), Token Collectors (TCs), and a Central Controller (CC).

We also briefly show the main operations involved in the identity authentication phase at a TD, the token distribution phase at a TD, and the token audition phase at a TC, in Fig. 2, Fig. 3, and Fig. 4, respectively.

Appendix C
Identity Initiation

In the identity initiation process, the TTP computes s_i as follows:

$$s_i = w^{\frac{1}{\epsilon_i}} \mod n_T = w^{d_i} \mod n_T, \qquad (1)$$

where $\epsilon_i d_i \equiv 1 \mod \phi(n_T)$. The derivation process of (1) is as follows. To prove $w^{1/\epsilon_i} \mod n_T = w^{d_i} \mod n_T$ is equivalent to prove $w \mod n_T = w^{d_i \epsilon_i} \mod n_T$. Let $l \geq 0$ be an integer, we have $w^{d_i \epsilon_i} \mod n_T = w^{1+l\phi(n_T)} \mod n_T$. Since $w^m \equiv 1 \mod n_T$ and m is a factor of $\phi(n_T)$, we can know that $w^{l\phi(n_T)} \equiv 1 \mod n_T$. Thus, $w \mod n_T$ is equal to $w^{d_i \epsilon_i} \mod n_T$ and equation (1) directly follows.

Appendix D
Security and Privacy of Identity Authentication at a TD

In this section, we give the detailed proof of security and privacy of the proposed MU's identity authentication phase at a TD. Recall that in the last step of the identity authentication phase, the TD checks if the following verification equation holds

$$x \equiv y^{\epsilon_i} \cdot w^{-r_{TD}} \mod n_T. \qquad (2)$$

The identity authentication succeeds if (2) holds, and fails otherwise.

Completeness A legal MU can always be successfully authenticated.

Proof: According to the authentication steps described above, if this is a legal MU who does own a valid ID issued by the TTP, i.e., the s_i corresponding to ϵ_i, then the right-hand side of equation (2) can be calculated as:

$$\begin{aligned} y^{\epsilon_i} \cdot w^{-r_{TD}} &\equiv (r_i' \cdot (s_i)^{r_{TD}})^{\epsilon_i} \cdot w^{-r_{TD}} \\ &\equiv (r_i')^{\epsilon_i} \cdot (s_i)^{\epsilon_i \cdot r_{TD}} \cdot w^{-r_{TD}} \\ &\equiv (r_i')^{\epsilon_i} \\ &\equiv x \mod n_T. \end{aligned}$$

Thus, if this MU is a genuine user in the system and the certificate $cert_i$ does belong to it, (2) always holds and the identity authentication can always succeed. □

Soundness An illegal MU can be successfully authenticated with a negligible probability.

Proof: Consider an illegal MU who does not know the s_i corresponding to ϵ_i. Suppose that it can correctly predict the question r_{TD}. Then, it can deceive the TD by choosing an r_i' such that $r_i' + r_{TD}$ can be divided by ϵ_i, and sending $x = w^{r_i'} \mod n_T$ and $y = w^{(r_i' + r_{TD})/\epsilon_i} \mod n_T$ to the TD. Thus, we get

$$\begin{aligned} y^{\epsilon_i} \cdot w^{-r_{TD}} &\equiv (w^{(r_i' + r_{TD})/\epsilon_i})^{\epsilon_i} \cdot w^{-r_{TD}} \\ &\equiv w^{r_i' + r_{TD}} \cdot w^{-r_{TD}} \\ &\equiv x \mod n_T, \end{aligned}$$

and (2) holds. However, the probability that the MU can correctly predict r_{TD} is $1/(n_T - 1)$. For a sufficiently large n_T, which is 1024-bit in our setting, this probability is negligible.

Next, we prove that it is impossible for an illegal MU to increase this probability. Specifically, when an illegal MU cannot correctly predict the question r_{TD}, it needs to be able to know $y = (x \cdot w^{r_{TD}})^{1/\epsilon_i} \mod n_T$ in order for

