RescueMe: Location-Based Secure and Dependable VANETs for Disaster Rescue

Jinyuan Sun, *Member*, *IEEE*, Xiaoyan Zhu, Chi Zhang, *Student Member*, *IEEE*, and Yuguang Fang, *Fellow*, *IEEE*

***Abstract*—Natural disasters and terrorism threaten our nation's safety and security, rendering post-disaster rescue mission critical. It is of paramount importance to carry out rescue work relying on secure and dependable networking. In this paper, we propose RescueMe, location-based vehicular ad hoc networks (VANETs), to aid in secure and dependable rescue planning for the efficient allocation of rescue resources. RescueMe leverages the location information stored during normal network operations to facilitate post-disaster rescue planning, while guaranteeing that the sensitive user location information is not exploited to trace a user's whereabouts when disasters are absent, even if the most powerful collusion attack is allowed. We provide a novel construction for the location update message, and propose several enhancements, to achieve the functional and security goals of RescueMe.**

***Index Terms*—Dependability, Disaster Rescue, Location Privacy, Redundancy, Security, vehicular ad hoc networks.**

I. Introduction

THE DEVASTATING consequences caused by recent natural disasters (e.g., Hurricane Katrina, Earthquake in China) and terrorism (e.g., 9/11) manifest the vulnerability of existing communication infrastructures, and indicate the need for dependable and resilient disaster rescue networks. The most important task of such networks is to provide necessary information for locating and rescuing survivors, which has higher priority than any other rescue and recovery work (e.g., saving public or private assets, repairing damaged radio towers). Locating potential survivors, and rescue planning where the number of people trapped in a geographic area is estimated for efficient allocation of resources (e.g., first aid kit, medical care personnel, ambulance, helicopter, etc.), are two key issues in rescuing survivors. Unfortunately, network connectivity which is leveraged to perform the locating and collection of useful information, cannot be guaranteed during or after disaster. Worst of all, the connectivity pattern is unpredictable due to the unforeseeable destruction to the infrastructures (e.g., fiber and cable connections, cellular base station, mobile switching center, wireless access point, power supplies) caused by the disaster.

Many research works have concentrated on the design of post-disaster networks to address the challenging connectivity issue. In particular, the disaster relief network can be modeled as a delay tolerant network (DTN) [1], [2] in that these two networks both feature communications in a disconnected fashion. However, there is a shift in communication paradigm since disaster relief network mainly involves finding a service (i.e., anycast) rather than end-to-end delivery (i.e., unicast). There are also works [3]–[8] that leverage cellular networks, wireless sensor networks, or other wireless networks to enable communications for post-disaster rescue and resource allocation. These works mainly focus on the functional aspects of disaster rescue networking with little security consideration (except some security issues pointed out by Ansari *et al.* [5]). Moreover, dedicated disaster response infrastructures in disaster areas have been proposed [9]–[11], which proactively address the unpredictable connectivity issue and may be reliable if designed and deployed according to the disaster type/characteristics of a particular geographic area. Nevertheless, dedicated infrastructures may not be available in many regions and thus an alternative and reliable solution is desired. Based on the above observations, we argue that 1) it will be desirable if we can adopt the proactive approach (i.e., pre-disaster preparation), while avoiding the requirement for dedicated infrastructures by building disaster rescue networks on existing communication infrastructures, and 2) since providing Internet access in disaster rescue networks is very challenging, we can relax the reliance on connectivity and explore other resources. Inspired by the evolving cache-and-forward packet delivery mechanism, we shift the reliance towards storage. We further observe that little attention has been paid in the literature to the related security and privacy issues, which are of paramount importance since the chaos created by disasters provides more opportunities for attackers. Last but not least, temporary roles (e.g., volunteers as rescuers) assigned solely for disaster rescue purposes may be abused to compromise users' location privacy. These arguments serve as the main design guide in our disaster rescue VANETs.

Our contributions. In this paper, we propose RescueMe, location-based secure and dependable VANETs for disaster rescue, which securely and routinely stores user location information using existing infrastructure/services, and reliably retrieves such information for disaster rescue. To the best of our knowledge, this is the first work to comprehensively study relevant security and privacy issues by characterizing the features of disaster rescue networks, and to exploit the unique communication capabilities of VANETs (i.e., vehicle-

Manuscript received 5 January 2010; revised 7 May and 12 July 2010.

J. Sun is with the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996 (e-mail: jysun@eecs.utk.edu).

C. Zhang and Y. Fang are with the Department of Electrical and Computer Engineering, University of Florida, PO Box 116130, Gainesville, FL 32611 (e-mail: zhangchi@ufl.edu, fang@ece.ufl.edu).

X. Zhu is with Xidian University, Xi'an, China (e-mail: xy.ok@hotmail.com).

Digital Object Identifier 10.1109/JSAC.2011.110314.

to-vehicle, vehicle-to-infrastructure) to fulfill the rescue requirements. The major challenge of RescueMe stems from the storage of sensitive location information. It is pertinent to user privacy since it reveals a user's whereabouts. On the one hand, users (i.e., future survivors) update and store their location information for future rescuers to roughly determine where the survivors could be trapped, in case the survivors cannot access the Internet to send rescue requests during or after disaster. On the other hand, the rescuers should not be able to abuse their rights to illegally trace a user during normal network operations (i.e., no disaster is present). In other words, users' location privacy should be context-aware or conditional, which is the core issue RescueMe attempts to resolve. Note that the functional requirement of RescueMe, i.e., rescue planning, overrides the security requirement, i.e., location privacy, during disaster rescue, although most security requirements defined in Section 3.2 can still be satisfied due to partial connectivity. The contributions of our work can be summarized as follows:

1. We characterize the unique features of disaster rescue networks, in terms of shifted design paradigm, different storage/retrieval requirement, and special use of location information, compared with existing networks/applications, in order to accurately define the design objectives.
2. We design the location update message, the key data structure serving as a building block of RescueMe, to achieve location privacy requirement for pre-disaster storage. We apply twists on the blind signature primitive in a non-straightforward way to construct the core component of the location update message, which is leveraged to calculate the trapped population in a geographic area for efficient rescue planning.
3. We propose to rely on redundancy storage (i.e., storing location update messages that are originated from a geographically similar area) to achieve dependability in our disaster rescue VANETs, given that the Internet is partially accessible.
4. We demonstrate the incorporation of RescueMe into the existing infrastructure/services through the techniques of piggybacking and locating/positioning, in accordance with the design objective of minimal deployment effort and negligible incurred overhead.

The rest of this paper is organized as follows. Section 2 describes the network and threat models. Section 3 identifies the challenges and design objectives of RescueMe. The design and construction of location update message are presented in detail in Section 4, followed by the elaboration of RescueMe in Section 5. Performance analysis and possible enhancements are detailed in Section 6. Finally, Section 7 concludes the paper with future work.

II. System Model

We overview the RescueMe network and present the network and threat models in this section.

A. Overview and Network Model of RescueMe

The entities in the RescueMe network include the users, rescuers, rescue authority, and key authority. RescueMe is built on the existing communication infrastructures and offers distinct network operations before or after disasters to facilitate the rescue work. The users are just the normal network users who enjoy various applications/services provided by the network when there is no disaster around, and become potential survivors when disasters occur. The users update their location information by attaching such information to their regular communication data when engaged in daily networking activities (e.g., downloading/uploading, data outsourcing, requesting location-based services, peer-to-peer communications). The location information is then stored with the data at the destinations (e.g., remote storage servers, service providers, peers). In the following context, we use the name storage server to indicate the various sources of storage. The stored location information across the network is expected to be accessible to assist in rescue work when the users become survivors of a disaster.

The rescuers and rescue authority are the roles temporarily assigned by the key authority for disaster rescue. They are played by existing regular roles (e.g., policeman, firefighter, survivors' families who volunteer) and the identity management authority, respectively. Specifically, the rescuers retrieve the user location information to perform rescue planning, which in RescueMe refers to accurately counting the number of people trapped in a geographic area for the purpose of rescue resource allocation. Users register at their local identity management authority, which could be the vehicle registration office, social security office, or any authority that manages identity.

The identity management authority issues tokens to registered users who will use these tokens to construct location update messages, such that the location privacy of the users can be guaranteed against illegal tracing when the network is in normal operations. Such tracing is possible since the rights assigned to rescuers for temporarily accessing the sensitive location information may be abused for non-rescue purposes. The identity management authority acts as the rescue authority that possesses certain access rights to the users' location information, in post-disaster rescue. Note that the key authority is temporarily set up to assign keys for disaster rescue and will be shut down during normal network operations. The identity management authority, given its importance, is assumed to have multiple backups, at least one of which will survive the disaster as part of the available infrastructures. The RescueMe disaster rescue network is depicted in Fig. 1.

B. Threat Model

The threat model defines the attackers and their possible attacks to RescueMe. The storage servers are honest but curious and will not maliciously delete or modify the location data. They are generally untrusted by users who outsource the information for storage. The rescuers are curious but reasonable or non-malicious in that they will attempt to learn the users' whereabouts while having no incentives to impede the functioning (i.e., rescue planning) of the disaster rescue network. The rescuers are allowed to collude with the key authority to obtain the decryption key and read the encrypted location information at will. Similarly, the rescue authority

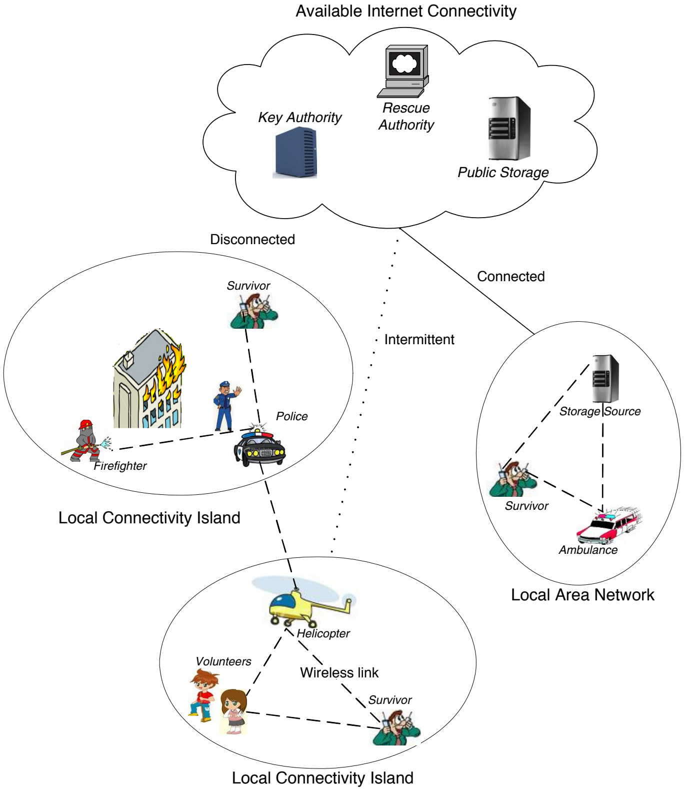


Fig. 1. The RescueMe Disaster Rescue Network.

will try to learn the identity and locations of the users, and is allowed to collude with the rescuers and key authority. In addition, active outsiders (e.g., terrorists, enemies) can inject bogus location information or modify legitimate information to destruct the rescue work. Passive eavesdroppers will attempt to intercept the location information on the fly.

III. Challenges and Design Objectives of RescueMe

Challenges/Characteristics of RescueMe. The connectivity and services provided by disaster rescue networks differ from existing wired and wireless networks in several ways. The first step in designing a disaster rescue network is thus to identify the unique characteristics and challenges which will assist us in defining the objectives of such network.

1. Shift in the main design paradigm: Disaster rescue network features totally unpredictable connectivity as a result of unforeseeable infrastructure damage. Specifically, the network connectivity can be intermittent, isolated (i.e., connectivity islands), or partially connected (i.e., some parts of the infrastructure are available). Apparently, we cannot count on such type of network to relay information simply because no delivery can be guaranteed. The main goal of disaster rescue network is therefore shifted from end-to-end delivery to persistent storage and redundancy storage. Persistent storage requires the information (e.g., rescue request messages) to be stored at available sources as long as possible so that it will eventually reach the infrastructure. This storage goal is with respect to post-disaster information storage, in the sense that after the network is disconnected, location updates can be stored in the local connectivity islands and delivered as nodes in these islands reach out. Redundancy storage indicates that information should have multiple copies and be stored in a distributed fashion across the network so that at least one copy can be accessed through the available infrastructure (i.e., that part of the infrastructure is undamaged or slowly recovered). This storage goal is with respect to pre-disaster information storage and captures the following fact: since the network will be unavailable or the survivors may be unable (e.g., physically injured) to send rescue requests during or after disasters, the rescuers can rely on the pre-stored location information to more efficiently search for survivors. The redundancy storage goal serves as the major guide in the design of RescueMe. The persistent storage issues are not addressed in this paper.
2. Different storage/retrieval requirement: As the redundancy storage requirement specifies, pre-storage of location information before disasters would be of great importance to post-disaster rescue. However, privacy concern is raised when storing location updates during normal network operations, since location is sensitive information which can be exploited to illegally trace a person's whereabouts. How to ensure that the location information is retrieved by designated personnel only when disaster occurs (context-aware or conditional retrieval), in other words, how to ensure that little useful sensitive information will be leaked even if the location informaiton is retrieved illegally in normal operations, is a very challenging yet important issue to be addressed in RescueMe.
3. Special usage of location information: In most location-based applications, the location information is mainly used to locate entities and services of interest. For example, package and personnel tracking, lost-device tracking, requesting the nearest restaurants and gas stations, navigation, receiving notification of traffic and road conditions, 9-1-1 emergency services, etc. In RescueMe, however, the location information is mainly used to estimate the number of trapped people in a geographic area, in order to facilitate rescue planning. The usage here is different from the traditional sense in that the identity of an individual (i.e., who is trapped) is less of a concern than whether and how many people are trapped, from the rescue planners' perspective.

Design Objectives of RescueMe. Based on the above characteristics of disaster rescue network, we envision that RescueMe should satisfy the following main functional and security objectives.

1. Pre-disaster location privacy: This requirement is indispensable to preserving the location privacy of network users in normal operations. Location privacy is threefold: anonymity, unlinkability and timing attack proof. Anonymity specifies that the sender of any received location updates cannot be identified, or at least cannot be distinguished among a group of senders (i.e., sender ambiguity). Unlinkability requires that any two or more

location updates cannot be linked to have originated from a same sender. Timing attack proof states that the transmission pattern of location updates cannot be exploited to identify the sender, nor can the actual transmissions be tied to their corresponding receptions at the storage site. The location privacy requirement should be fulfilled regardless of the access rights of entities, even if collusion is allowed among entities in the network.