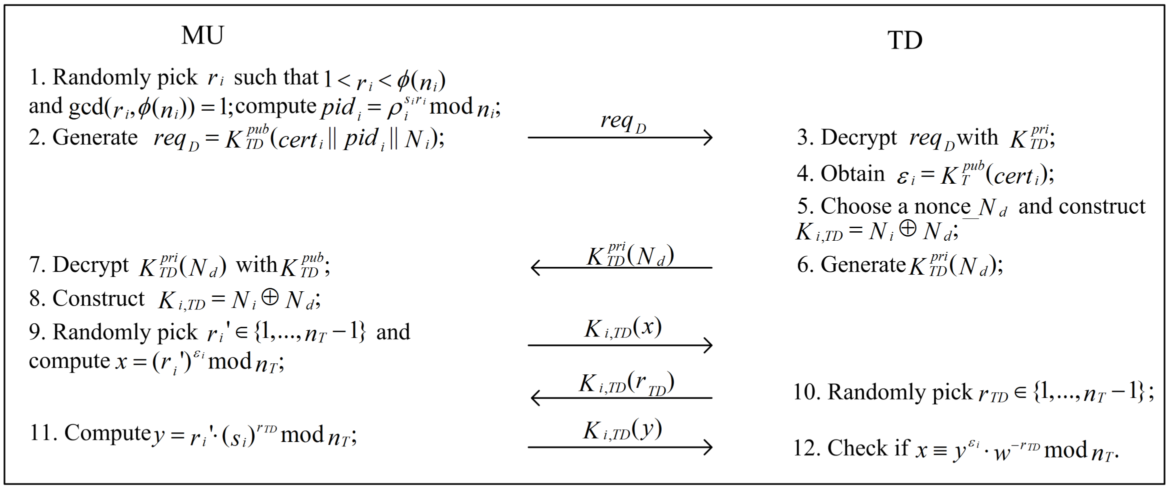

MU
TD
1. Randomly pick r_i such that $1 < r_i < \phi(n_i)$ and $\gcd(r_i,\phi(n_i))=1$; compute $pid_i = \rho_i^{s_i r_i} \bmod n_i$;
2. Generate $req_D = K_{TD}^{pub}(cert_i \| pid_i \| N_i)$;
req_D
3. Decrypt req_D with K_{TD}^{pri};
4. Obtain $\varepsilon_i = K_T^{pub}(cert_i)$;
5. Choose a nonce N_d and construct $K_{i,TD} = N_i \oplus N_d$;
6. Generate $K_{TD}^{pri}(N_d)$;
$K_{TD}^{pri}(N_d)$
7. Decrypt $K_{TD}^{pri}(N_d)$ with K_{TD}^{pub};
8. Construct $K_{i,TD} = N_i \oplus N_d$;
9. Randomly pick $r_i' \in \{1,\dots,n_T-1\}$ and compute $x = (r_i')^{\varepsilon_i} \bmod n_T$;
$K_{i,TD}(x)$
$K_{i,TD}(r_{TD})$
10. Randomly pick $r_{TD} \in \{1,\dots,n_T-1\}$;
11. Compute $y = r_i' \cdot (s_i)^{r_{TD}} \bmod n_T$;
$K_{i,TD}(y)$
12. Check if $x \equiv y^{\varepsilon_i} \cdot w^{-r_{TD}} \bmod n_T$.

Fig. 2. The MU's identity authentication phase. Note that from Step 9 to Step 12, x, r_{TD}, and y are encrypted/decrypted by the shared symmetric session key $K_{i,TD}$ between the MU and the TD.

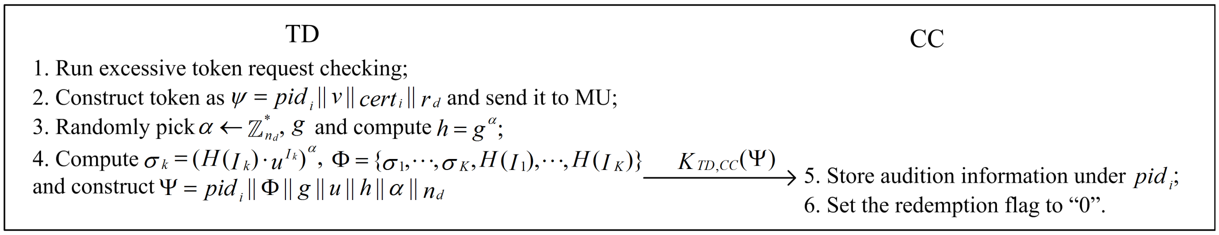

TD
CC
1. Run excessive token request checking;
2. Construct token as $\psi = pid_i \| v \| cert_i \| r_d$ and send it to MU;
3. Randomly pick $\alpha \leftarrow \mathbb{Z}_{n_d}^*$, g and compute $h = g^{\alpha}$;
4. Compute $\sigma_k = (H(I_k) \cdot u^{I_k})^{\alpha}$, $\Phi = \{\sigma_1, \cdots, \sigma_K, H(I_1), \cdots, H(I_K)\}$ and construct $\Psi = pid_i \| \Phi \| g \| u \| h \| \alpha \| n_d$
$K_{TD,CC}(\Psi)$
5. Store audition information under pid_i;
6. Set the redemption flag to "0".

Fig. 3. The token distribution phase. Note that in Step 4 and Step 5, Ψ is encrypted/decrypted by a pre-established shared symmetric key $K_{TD,CC}$ between the TD and the CC.