2. Post-disaster locatability: Survivors should be locatable by the rescuers in post-disaster rescue leveraging the location information stored beforehand, where location privacy requirement can be overridden (i.e., some sensitive information can be revealed by the rescue authority).
3. Dependability: It is essentially indicated by redundancy storage, which enhances the chance of acquiring survivors' location information and is defined in **Challenges/Characteristics of RescueMe**.
4. Access control: Different roles in RescueMe will have different rights in accessing survivors' location information. Specifically, storage servers merely provide storage for the encrypted location information but have no access to this information. Rescuers should only be able to collect encrypted location information from the storage site and read the insensitive portion of such information, e.g., "I am at Archer Rd. and 34th St. on 07/25/2009 13:48" to filter out less useful data. The rescue authority should have higher access right to the encrypted location information in order to estimate the number of survivors in an area and plan the rescue accordingly. Note that the rescue authority can delegate its access right to the rescuers using secret sharing technique [12]. Neither the rescue authority nor the rescuers will have access to the identity information of the survivors in the encrypted location data, since such information is unnecessary in determining the number of trapped survivors. The families and friends, on the other hand, can access a survivor's identity information which is of the highest concern to them.
5. Minimal deployment effort: This objective is desirable in the design of disaster rescue network. It requires no dedicated infrastructure but the reliance on the existing networks/services. The reason for this requirement is intuitive: disasters are relatively rare events which means the dedicated infrastructure may have no or very low utilization. The most prominent benefit of building RescueMe on existing networks is obviously the huge savings in deployment and maintenance costs incurred by dedicated disaster rescue networks.

IV. Core Design: Location Update Message

We assume for now a secure and anonymous communication backbone via which the location information will be stored and retrieved. The realization of such backbone will be explained in Section 5.3. This section focuses on the core techniques of RescueMe, the construction of the location update message to enable secure and dependable storage/retrieval.

A. Constructing Location Update Message for Secure Storage

The storage of location update messages is performed in normal network operations as the preparation for future disaster rescues. The location update message should be constructed in such a way that 1) it preserves location privacy when no disaster occurs, while is still usable for future disaster rescue, and 2) minimum necessary sensitive information need be leaked to facilitate the rescue. First of all, the location update messages should be encrypted to prevent eavesdroppers and the untrusted storage servers to learn useful information. More challengingly, the rescuers who possess the decryption key by colluding with the key authority can read the messages, but should not obtain useful information to undermine the privacy, i.e., the anonymity, unlinkability, and timing attack proof properties of the location information, during normal network operations. In other words, the rescuers will be unable to abuse their access rights. More challenging still, the rescuers should correctly count the number of people trapped in a location, which means that two or more update messages should be identified to come from a same user (without recovering the user identity), and the population counter should be incremented by only one. This functionality is actually contradictive to the unlinkability property which requires that updates cannot be linked to have originated from a same user. In addition, the location update messages of a particular user should be identifiable by his/her families and friends who are likely to volunteer.

To address all these issues, the location update message is constructed by the user as follows:

1) Generating $SKE(ID_{user})$: The user encrypts the identity information such as name, email address, phone number, etc., using any semantically secure symmetric key encryption (SKE) scheme. The secret key of SKE is generated by a pseudorandom number generator (PRNG), which takes as input a secret seed s shared between the user and families/friends, and outputs pseudorandom secret keys, one for each location update message. The PRNG can be constructed from block ciphers (e.g., AES), hash functions (e.g., SHA-1), modular exponentiators, or linear feedback shift registers (LFSRs) based on stop-and-go generators [13]. The SKE enables the families/friends and no one else to learn the identity of the user based on the collected location message. The different ciphertexts (of a same original plaintext, i.e., the identity information) produced by different secret keys prevent the rescuers from linking location update messages to a same user.

2) Generating $BSIG$: The user obtains a blind signature or token $S'(x)$ on user-chosen information x from the rescue authority. The user then challenges him/herself with d and self-responses with r_1, r_2. The user forms $BSIG$ for counting purposes as $BSIG = (r_1, r_2, Other)$, where $Other$ contains necessary information for verifying the token and will be discussed in Section 6. $BSIG$ is the key information in the location update message to eliminate over-counting for disaster planning, which will be demonstrated in Section 4.2. We postpone the protocol and instantiation of $BSIG$, and the design

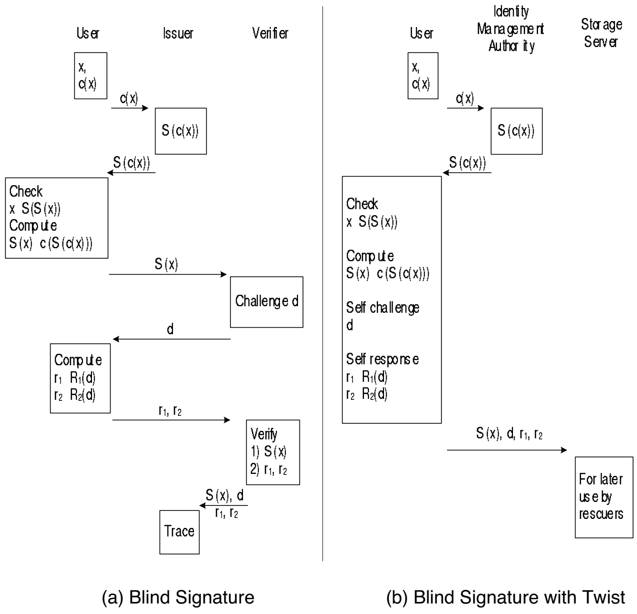
rationale behind the self-challenge/response to Section 4.3.

3) Generating location update message: The user forms the final location update message to be stored as $PEKS_{role}(location,\ BSIG, SKE(ID_{user}))$, where $location$ denotes the insensitive portion of location information (e.g., "My location for disaster rescue: Archer Rd. and 34th St. at 13:48 07/25/2009"). The plaintext message $(location, BSIG, SKE(ID_{user}))$ is encrypted using the public key of a role (e.g., "$Disaster\_Rescuer$"). The syntax of the role need be agreed upon so that the person in that role can successfully obtain the corresponding private key. The searchable public key encryption $PEKS$ enables the storage server to retrieve messages containing certain keywords, e.g., "My location for disaster rescue", without learning any other content of the message.