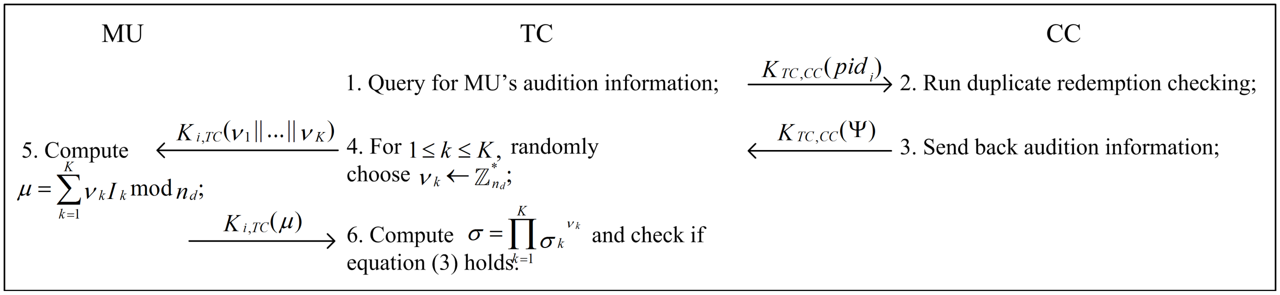

MU
TC
CC
1. Query for MU's audition information;
$K_{TC,CC}(pid_i)$
2. Run duplicate redemption checking;
5. Compute $\mu = \sum_{k=1}^{K} \nu_k I_k \bmod n_d$;
$K_{i,TC}(\nu_1 \| \dots \| \nu_K)$
4. For $1 \le k \le K$, randomly choose $\nu_k \leftarrow \mathbb{Z}_{n_d}^*$;
$K_{TC,CC}(\Psi)$
3. Send back audition information;
$K_{i,TC}(\mu)$
6. Compute $\sigma = \prod_{k=1}^{K} \sigma_k^{\nu_k}$ and check if equation (3) holds.

Fig. 4. The token audition phase. Note that the transmissions between the TC and the CC are encrypted/decrypted by a pre-established shared symmetric key $K_{TC,CC}$ between them, and the data exchanged between the TC and the MU is encrypted/decrypted by their shared symmetric key $K_{i,TC}$ established in the MU's identity authentication phase.

(2) to hold and to pass the identity authentication. Note that we have $x \cdot w^{r_{TD}} \mod n_T \in \mathbb{Z}_{n_T}$. Suppose that for $x \cdot w^{r_{TD}} \in \mathbb{Z}_{n_T} \backslash \mathbb{Z}_{n_T}^*$, this illegal MU can compute y in such a way with a nonnegligible probability, and hence can compute $(x \cdot w^{r_{TD}})^{1/\epsilon_i} \mod n_T$ regarding two questions $\widetilde{r}_{TD}$ and $\widehat{r}_{TD}$, denoted by $\widetilde{y}$ and $\widehat{y}$, respectively. Then, it means that the illegal user can calculate the following

$$\frac{\widetilde{y}}{\widehat{y}} \equiv \frac{(xw^{\widetilde{r}_{TD}})^{1/\epsilon_i}}{(xw^{\widehat{r}_{TD}})^{1/\epsilon_i}} \equiv w^{\frac{\widetilde{r}_{TD} - \widehat{r}_{TD}}{\epsilon_i}} \mod n_T,$$

i.e., the ϵ_i-th root of $w^{\widetilde{r}_{TC} - \widehat{r}_{TC}}$. Since w is a generator of a sufficiently large subgroup $\mathbb{G}_{n_T}$ of $\mathbb{Z}_{n_T}^*$, then $w^{\widetilde{r}_{TC} - \widehat{r}_{TC}} \in \mathbb{Z}_{n_T}^*$. This indicates that the MU can compute the ϵ_i-th root of element from $\mathbb{Z}_{n_T}^*$ in with nonnegligible probability without knowing factoring of n_T, which contradicts to the strong RSA assumption we described in Section 3.3.

Therefore, the soundness of the identity authentication phase directly follows. □

Privacy Reservation During the identity authentication phase, the TD cannot infer the MU's identity from their conversations.

Proof: During the identity authentication phase, the MU transmits req_D, x and y to the TD, out of which req_D, containing $cert_i$ (ϵ_i) and pid_i, and y are related to its identity s_i. The TD cannot compute s_i based on ϵ_i and y due to the strong RSA assumption. It cannot compute s_i given pid_i either, due to the DLP assumption. Thus, the MU's identity is protected in the authentication phase. □

Appendix E
Security and Privacy of Token Audition at a TC

In this section, we give the detailed proof of security and privacy of the proposed token audition phase at a TC. Recall that in the last step of the token audition phase, the TC checks the verification equation below:

$$e(\sigma, g) = e\bigg((\prod_{k=1}^{K} H^{\nu_k}(I_k)) \cdot u^{\mu}, h \bigg). \tag{3}$$

The TC determines that the MU's token is not forged or modified if (3) holds, and aborts the redemption process otherwise.

Completeness A valid token without being tampered can always pass the audition.