B. Secure Retrieval of Location Update Message

As will be clear in this section, the construction in the previous section serves as the preparatory step to ensuring secure retrieval of location update messages, leaking no sensitive information to the storage server and the rescuers. The location update messages are retrieved upon disasters and the trapped population in a geographic area is counted as follows:

1) The rescuers obtain the private key corresponding to the public key of a role (e.g., "$Disaster\_Rescuer$"), and uses the private key to decrypt $PEKS_{role}(location, BSIG, SKE(ID_{user}))$ which is retrieved by the storage server (i.e., the available infrastructure) upon the rescuers' request for messages a) containing "My location for disaster rescue" keywords and b) received in a specified time interval (e.g., within a day before disaster).
2) The rescuers sort the decrypted messages to several divided geographic areas, and count the trapped population for each area. The rescuers first get rid of the duplicate copies of location update messages that were stored for redundancy reason, based on the same $BSIG$ contained in these messages (cf. Section 5.1), since it is possible that multiple copies of a message are retrieved through the available infrastructure. Note that these messages record geographically similar locations, and thus the linkability induced by the same $BSIG$ here gives away no useful information.
3) The rescuers then identify location update messages that are generated in different geographic areas but are from the same user. On the one hand, it is imperative since the rescuers are concerned with the latest location where the survivor is mostly likely to be trapped. The past location update messages retrieved from the available infrastructure in this case can be misleading and cause inefficient planning. On the other hand, it is very challenging because it violates the unlinkability requirement of RescueMe as explained before. This is where the information in $BSIG$ comes into play. According to the techniques of restrictive blind signature, different response pairs (i.e., (r_1, r_2) in $BSIG$) for geographically different location update messages can be generated using the same token (i.e., $S'(x)$), so that $BSIG$ appears different but the same value computed from the response pairs can be recovered to indicate the same sender. Direct application of this technique creates a subtle linkability attack in our context, which will be explained in Section 6 together with its countermeasure. We provide more details in Section 4.3 on how we place non-intuitive twist into the original restrictive blind signature primitive to suit the disaster rescue network.



(a) Blind Signature

(b) Blind Signature with Twist

Fig. 2. Illustration of Blind Signature with Twist.

C. Blind Signature with Twist

The symmetric key encryption primitive and searchable public-key encryption primitive used to construct location update messages are applied in a straightforward way. The blind signature primitive, however, involves twist in both the protocol execution and the purpose/function of the original primitive which cannot be applied in our disaster rescue network in a straightforward manner. In this section, we explicitly point out the differences with the original primitive and the suitability of the twist in our context.

1. Twist in purpose/function: Traditionally, double-spending a coin is considered a misbehavior that should be prevented. Users who desire privacy will not be tempted to double-spend, knowing that the consequence could be the recovery of his/her identity. In RescueMe, this "misbehavior" becomes a desired property for eliminating over-counting. Specifically, users "double-spend" a token to imply that multiple location update messages are from a same sender, and therefore the trapped population in an area can be accurately determined.
2. Twist in protocol execution: Provided with the above design concept, there is a corresponding change in the protocol execution. As shown in Fig. 2 (a), in the traditional scenario, the user spends a coin at a verifier for

purchase. The verifier initiates the challenge-response procedure with the user, to check if the coin was blindly signed by the issuer and decide on the acceptance of the coin. The verifier later submits the coin and the results of the challenge/response procedure to the issuer for the identification of double-spender. In our scenario, nevertheless, the user performs the challenge-response procedure by him/herself and stores the results (computed from the token $S'(x)$), which will be used by the rescuers to later count the trapped population (i.e., by checking for double-spending and eliminating redundant messages). Therefore, the protocol execution involving the verifier in the traditional scenario, is now executed partially by the user and partially by the rescuers in our scenario. The comparison of the protocol execution in both scenarios is best illustrated in Fig. 2.

Instantiation of blind signature. In Fig. 2 (b), the user blinds x, the message to be signed by the issuer (i.e., the identity management authority), to obtain $c(x)$. In the traditional primitive [14], the message x embeds identifying information which can be later used for tracing. This feature is not needed in our scenario, since our primary goal is to determine whether two location messages are sent by a same user, the identity of whom need not be known. This is implied in the aforementioned twist that "double-spending" serves as a desired property in our disaster rescue network. The issuer uses its private key to sign $c(x)$, and sends the signature $S'(c(x))$ back to the user. The user checks if $x = S(S'(x))$ holds to assure the validity of the issuer's signature, where $S(\cdot)$ is the signature verification algorithm based on the issuer's public key. The user then removes the blinding factor using $c'(\cdot)$ and obtains the desired token $S'(x)$, such that the issuer cannot recognize $S'(x)$ and therefore cannot link $S'(x)$ to the user's identity even if $S'(x)$ is shown later. This token is used to eliminate over-counting in the rescue planning. Different tokens $S'(x)$'s can be acquired by repeating the protocol execution in Fig. 2 (b). The self challenge/response follows the same procedure as in the traditional challenge/response shown in Fig. 2 (a). The resulting information d, r_1, and r_2 is necessary for verifying the validity of $S'(x)$ to ensure the legitimacy of the user. Any provably secure restrictive blind signature construction such as [14]–[16] can be leveraged to instantiate the illustrated blind signature protocol in Fig. 2. Interested readers are referred to [17] for an application example where the restrictive blind signature instantiation in [16] is employed.

V. Location-Based Secure and Dependable Disaster Rescue Networking

Having presented the design rationale and details of location update message, we are now ready to dive into the networking aspects.

A. Privacy-Preserving Location Information Dissemination

Redundancy storage. The accuracy of the user's location need not be high in our disaster rescue network, since the rescuers only require the locatability of survivors in a rough area where the rescue work can be guided. Once the rescuers enter that area, more location information could be collected within the local connectivity islands where the latest location information waits to be transmitted through recovering connections. Therefore, we consider the location information contained in the multiple transmissions/updates from a same *geographic area* to be equivalent, i.e., these transmissions record a same location (e.g., 3800 SW 20th Ave.) even though some of them record the neighborhood (e.g., the gas station three blocks away). We do not pursue on how to define a geographic area for the redundancy storage purpose. It is a design issue depending on the degree of redundancy and locating accuracy. These multiple transmissions from *an area* essentially achieves redundancy storage in RescueMe. It differs from the common sense redundancy storage for storing location information, where multiple transmissions from *a location* are stored. We will show the advantage of this approach shortly in Section 5.2. The multiple updates constituting a redundancy storage contain the same $BSIG$ (serving the purpose of a tag) for future rescuers to easily eliminate the redundancy if more than one of these updates are retrieved. Updates from different areas have different $BSIG$'s formed by the same token $S'(x)$ (or different tokens, cf. Section 6), so that all but the most recent location of a user can be disregarded. These updates are disseminated according to the mechanism described below.

Disseminating location information. The construction of location update message ensures the anonymity/ ambiguity and unlinkability requirements of location privacy. To thwart the timing attack launched by global observers, the transmission of location update messages should not be event-triggered. For example, if the event is to send an update message when the location changes, the actual transmission of the message, which can be detected by the Internet connection of the device, should not occur as location of the user changes. Instead, the trigger should incorporate some source of randomness so that the transmissions cannot be predictable even if the events (i.e., location changes) are observed. Recall that we use a pseudorandom number generator (PRNG) with a secret seed s to generate secret keys for encrypting the user identity in each location update message. This technique can again be employed to produce pseudorandom inter-transmission times of the location update messages, rendering the actual transmission to appear equally likely from any user and thus unlinkable to the observed event. The PRNG is fed with a secret seed s' and outputs inter-transmission times for scheduling the transmissions. The seed s' can be randomly selected or generated from s by inputting s to the PRNG. The advantage of the latter approach is that the user will need to store only one secret, s, for both the pseudorandom secret keys and inter-transmission times. The user sets an initial time reference based on the clock of his/her device (e.g., cell phone, PDA, laptop, vehicle), and adds the generated inter-transmission times to schedule each transmission.