Proof: According to the token audition phase described above, if an MU owns a valid token which is also unmodified, we can have

$$\begin{aligned} e(\sigma, g) &= e\bigg(\prod_{k=1}^{K} \sigma_k^{\nu_k}, g \bigg) \\ &= e\bigg(\prod_{k=1}^{K} \big(H(I_k) \cdot u^{I_k} \big)^{\alpha\nu_k}, g \bigg) \\ &= e\bigg((\prod_{k=1}^{K} H^{\nu_k}(I_k)) \cdot u^{\sum_{k=1}^{K} I_k \nu_k}, g \bigg)^{\alpha} \\ &= e\bigg((\prod_{k=1}^{K} H^{\nu_k}(I_k)) \cdot u^{\mu}, g^{\alpha} \bigg) \\ &= e\bigg((\prod_{k=1}^{K} H^{\nu_k}(I_k)) \cdot u^{\mu}, h \bigg), \end{aligned}$$

i.e., (3) can always hold. □

Soundness Under the DLP assumption, the probability that a forged or tampered token passes the audition is negligible.

Proof: We prove the soundness of the token audition phase under random oracle model by using a sequence of games defined in [39]. Here, we develops two games, Game_0 and Game_1, to find out the success probability of an invalid or tampered token passing our token audition test. In particular, these two games are in an incremental mode: Game_0 is the real attack challenge game, whose success probability is what we are concerned about but cannot directly obtain; while Game_1 is a constrained game from Game_0, with a lower success probability which can be easily perceived. If we can prove that, first, the success probability of Game_1 is negligible, and second, the difference between the success probabilities of Game_0 and of Game_1 is negligible, then the soundness of the token audition phase directly follows.

Game$_0$: This is the real challenge game. A misbehaving MU (adversary) submits queries to a store oracle who provides one token ψ for each query. Then, in the following protocol executions, the environment plays the part of a TC. When the MU is asked to participate in the token audition, it can compute and submit the corresponding μ to the environment following the token audition phase as described before. We denote the success probability that the adversary passes this test in Game_0 as $\Pr_{\text{Game}_0}$.

Game$_1$: Game_1 is similar to Game_0, except that there is a simulator in Game_1 observing all the instances of token audition phase between the environment and the misbehaving MU. If in any of these instances the adversary is successful (i.e., the MU passes the verification equation (3)) but its submitted μ' is not equal to the expected value μ, the environment would still declare failure and abort the whole process. Obviously, Game_1 is a stricter game compared with Game_0. Denote the misbehaving MU's success probability in Game_1 by $\Pr_{\text{Game}_1}$. Then, the difference between $\Pr_{\text{Game}_0}$ and $\Pr_{\text{Game}_1}$ directly comes from the probability that the MU provides a μ' different from the expected value μ but the verification equation still holds.

Define $\Delta\mu = \mu' - \mu$, which is a nonzero value. We now show that the difference between $\Pr_{\text{Game}_0}$ and $\Pr_{\text{Game}_1}$ is negligible by contradiction.

Assume the simulator is given some inputs $\xi, l \in \mathbb{G}_1$, and its goal is to calculate x_s such that $l = \xi^{x_s}$. The simulator chooses random values $\beta, \gamma \xleftarrow{R} \mathbb{Z}_{n_d}^*$ and sets $u = \xi^{\beta} \cdot l^{\gamma} \in \mathbb{G}_1$. The simulator continues interacting with the adversary until the condition specified in the definition of Game_1 occurs: the adversary succeeds in responding with μ' that is different from the expected value μ. Equating the verification equations using μ' and μ gives us

$$e(\prod_{k=1}^{K} H(I_k)^{\nu_k} \cdot u^{\mu}, v) = e(\prod_{k=1}^{K} H(I_k)^{\nu_k} \cdot u^{\mu'}, v),$$

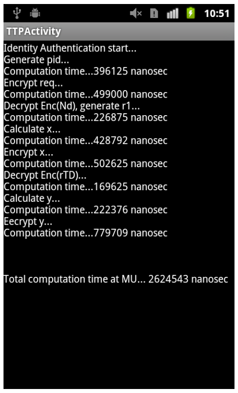
from which we have $u^{\mu} = u^{\mu'}$, and consequently,

$$1 = u^{\Delta\mu} = (\xi^{\beta} \cdot l^{\gamma})^{\Delta\mu}.$$

The above equation holds if and only if $\xi^{\beta} \cdot l^{\gamma} = \xi^{\beta + x_s\gamma} = 1$. Therefore, we can see that the simulator can thus find the solution to the discrete logarithm problem $l = \xi^{x_s}$, i.e., $x_s = -\beta/\gamma$, which contradicts with our DLP assumption, unless $\gamma = 0$, whose probability is $1/(n_d - 1)$ (n_d is a 512-bit prime as mentioned before). As a result, we can get that the difference between the adversary's success probabilities in Game_0 and Game_1 is $1/(n_d - 1)$, which is negligible.

Conclusion. In Game_1, the adversary is successful only when it is able to provide the real expected value μ. Since without the content $\{I_k | 1 \leq k \leq K\}$ of a valid token, the adversary can only obtain μ by guessing, obviously its success probability $\Pr_{\text{Game}_1}$ is negligible. Besides, since we have proven that the difference between $\Pr_{\text{Game}_0}$ and $\Pr_{\text{Game}_1}$ is negligible, the adversary's success probability in Game_0 which simulates the real attack from a misbehaving MU, i.e., $\Pr_{\text{Game}_0}$, is also negligible. This completes the proof. □

Fig. 5. Our testbed for the LocaWard system.



```
Identity Authentication start...
Generate pid...
Computation time...396125 nanosec
Encrypt req...
Computation time...499000 nanosec
Decrypt Enc(Nd), generate r1...
Computation time...226875 nanosec
Calculate x...
Computation time...428792 nanosec
Encrypt x...
Computation time...502625 nanosec
Decrypt Enc(rTD)...
Computation time...169625 nanosec
Calculate y...
Computation time...222376 nanosec
Eecrypt y...
Computation time...779709 nanosec



Total computation time at MU... 2624543 nanosec
```

Fig. 6. Step-by-step consumption time for identity authentication in token distribution process.

Privacy Reservation During the token audition phase, the TC cannot recover the MU's private information in the token.

Proof: As we can see, the only message sent from the MU that contains the token information $\{I_k | 1 \leq k \leq K\}$ in the token audition phase is $\mu = \sum_{1 \leq k \leq K} \nu_k I_k$. Although the TC knows the value of $\{\nu_k\}$ $(1 \leq k \leq K)$, and pid_i (i.e., I_1) and $cert_i$ (i.e., I_3) during identity authentication phase, it is still unable to calculate the value of v due to the existence of r_d. Obviously, when the token also carries the MU's other private information, e.g., for commercial purposes, r_d (i.e., I_4) can be omitted from the token and the private information in the token can still be preserved.

Besides, although the TC has the knowledge of α and Φ, it is computationally intractable for the TC to obtain I_k's. □

Appendix F
More Experiment Results

In this section, we implement the proposed LocaWard system on a testbed consisting of a laptop and an Android smartphone as shown in Fig. 5, and present more experiment results in terms of computation, communication, energy, and storage costs. The notations for the operations involved in the proposed protocol are also shown in Table 1.

TABLE 1
Notations of Operations.

Operations	Description
$Pair_{\mathbb{G}}$	Pairing $e(g,h)$, where $g,h \in \mathbb{G}$
$Exp_{\mathbb{G}}$	Exponentiation g^a, where $g \in \mathbb{G}$
$Hash_{\mathbb{G}}$	Hash values to group $\mathbb{G}$
Mul	Multiplication
Enc/Dec	Encryption/Decryption
Add	Addition
XOR	Exclusive or

F.1 Efficiency of Token Distribution – Single Token Request

In this section, we show the detailed computation, communication, and energy consumption costs in the token distribution process in Table 2, Table 3, and Table 4, respectively, in the case of only one single token request. We also show the step-by-step identity authentication done in one trial at the MU in Fig. 6.

TABLE 2
Computation Time in the Token Distribution Process.

Computation Time (ms)	MU	TD	Total
Identity Authentication	2.73	2.52	5.25
Token Distribution	0.45	0.69	1.14

TABLE 3
Communication Cost in the Token Distribution Process.

Identity Authentication	MU	TD	Total
Payload Size (bytes)	512	256	768
Communication Time (ms)	168.39	111.48	279.87
Token Distribution	MU	TD	Total
Payload Size (bytes)	0	320	320
Communication Time (ms)	0	59.32	59.32

TABLE 4
Energy Consumption of the MU in Token Distribution Process.

Token Distribution Process	Energy Consumption at MU (mAh)
Identity Authentication	0.84
Token Distribution	0.15
Total	0.99

F.2 Efficiency of Token Distribution – Multiple Token Requests

We then illustrate the efficiency of the token distribution process when there are multiple MUs in the system. The time between two adjacent arriving MUs is assumed to follow the exponential distribution with parameter λ_1. Besides, each MU requests one token as soon as it arrives. Note that in the experiment we actually use one smartphone, which is programmed to consecutively generate token requests following the exponential distribution, to simulate the requests generated by multiple MUs. As for the TD, we set its excessive token request checking window to 30 minutes, indicating that the TD

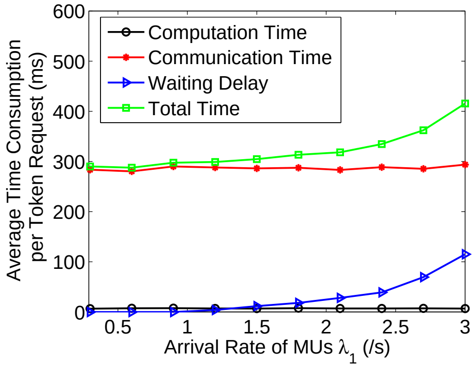


Fig. 7. Time consumption in the token distribution process under different MU arrival rates.