B. Building RescueMe on Existing Infrastructure

Piggybacking location update messages. We have mentioned that the redundancy (or duplicates) in RescueMe refers to the location update messages for a same geographic area, not a same location. The main reason for this design is to utilize the existing infrastructure and services with minimum possible modification, since we attempt to fulfill the design goal of minimal deployment effort and negligible added overhead to the existing network. Users nowadays are offered more and more varieties of Internet applications/services, e.g., storage outsourcing, video sharing, online shopping, instant messaging, web surfing, location-based services (LBS), etc. Users routinely enjoy the tremendous benefits and convenience as a result of ubiquitous network access. RescueMe can exploit this trend to piggyback the location update messages for disaster rescue onto daily packets/messages transmitted and stored in the above applications/services. The piggybacking need not be performed on every application packet/message, but on those whose sending times approximate the scheduled inter-transmission times of location update messages.

Locating/positioning. In order to update locations and store them for rescue use, the user device should be able to locate itself. Depending on the type and capability of the devices, such locating/positioning can be realized through the Global Positioning System (GPS), the Real Time Locating System (RTLS), GSM beacons, tracing routes to nearby routers, external and internal IP addresses, or the addresses (of the device itself or its nearby facilities/businesses) returned by a location-based service. Since RescueMe requires relatively low location accuracy, many locating methods can be readily applied. In particular, as the location-based services prevail, users can easily learn their current locations by requesting a service (e.g., querying for close-by restaurants), which involves no specialized equipment or software. We envision that at least one of the locating/positioning techniques will be available for acquiring $location$ in the location update message.

By proposing to use piggybacking for location information delivery, and widely available locating techniques for location information acquirement, we attain the design objective of minimal deployment effort by placing no dedicated infrastructure and negligible overhead into the existing network. Note that some computation overhead is incurred at the user side during the construction of location update messages. However, the only computationally expensive pairing operations involved in ID-based encryption and searchable public-key encryption can be pre-computed, once and for all location update messages. The blind signature may involve pairing operations that can be pre-computed but need to be performed for each token generation. Such computation load is considered light even on low-end devices such as cheap sensors, needless to say on much more powerful user devices.

C. Other Key Design Issues

Role assignment and access control. The identity management authority and existing roles (e.g., firefighter, policeman) play the special roles of rescue authority and rescuer, respectively, for disaster rescue, and should relinquish the access rights associated with the special roles after the rescue. The role-based encryption adopted in RescueMe network provides first line of defense against abusing access rights when the network is under normal operations. Since the key authority is temporarily created to assign private keys corresponding to roles and is shut down after disaster, it is very unlikely to compromise the key authority to illegally obtain the private keys. Although the assigned private keys may continue to be used after disaster rescue, such keys can be easily rendered invalid by the user encrypting future messages under a distinct role string (e.g., "$Disaster\_Rescuer\_T$" where T is the end time of the most recent disaster). Without a correctly generated private key for the role under which the messages are encrypted, it is cryptographically intractable to decrypt the ciphertext messages.

We will see in Section 6 that the rescuers need approval from the rescue authority to determine if two or more location update messages are from a same user. This is the second line of defense against the rescuers' abuse of access rights in normal operations, in case they collude with the key authority to illegally acquire the role-based private key. This step is indispensable in preventing the rescuers to link multiple location messages to trace a user's whereabouts. Although this tracing ability of the rescuers does not compromise the anonymity/ambiguity objective (i.e., the user's identity is unknown), it subverts the unlinkability objective of the location privacy requirement. To avoid single point of failure, the rescue authority can split its role-based private key to a threshold number of rescuers by employing the secret sharing technique [12], such that any number of rescuers below the threshold cannot recover the private key for decrypting certain sensitive location information. Another benefit of this sharing is to raise the bar for collusion attacks by the rescuers, through the increase of the threshold value.

Public Key Infrastructure (PKI). Identity-based (ID-based) public key infrastructure [18] should be employed in RescueMe, since the role-based encryption relies on the unique property of ID-based PKI, i.e., both the public key and the corresponding private key can be assigned posterior to the encryption using the public key (i.e., the ID), so long as this public key is known at the time of encryption. The ID-based PKI also enables authentication and key establishment in RescueMe, and hence serves as the secure backbone we have assumed prior to the descriptions in Section 4. Authentication takes place in RescueMe right before 1) the users obtain tokens from the identity management authority, 2) the rescuers request the rescue authority to decrypt a portion of $BSIG$ and verify a token (cf. Section 6), or 3) the key authority assigns role-based keys to the rescue authority and rescuers. Authentication assures the authenticity of an identity and the authorization of an individual. Key establishment occurs when the parties involved in authentication need to establish shared secret key to facilitate further efficient communications.

Anonymous substrate. In addressing privacy and anonymity on the Internet, Dingledine [19] argues that cryptography alone will not hide the existence of confidential communication relationships and implemented an anonymous communication overlay network, Tor [20], based on the anonymous routing protocol, i.e., the onion routing [21]. In addressing the privacy preserving issue in vehicular ad hoc networks (VANETs) where the vehicles enjoy various VANET applications, Raya and Hubaux [22] claim that all vehicle identifiers, in particular the MAC and IP addresses, must change over time, in addition to the frequent update of the anonymous keys (pseudo-identities). Analogously, the proposed location-based privacy preserving disaster rescue network relies on effective anonymous communication substrate [19], [20] to obfuscate the transmissions of location update messages, in addition to all the privacy-preserving techniques proposed in the design of RescueMe such as the use of tokens, pseudorandom secret keys (for SKE) and inter-transmission times, etc. Otherwise, if the network ID (i.e., IP address, MAC address) of a user device is fixed and exposed in packet forwarding, the location privacy guarantees claimed in RescueMe will be undermined.

VI. Analysis and Enhancements

This section elaborates on how the security requirements are achieved in RescueMe based on the objectives and threat model defined before, and how to enhance the resilience of RescueMe to some subtle attacks that have not been addressed in previous sections. The efficiency of the proposed scheme in terms of storage and computation is also discussed.

A. Security Analysis

Location privacy. As specified in Section 3.2, this security objective requires anonymity/ambiguity, unlinkability, and timing attack proof in our disaster rescue context. First, we consider the simplest case where no collusion is allowed. The timing attack proof requirement, mainly against global observers, has shown to be fulfilled by the pseudorandomly generated inter-transmission times and the anonymous substrate. The former removes the predictability of location message transmissions and the latter breaks the link between transmissions and receptions. With the pseudorandom number generator (PRNG) in place, it is extremely unlikely for other entities in the network to successfully launch timing attacks. In what follows, we focus on discussing the anonymity/ambiguity and unlinkability requirements. 1) For the storage server: Anonymity/ambiguity is guaranteed through the encryption of the location messages and the anonymous communication substrate, such that the storage server learns neither the content of the messages nor the identity of the sender. Since the ciphertexts appear random and thus no fixed pattern can be leveraged to deduce whether they are from the same sender, the unlinkability requirement can also be easily satisfied. 2) For the rescuers: When no disaster takes place, the private key corresponding to the rescuer role cannot be acquired from the key authority. In this case, even if the location messages are retrieved from the server by specifying the keywords, the ciphertexts cannot be decrypted. Therefore, the rescuers, as the storage server, obtain no useful information to compromise either anonymity/ambiguity or unlinkability. 3) For the rescue authority: As usual, the encrypted location messages reveal no useful information for the rescue authority to compromise location privacy. The only possible way for the rescue authority to learn the identity of location messages or link these messages is through collusion.

Collusion resistance and security enhancements. In our disaster rescue context, not all collusions are meaningful. Colluding with the storage server will not yield useful information, since the server has no access rights to any user information. Collusion among rescue authority, rescuers, and key authority can be damaging: rescue authority manages the identity of the users and engages in token generations, rescuers can potentially decrypt the ciphertexts to read location messages, and key authority controls the issuance of such a decryption key. We now discuss what we believe the most powerful collusion attack, the one launched by the collusion of all these entities, and propose enhancements to the existing RescueMe network. We first describe the collusion between rescuers and key authority, and discuss the resulting attacks on location privacy and some unsuitable countermeasures. It articulates the reason for the design proposed in our paper. We then proceed to incorporate the corrupted rescue authority in the collusion framework and show the resilience of our disaster rescue network. In the two-party collusion attack, the key authority is corrupted and issues the role-based private key to the rescuers whenever the key is demanded, even if no disaster is around. Equipped with the private key, the rescuers can decrypt and read users' location messages at will, abusing their access rights. The anonymity/ambiguity will be preserved because the location messages contains no (readable) identity information. However, the unlinkability guarantee is challenging in that no information/pattern in the location messages should enable the rescuers to link these messages, given that now the rescuers can decrypt these messages at will.

1) Two-party collusion: This design challenge necessitates the employment of our blind signature with twist. As mentioned before, we use PRNG to generate pseudorandom secret keys for encrypting the user's identity in each location message, enabling families and friends to locate survivors. Since the identity is encrypted under a different secret key each time, the resulting ciphertexts appear random and hence unlinkable with each other. The information in $BSIG$, indispensable for rescue planning, serves the same unlinkability purpose. The different response pairs derived from the same token render no linkability at a first glance. However, the restrictive blind signature yields an additional functionality that is not needed and will cause a subtle linkability attack in RescueMe. When multiple location update messages $M = (m_1, m_2, \cdots, m_k)$ from the same user are collected by the corrupted rescuers, a simple check $\frac{r_1 - r_1'}{r_2 - r_2'}$ [14] can be applied to each pair of messages in M (e.g., m_1 and m_2, m_2 and m_3, etc). If the checks return the same result, the rescuers can deduce that these messages are from the same user and break the unlinkability. We thus propose modification to $BSIG$ as $BSIG = (d \oplus r_1, d \oplus r_2, IBE_{Rescue\_Authority}(d \parallel S'(x)))$, where $\oplus$ denotes the exclusive-or operation that is used to thwart the above linkability attack. $IBE_{Rescue\_Authority}$ denotes ID-based encryption [18] under the rescue authority's role string, and is incorporated to restrict the access to the information $d \parallel S'(x)$ that can potentially cause linkability, such that only rescue authority can access this information. As a result, the rescuers are unable to perform the linkability attack without colluding with the rescue authority. Note that IBE produces different ciphertext upon each encryption even if the same plaintext is used as input. More details on this encryption will be covered in the three-party collusion attack below.

A naive approach to the linkability attack would be to simply place a different sequence number in location messages that are from different geographic areas. If two or more such messages are retrieved in post-disaster rescue, the rescuers will be unable to determine that they are from a same sender which affects the accuracy of population counting. Another approach is to employ the hash chain technique where the user pre-shares a secret seed with the rescue authority, and embeds a hash value as the sequence number into each location message, starting from the last value in the chain. The values of the hash chain are derived one by one from hashing the seed and each state value. This would render the sequence number random and unlinkable, and prevent the rescuers from illegally linking messages in normal network operations. The problem with this approach is that after the rescue authority approves the retrieval and delegates the secret seed to the rescuers for legitimate rescue purpose, the rescuers can potentially use this seed to link all past locations of a user.

2) Three-party collusion: Now consider the case where the rescue authority joins in the collusion when there is no disaster. Prior to the protocol execution in Fig. 2 (b), the user presents his/her identity to authenticate with the rescue authority (i.e., identity management authority), from whom the token is acquired. Since the rescuers can decrypt the location messages at will using the private key assigned by the corrupted key authority, the rescue authority can easily obtain $BSIG$ through the collusion with the rescuers and decrypt for $d \parallel S'(x)$. Due to the blindness property of blind signatures, the rescue authority is still not able to tie the token $S'(x)$ in $BSIG$ to any identity. Furthermore, since the user is not required to embed his/her identifying information into x as mentioned in Section 4.3, obtaining the result from $\frac{r_1 - r_1'}{r_2 - r_2'}$ leaks no information on the user identity to the colluding parties. Therefore, anonymity can be guaranteed in this three-party collusion scenario. The unlinkability guarantee, on the other hand, will be compromised in this three-party collusion scenario, since the colluding parties can now check and compare the results of $\frac{r_1 - r_1'}{r_2 - r_2'}$ as mentioned in the linkability attack. This attack cannot be eradicated if the three parties collude in normal network operations, because linkability will be desired in post-disaster rescue planning. However, we can alleviate the impact of such attack by proposing the following temporal redundancy mechanism.

If the same token is used for all location messages (i.e., multiple-spending), the colluders can easily link these messages. It is clear that different tokens should be included in the location messages from different geographic areas. This however, would render the location messages unlinkable even for the legitimate rescue planning, due to the fact that over-counting of the population may occur. To address this problem, we propose to exploit the temporal redundancy for location update messages. Recall that we use spatial redundancy to store "similar" location update messages that are originated in a geographic area. The temporal redundancy works in an analogous way. The idea is to place the same token $S'(x)$ in location messages that are "similar" in transmission times (but different in geographic areas), and different tokens otherwise. Here time is divided into intervals (e.g., 14:00-14:59, 15:00-15:59, etc.), and all (geographically different) location messages whose scheduled transmission times fall into a specific interval, will contain the same token. Note that the intervals can be overlapping or non-overlapping. In the former case, if the scheduled transmission time of a location update message falls within both intervals, either token can be carried. It is possible to transmit multiple geographically different location messages in one time interval, since users move frequently during the active period of a day. The design of temporal redundancy is in accordance with the reality that the server deletes obsolete data to make room for newly arrived data. Since we leverage the existing infrastructure for storage (i.e., no dedicated storage facilities), the data storage load at the server is expected to be heavy, and user data will arrive and be deleted frequently. Based on this observation, if the length of the time interval is properly set, location update messages that carry different tokens and are hence strictly unlinkable, will not likely to coexist in the network, which will not cause over-counting of trapped population in post-disaster rescue. The design of this interval involves tradeoff between countability and privacy, i.e., it should be long enough to ensure that all potentially retrieved messages of a user contain the same token, to eliminate over-counting in post-disaster rescue; and short enough so that few messages with a same token coexist in the network, to preserve location privacy in normal network operations.

Active outsider attacks. In the description of RescueMe so far, we have not mentioned the verifiability of the tokens but mainly used the tokens to eliminate over-counting, since we assumed that the users will not attempt to destruct the network by injecting ill-formed location messages to exhaust the rescuers' resource. However, verifying the validity of the tokens (i.e., they are indeed signed by the rescue authority after assuring the authenticity of a user) by the rescuers is imperative in the terrorism type of disasters or battlefield, where the terrorists or enemies may send their teams (i.e., active outsiders) to subvert the disaster rescue network. These outsiders can intercept the token which is not considered secret information and forge location update messages, or inject bogus location update messages attempting to consume the rescuers' resource in the rescue work. This is another key reason that we need to incorporate $d \parallel S'(x)$ into $BSIG$. In this case, the verifiability of the token can be easily achieved, leveraging d, r_1, r_2, and $S'(x)$, to assure that the token is generated by the rescue authority for *the authentic user*. Note that due to the properties of restrictive blind signature, it is not possible for outsiders to generate valid challenge/responses which can pass the verification, based on the intercepted token, without compromising the user to obtain related secrets. Using the schemes in [15], [16] for instance, the verification can be carried out by inputting the decrypted token $S'(x)$ and some domain public parameters of the identity management authority (cf. signature verification in [16]).