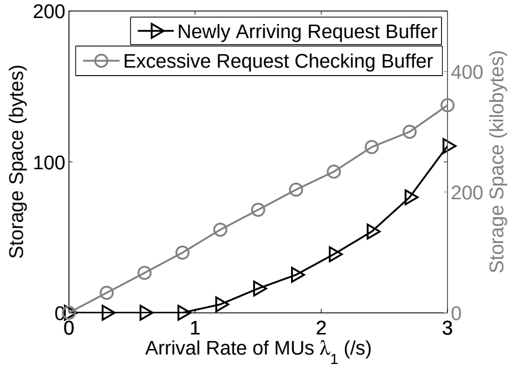


Fig. 8. Storage space required at a TD in the token distribution process under different MU arrival rates.

offers only one token to each requesting MU within 30 minutes.

Fig. 7 gives the average time consumption for each token request in token distribution process. We find that both the computation and communication time stay almost unchanged[1] as the arrival rate of MUs (λ_1) increases from 0 to $3/s$. Here, the computation time includes the time consumed by both the MU and the TD, the communication time indicates the time taken by all their communications, and the waiting delay is the average waiting time for each token request queuing in the TD's buffer before it is handled. We can see that as λ_1 increases to 3, the average waiting delay first remains 0 and then grows to around 115 ms, which is still within a reasonable range. Besides, obviously one way to solve the increasing waiting delay problem is to deploy multiple APs and have each of them handle part of the token requests.

We also show the storage space needed by the TD in token distribution process in Fig. 8. The storage space is needed mainly for two purposes, one for storing the newly arriving token requests if the TD is currently handling one, and the other for storing ϵ_i's within the excessive token request checking window. We notice the space required for second purpose is larger than that for the first one when $\lambda_1 \leq 3$. In addition, as λ_1 increases, the newly arriving request buffer size needs to increase linearly while the excessive request checking buffer size needs to increase faster.

F.3 Efficiency of Token Redemption – Single Token Request

In this section, we show the detailed computation, communication, and energy consumption costs in the token redemption process in Table 5, Table 6, and Table 7, respectively, in the case of one single token redemption request. Note that although the dominant computation complexity in the reward distribution phase at the TC and that in the identity authentication phase at the TC/TD are the same, i.e., $2 \times Exp$, the computation time in the former case, i.e., 20.62 ms, is much larger than that in the latter case, i.e, 2.95 ms. This is because we have to use a special function in jPBC to perform Exp in the reward distribution phase due to other paring operations. It is less efficient than the function we use in the identity authentication phase.

TABLE 5
Computation Time in the Token Redemption Process.

Computation Time (ms)	MU	TD	Total
Identity Authentication	2.8	2.95	5.75
Token Audition	0.67	218.75	219.42
Token Property Validation	0	22.5	22.5
Reward Distribution	0.47	20.62	21.09

F.4 Efficiency of Token Redemption – Multiple Token Requests

Next, we illustrate the experiment results when there are multiple MUs redeeming their tokens, which is simulated by one smartphone as before. The time between two adjacent arriving MUs is assumed to follow the exponential distribution with parameter λ_2. Besides, each MU can redeem multiple obtained tokens. In the experiment, we let all MUs have the same number of tokens, which is denoted by N.

[1] Note that since our experiment adopts one smartphone to simulate multiple token requests from different MUs, the obtained results for communication time actually do not take into consideration the delay caused by backoff waiting or collisions when multiple users coexist in the same domain. In fact, considering the TCP and IP packet headers (20 bytes each), the total transmission time of one token request can be calculated as $\tau = (512{+}256{+}320{+}40{\times}6){\times}8/(2{\times}10^6) = 5.31$ms, even if we assume the channel rate between the MU and the TD is as low as 2Mbps. Since the time between two adjacent arriving MUs follows the exponential distribution with parameter λ_1, then the arriving MUs follow a Poisson process and the probability of k arriving MUs within time interval 4τ is $\Pr(k) = \frac{(\lambda_1 \cdot 4\tau)^k}{k!} e^{-4\lambda_1 \tau}$. Thus, the probability that one token request does not affect any other requests can be estimated by $\Pr(k = 0) = e^{-21.24 \times 10^{-3} \lambda_1}$, which is about 94% even when λ_1 increases to $3/s$. This probability becomes 98.9% when the TD is connected to its AP via a cable. So the communication times we obtain here are good approximations. Besides, the transmission time of 5.31 ms is very short compared to the total communication cost of 339.19 ms. In addition, since the WiFi access point in our building already hosts a lot of users, the real communication time may be even less.

TABLE 6
Communication Cost in the Token Redemption Process.