Access control. The role-based technique combined with ID-based encryption ensures the rescuers and rescue authority to access designated user information, and only for post-disaster rescue purposes. The encryption also prevents the storage server from accessing the location update messages. The symmetric key encrypted user identity $SKE(ID_{user})$ in the location update messages enables families and friends to locate a particular survivor (not any person), by cooperating with the rescuers.

B. Efficiency Analysis

Most pairing-based cryptosystems need to work in: 1) a subgroup of the elliptic curve $E(F_q)$ of sufficiently large prime order p, and 2) a sufficiently large finite field F_{q^k}, where q is the size of the field over which the curve is defined and k is the embedding degree. For current minimum levels of security, we require that $p > 2^{160}$ and $q^k > 2^{1024}$ [23] to ensure the hardness of the DLP in G_1 and G_2, the additive and multiplicative group, respectively. To improve the computation efficiency when working with $E(F_q)$, we tend to keep q small while maintaining the security with larger values of k. According to [23], a popular choice is to work with points in $E(F_q)$ where $q \approx 2^{170}$, and to have a curve with embedding degree $k = 6$ so that $q^k \approx 2^{1024}$. In the following analysis, we will use the parameter values given above, resulting in the elements in G_1 and G_2 to be roughly 171-bit (using point compression) and 1024-bit, respectively. We further assume SHA-1 [24] is used to compute the keyed-hash message authentication code (HMAC), which yields a 160-bit output. These parameter values will be used for the efficiency analysis in this section.

Storage. Major storage overhead in the proposed scheme is due to the redundancy storage of location update messages piggybacked in daily user messages. Each location update message $PEKS_{role}(location, d \oplus r_1, d \oplus r_2, IBE_{Rescue\_Authority}(d \parallel S'(x)), SKE(ID_{user}))$ takes roughly 0.46k bytes, assuming ID-based encryption [18], the scheme in [15], and AES block cipher are used to instantiate $PEKS$ [25], blind signature, and SKE, respectively. In addition, assume the ID-based domain parameters are set up as in [17]. This storage overhead is trivial to the dramatically advanced data storage technology, e.g., the magnetic storage devices have increased 100-fold in capacity while dropping in cost to \$0.50/GB. Storage at end users mainly includes storing the 342-bit ID-based public/privacy key pair (for authentication with the identity management authority), the 160-bit secret seed s for PRNG, and three 1024-bit elements used in computing $PEKS$ and IBE, totalling 0.45k bytes. Note that the token $S'(x)$ and related parameters d, r_1, r_2 are for the construction of location update messages and can be erased from the user device after the corresponding messages are sent out for storage.

Computation. We are most concerned with the computation load at the end users since the authorities and rescuers are either servers or computationally powerful devices. Pairing is the most expensive task among all operations including point multiplications and additions, hashing, etc. Tables 4.3 and 5.2 of [26] show that pairing operations count for all the high computation costs. Pairing is needed in the following events: users requesting tokens from the identity management authority (similar to the ticket issuance protocol in [17]), users computing $PEKS$ and IBE for location update messages. The pairing operations in the former event can either be computed once for all the token requests or be pre-computed and (temporarily) stored for each such request. During requesting tokens, users need authenticate with the identity management authority where digital signature [27] is used and only one pairing should be performed in real time for signature verification. In the latter event, three pairing operations are needed (two for $PEKS$ and one for IBE), all of which can be computed once and stored for future use contributing to the three 1024-bit storage elements mentioned above. Assuming Tate pairing is used, it is shown in [28] that the time taken for computing a Tate pairing is 20ms, 23ms, and 26ms, in the underlying base field of F_p (where $|p| = 512$-bit), $F_{2^{271}}$, and $F_{3^{97}}$, respectively. The first two fields have similar levels of security to 1024-bit RSA while the last field has effective 922-bit security. Recent progress [29] shows that the computation time of Tate pairing on elliptic curves in characteristic 2 and 3 has been significantly improved, rendering pairing-based cryptosystems more realistic in security applications. We conclude that the real-time computation load for the end users is very acceptable.

VII. Conclusion and Future Work

In this paper, we propose a location-based secure and dependable disaster rescue network. By solving the challenging problem of exploiting the stored location information for post-disaster rescue, and at the same time preserving location privacy in normal network operations, RescueMe offers a functional, secure, and sound networking solution for disaster rescue, which is likely to gain user acceptance and requires little deployment effort. We plan to carry out simulations for different connectivity scenarios and different parameter settings (e.g., time interval in the temporal redundancy mechanism), by incorporating various locating/positioning techniques, and different methods for dividing geographic areas in redundancy storage, to evaluate the performances of the proposed networking solution.

Acknowledgment

This work was partially supported by the U.S. National Science Foundation under grants CNS-0916391 and CNS-0716450, and the China 111 Project under Grant B08038. The work of Zhu was also partially supported by the China Fundamental Research Funds for the Central Universities under Grant No. JY10000901021.

References

[1] K. Fall, “A delay-tolerant network architecture for challenged internets,” *Proc. ACM SIGCOMM*, 2003.

[2] S. Jain, K. Fall, and R. Patra, “Routing in a delay tolerant network,” *Proc. ACM SIGCOMM*, 2004.

[3] E. Cayirci and T. Coplu, “SENDROM: Sensor networks for disaster relief operations management,” *Wireless Networks*, vol. 13, no. 3, May 2007.

[4] S. Suman and M. Mitsuji, “A framework for disaster management system and wsn protocol for rescue operation,” *in Proc. IEEE TENCON*, Nov. 2007.

[5] N. Ansari, C. Zhang, R. Rojas-Cessa, P. Sakarindr, E. S. H. Hou, and S. De, “Networking for critical conditions,” *IEEE Wireless Commun.*, vol. 15, no. 2, pp. 73–81, Apr. 2008.

[6] T. Fujiwara, H. Makie, and T. Watanabe, “A framework for data collection system with sensor networks in disaster circumstances,” *Int. Workshop on Ad Hoc Networks*, June 2004.

[7] G. Ranjan, A. Kumar, G. Rammurthy, and M.B. Srinivas, “A natural disasters management system based on location aware distributed sensor networks,” *in Proc. IEEE MASS*, Nov. 2005.

[8] Y. Chu, X. Huang, and A. Ganz, "WISTA: a wireless transmission system for disaster patient care," *in Proc. 2nd Int. Conf. on Broadband Networks*, Oct. 2005.

[9] N. Pogkas, G. E. Karastergios, C. P. Antonopoulos, S. Koubias, and G. Papadopoulos, "Architecture design and implementation of an ad-hoc network for disaster relief operations," *IEEE Trans. Ind. Informat.*, vol. 3, no. 7, Feb. 2007.

[10] F. Hoeksema, M. Heskamp, R. Schiphorst, and K. Slump, "A node architecture for disaster relief networking," *DySPAN 2005*, Nov. 2005.

[11] N. Ahmed, K. Jamshaid, and O. Khan, "Safire: A self-organizing architecture for information exchange between first responders," *2nd IEEE Workshop on Networking Technologies for Software Define Radio Networks*, June 2007.

[12] A. Shamir, "How to share a secret," *Comm. of the ACM*, vol. 22, pp. 612–613, 1979.

[13] J.-P. Kaps amd G. Gaubatz and B. Sunar, "Cryptography on a speck of dust," *computer*, vol. 40, no. 2, Feb. 2007.

[14] S. Brands, "Untraceable off-line cash in wallets with observers," *in Proc. CRYPTO'93, 13th Annual Int'l Cryptology Conf. on Advances in Cyptology*, pp. 302–318, Aug. 1993.

[15] X. Chen, F. Zhang, Y. Mu, and W. Susilo, "Efficient provably secure restrictive partially blind signatures from bilinear pairings," *in Proc. 10th Conf. on Financial Cryptography and Data Security, FC 2006*, pp. 251–265, Feb. 2006.

[16] X. Chen, F. Zhang, and S. Liu, "ID-based restrictive partially blind signatures and applications," *Journal of Systems and Software*, vol. 80, no. 2, pp. 164–171, Feb. 2007.

[17] J. Sun, C. Zhang, and Y. Fang, "A security architecture achieving anonymity and traceability in wireless mesh networks," *IEEE Conf. on Computer Communications (INFOCOM)*, pp. 1687–1695, Apr. 2008.

[18] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing. extended abstract in CRYPTO 2001," *SIAM J. of Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[19] R. Dingledine, "Tor: An anonymous internet communication system," *Workshop on Vanishing Anonymity, The 15th Conf. on Computers, Freedom, and Privacy*, Apr. 2005.

[20] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," *in Proc. USENIX Security Symposium*, pp. 303–320, Aug. 2004.

[21] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, "Anonymous connections and onion routing," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 4, pp. 482–494, May 1998.

[22] M. Raya and J-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security, Special Issue on Security of Ad Hoc and Sensor Networks*, vol. 15, no. 1, pp. 39–68, 2007.

[23] S. D. Galbraith, "Pairings. In I. F. Blake, G. Seroussi, and N. P. Smart, editors," *Chapter 9 of Advances in Elliptic Curve Cryptography*, pp. 183–213, 2005.

[24] NIST, *Digital Hash Standard*, Federal Information Processing Standards (FIPS) Publication 180-1, Apr. 1995.

[25] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," *in EUROCRYPT 2004, LNCS 3027. Springer*, 2004.

[26] H. W. Lim, *On the Application of Identity-Based Cryptography in Grid Security*, Ph.D thesis, University of London, 2006.

[27] F. Hess, *Efficient identity-based signature schemes based on pairings*, SAC 2002, LNCS 2595, pp. 310-324, Springer-Verlag, 2002.

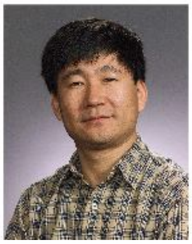
[28] P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," *CRYPTO 2002, Springer-Verlag, LNCS 2442*, pp. 354–368, 2002.

[29] P. S. L. M. Barreto, S. D. Galbraith, C. ÓhÉigeartaigh, and M. Scott, *Efficient pairing computation on supersingular abelian varieties*, Cryptology ePrint Archive, Report 2004/375, available at http://eprint.iacr.org/2004/375.pdf, Sept. 2005.

Jinyuan Sun received the BSc degree in computer information systems from Beijing Information Technology Institute, China, in 2003, the MASc degree in computer networks from Ryerson University, Canada, in 2005, and the PhD degree in electrical and computer engineering from the University of Florida, in 2010. She was a Network Test Developer at RuggedCom Inc., Ontario, Canada, 2005-2006. She has been an assistant professor in the Department of Electrical Engineering and Computer Science at University of Tennessee Knoxville since August 2010. Her research interests include the security protocol and architecture design of wireless networks.

Xiaoyan Zhu received her BE degree in Information Engineering from Xidian University, Xian, China, in July 2000, and her ME degree in Information and Communications Engineering from Xidian University, Xian, China, in March 2004. She is now working towards her Ph.D. degree at Xidian University. Her research interests include wireless security and network coding.

Chi Zhang received the B.E. and M.E. degrees in Electrical Engineering from Huazhong University of Science and Technology, Wuhan, China, in July 1999 and January 2002, respectively. Since September 2004, he has been working towards the Ph.D. degree in the Department of Electrical and Computer Engineering at the University of Florida, Gainesville, Florida, USA. His research interests are network and distributed system security, wireless networking, and mobile computing, with emphasis on mobile ad hoc networks, wireless sensor networks, wireless mesh networks, and heterogeneous wired/wireless networks.



Yuguang Fang received the PhD degree in systems, control and industrial engineering from Case Western Reserve University in January 1994 and the PhD degree in electrical engineer-ing from Boston University in May 1997. He held a postdoctoral position in the Department of Electrical and Computer Engineering at Boston University from June 1994 to August 1995. From June 1997 to July 1998, he was a visiting assistant professor in the Department of Elec-trical Engineering at the University of Texas at Dallas. From July 1998 to May 2000, he was an assistant professor in the Department of Electrical and Computer Engineering at the New Jersey Institute of Technology. In May 2000, he joined the Department of Electrical and Computer Engineering at the University of Florida, Gainesville, where he received early promotion to associate professor with tenure in August 2003 and to full professor in August 2005. He holds a University of Florida Research Foundation (UFRF) Professorship from 2006 to 2009. His research interests span many areas, including wireless networks, mobile computing, mobile communications, wireless security, automatic con-trol, and neural networks. He has published more than 100 papers in refereed professional journals and more than 100 papers in refereed professional conferences. He received the US National Science Foundation Faculty Early Career Award in 2001 and the Office of Naval Research Young Investigator Award in 2002. He was the recipient of the Best Paper Award at the IEEE International Conference on Network Protocols (ICNP) in 2006 and the recipient of the IEEE TCGN Best Paper Award at the IEEE High-Speed Networks Symposium, IEEE Globecom in 2002. Dr. Fang has actively engaged in many professional activities. He is an editor for several journals, including the IEEE Transactions on Communications, the IEEE Transactions on Wireless Communications, the IEEE Transactions on Mobile Computing, ACM Wireless Networks,and the Journal of Computer Science and Technology, and a technical editor for IEEE Wireless Communications Magazine. He was also an editor of the IEEE Journal on Selected Areas in Communications: Wireless Communications Series, an area editor of the ACM Mobile Computing and Communications Review, an editor of Wireless Communications and Mobile Computing, and a feature editor for Scanning the Literature in IEEE Personal Communications. He also served on the Technical Program Committee of many professional conferences, such as ACM MobiCom 02 (committee cochair for Student Travel Award), MobiCom 01, IEEE INFOCOM 08, IEEE INFOCOM 07, INFOCOM 06, INFOCOM 05 (vice-chair for technical program committee), INFOCOM 04, INFOCOM 03, INFOCOM 00, INFOCOM 98, IEEE WCNC 04, WCNC 02, WCNC 00 (technical program vice-chair), WCNC 99, IEEE Globecom 04 (symposium cochair), Globecom 02, and the International Conference on Computer Communications and Networking (IC3N, technical program vice-chair). He is a fellow of the IEEE.