Identity Authentication	MU	TC	Total
Payload Size (bytes)	512	256	768
Communication Time (ms)	166.24	112.75	278.99
Token Audition	MU	TC	Total
Payload Size (bytes)	64	256	300
Communication Time (ms)	56.13	58.45	114.58
Token Property Validation	MU	TC	Total
Payload Size (bytes)	0	0	0
Communication Time (ms)	0	0	0
Reward Distribution	MU	TC	Total
Payload Size (bytes)	32	0	32
Communication Time (ms)	55.97	0	55.97

TABLE 7
Energy Consumption of the MU in the Token Redemption Process.

Token Redemption Process	Energy Consumption at MU (mAh)
Identity Authentication	0.89
Token Audition	0.34
Token Property Validation	0
Reward Distribution	0.25
Total	1.48

Fig. 9 presents the average time consumption per user in the token redemption process. We can see from Fig. 9(a) and Fig. 9(b) that the average computation time and communication time of each user both stay constant as λ_2 grows, and increase linearly as N grows. Note that for an MU with multiple tokens, the MU only needs to pass the identity authentication phase once. Each token still has to pass the other three phases in the process before it can be redeemed. Fig. 9(c) shows the average waiting time of each user before it is taken care of. We can see that it remains 0 when $\lambda_2 \leq 0.04$ or $N \leq 2$, and increases as either λ_2 or N increases. In addition, Fig. 9(d) gives the average total consumed time in the entire token redemption process, which is the sum of computation time, communication time and waiting delay.

We also show the results of required storage space at the TC when λ_2 and N take different values in Fig. 10. The space is needed to buffer the newly arriving token redemption requests if the TC is currently dealing with some other MU. Note that the TC does not need to store the tokens since they are kept at MUs and handled sequentially after each MU passes the identity authentication phase. Similar to the waiting delay depicted in Fig. 9(c), the required storage space first stays 0 and then increases as either λ_2 or N increases. To keep the total processing time and the storage space needed at the TC within a reasonable range, we can also implement multiple APs to cooperatively handle all the token redemption requests from MUs.

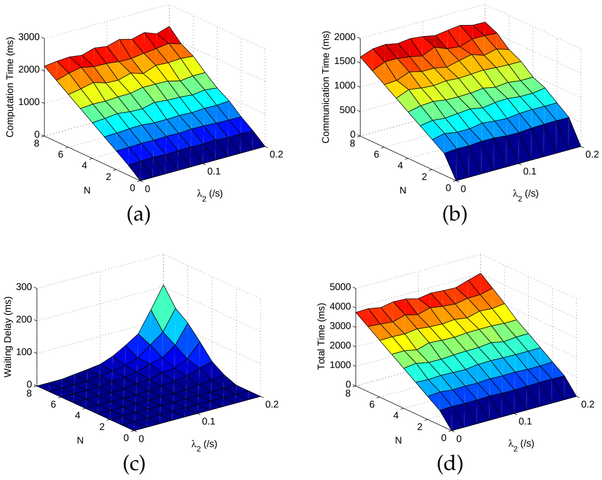


Fig. 9. Average time consumption per user in the token redemption process. (a) Computation time. (b) Communication time. (c) Waiting delay. (d) Total consumed time.

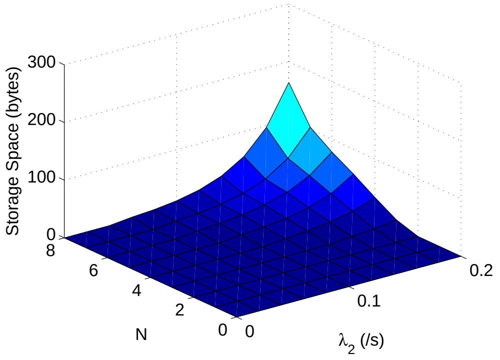


Fig. 10. Storage space required at a TC in the token redemption process under different MU arrival rates and number of tokens per user.

References

[1] https://foursquare.com/.

[2] http://www.facebook.com/about/location.

[3] W. Luo and U. Hengartner, "Proving your location without giving up your privacy," in *ACM HotMobile*, Annapolis, Maryland, February 2010.

[4] S. Saroiu and A. Wolman, "Enabling new mobile applications with location proofs," in *ACM HotMobile*, Santa Cruz, Califonia, February 2009.

[5] V. Lenders, E. Koukoumidis, P. Zhang, and M. Martonosi, "Location-based trust for mobile user-generated content: Applications, challenges and implementations," in *ACM HotMobile*, Napa Valley, Califonia, February 2008.

[6] S. Loreto, T. Mecklin, M. Opsenica, and H.-M. Rissanen, "Service broker architecture: Location business case and mashups," *IEEE Communication Magazine*, vol. 47, no. 4, pp. 97–103, 2009.

[7] http://www.loopt.com/about/tag/loopt-star/.

[8] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks," *IEEE Signal Processing Magzine*, vol. 22, no. 4, pp. 41–53, 2005.

[9] G. Sun, J. Chen, W. Guo, and K. R. Liu, "Signal processing techniques in network-aided positioning," *IEEE Signal Processing Magzine*, vol. 22, no. 4, pp. 12–23, 2005.

[10] M. Anisetti, C. A. Ardagna, V. Bellandi, E. Damiani, and S. Reale, "Map-based location and tracking in multipath outdoor mobile networks," *IEEE Transactions on Wireless Communications*, vol. 10, no. 3, pp. 814–824, 2011.

[11] Z. Zhu and G. Cao, "Towards privacy preserving and collusion resistance in location proof updating system," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, November 2011.

[12] B. Waters and E. Felton, *Secure, Private Proofs of Location*. Technical

Report TR-667-03, Department of Computer Science, Princeton University, January 2003.

[13] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *ACM WiSE*, San Diego, Califonia, September 2003.

[14] W. Luo and U. Hengartner, "Veriplace: A privacy-aware location proof architecture," in *ACM GIS*, San Jose, Maryland, November 2010.

[15] J. Kong and X. Hong, "Anodr: Anonymous on demand routing with untraceable routes for mobile ad-hoc networks," in *Proceeding of ACM MobiHoc*, Annapolis, Maryland, June 2003.

[16] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing source-location privacy in sensor network routing," in *Proceedings of IEEE ICDCS*, Columbus, Ohio, June 2005.

[17] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "Pda: Privacy-preserving data aggregation in wireless sensor networks," in *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, May 2007.

[18] M. Shao, S. Zhu, W. Zhang, and G. Cao, "pdcs: Security and privacy support for data-centric sensor networks," in *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, May 2007.

[19] B. Carbunar, Y. Yu, L. Shi, M. Pearce, and V. Vasudevan, "Query privacy in wireless sensor networks," in *Proceedings of IEEE SECON*, San Diego, California, June 2007.

[20] K. Ren and W. Lou, "A sophisticated privacy-enhanced yet accountable security framework for metropolitan wireless mesh networks," in *Proceedings of IEEE ICDCS*, Beijing, China, June 2008.

[21] C. Ardagna, S. Jajodia, P. Samarati, and A. Stavrou, "Providing mobile users' anonymity in hybrid networks," in *Proceedings of the 15th European Symposium on Research in Computer Security (ESORICS)*, Athens, Greece, September 2010.

[22] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *ACM Mobisys'03*, May 2003.

[23] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 1–18, January 2008.

[24] M. F. Mokbel, C. Y. Chow, and W. G. Aref, "The new casper: Query processing for location services without compromising privacy," in *Proceedings of VLDB*, 2006.

[25] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 12, pp. 1719 – 1733, December 2007.

[26] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized anonymization model," in *Proceedings of IEEE ICDCS*, Columbus, Ohio, June 2005.

[27] ——, "A peer-to-peer spatial cloaking algorithm for anonymous location-based service," in *Proceedings of ACM GIS*, Arlington, Virginia, November 2006.

[28] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proceedings of IEEE ICPS*, Santorini, Greece, July 2006.

[29] H. Lu, C. S. Jensen, and M. L. Yiu, "Pad: privacy-area aware, dummy-based location privacy in mobile services," in *Proceedings of ACM MobiDE*, Vancouver, Canada, June 2008.

[30] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *Proceedings of International Conference on Pervasive Computing*, Munich, Germany, May 2005.

[31] C. A. Ardagna, M. Cremonini, S. D. C. di Vimercati, and P. Samarati, "An obfuscation-based approach for protecting location privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 1, pp. 13–27, January 2011.

[32] A. Pingley, W. Yu, N. Zhang, X. Fu, and W. Zhao, "Cap: A context-aware privacy protection system for location-based services," in *Proceedings of IEEE ICDCS*, Montreal, Canada, June 2009.

[33] T. Xu and Y. Cai, "Exploring historical location data for anonymity preservation in location-based services," in *Proceedings of IEEE INFOCOM*, Phoenix, Arizona, April 2008.

[34] M. Terrovitis and N. Mamoulis, "Privacy preservation in the publication of trajectories," in *Proc. of MDM*, 2008.

[35] A. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 46–55, 2003.

[36] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Preserving privacy in gps traces via uncertainty-aware path cloaking," in *Proceedings of ACM CCS 2007*, Alexandria, VA, US, January 2007.

[37] X. Liu, H. Zhao, M. Pan, H. Yue, X. Li, and Y. Fang, "Traffic-aware multiple mix zone placement for protecting location privacy," in *Proceeding of the IEEE International Conference on Computer Communications (INFOCOM'12)*, Orlando, FL, USA, March 2012.

[38] J. Meyerowitz and R. R. Choudhury, "Hiding stars with fireworks: Location privacy through camouflage," in *Proceedings of ACM MobiCom*, Beijing, China, September 2009.

[39] H. Shacham and B. Waters, "Compact proofs of retrievability," *Proc. 14th Intl Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT 08)*, pp. 90–107, 2008.