Neighborhood Prediction based Decentralized Key Management for Mobile Wireless Networks

Xiuyuan Zheng, Yingying Chen, Hui Wang, Hongbo Liu, Ruilin Liu

Abstract—The wireless data collected in mobile environments provides tremendous opportunities to build new applications in various domains such as Vehicular Ad Hoc Networks and mobile social networks. One of the biggest challenges is how to store these data. Storing the data decentralized in wireless devices is an attractive approach because of its major advantages over centralized ones. In this work, to facilitate effective access control of the wireless data in distributed data storage, we propose a fully decentralized key management framework by utilizing a cryptography-based secret sharing method. The secret sharing method splits the keys into multiple shares and distributes them to multiple nodes, which brings the challenge that due to node mobility, these key shares may not be available in the neighborhood when they are needed for key reconstruction. To address this challenge arising from mobile environments, we propose the Transitive Prediction(TRAP) protocol that distributes key shares among devices that are traveling together, and we further theoretically analyze the incurred communication overhead. We derive a theoretical analysis of the robustness and security of our approach. Furthermore, inside TRAP, we develop three key distribution schemes that utilize the correlation relationship embedded among devices that are traveling together. Our key distribution schemes maximize the chance of successful key reconstruction and minimize the communication overhead. Our extensive simulation results, by using the generated data from city environment and NS-2 simulator, demonstrate that our key distribution schemes are highly effective, and thus provide strong evidence of the feasibility of applying our approach to support distributed data storage in wireless networks.

Index Terms—decentralized key management; mobile wireless networks; neighborhood prediction; secret sharing; distributed storage.

1 Introduction

THE rapid advancement of wireless technologies has led to a future where wireless networks will be pervasively deployed. As a matter of fact, with the increasing programmability of wireless devices and the continuously reducing cost of communication radios, mobile wireless networks are becoming a part of our social life. For instance, vehicles are equipped with wireless communication devices to form Vehicular Ad Hoc Networks (VANETs), in which vehicles have the sensing capability to collect data regarding to road conditions and traffic scenarios [29]. Another example is that data collection and real-time multimedia blogs [3], [21] enabled by various sensing capabilities on mobile phones, such as cameras, GPS, and accelerometers, provide geo-related information that supports effective mobile social collaboration. Thus, the wireless data collected in the mobile environments provide abundant information to build pervasive applications in our social life.

- *Xiuyuan Zheng, Yingying Chen, Hongbo Liu are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, 07030.*
  E-mail: {xzheng1, yingying.chen, hliu3}@stevens.edu
- *Hui Wang and Ruilin Liu are with Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ, 07030.*
  E-mail: {hui.wang, rliu3}@stevens.edu

Most of the existing work [27] requires the data to be sent back to centralized storage nodes continuously and only considers stable network topology. However, this may incur high communication overhead and excessive energy consumption among wireless devices by continuously forwarding the data to storage nodes. To address these issues, distributed data storage [13], [14], [23], [26], [27] in wireless networks has attracted much attention. The distributed data storage has major advantages over centralized approaches: storing the data on the collected wireless device or in-network storage nodes decreases the need of constant data forwarding back to centralized places, which largely reduces the communication in the network and the energy consumption on individual devices, and consequently eliminates the existence of centralized storage and enables efficient and resilient data access. Furthermore, as wireless networks become more pervasive, new-generation wireless devices with significant memory and powerful processing capabilities are available (i.e., smart phone and laptops), making the deployment of distributed data storage not only feasible but also practical. In this work, the collected data will be stored in each *collector node*, i.e., the mobile device that collects the data.

In many cases, the data collected by mobile wireless networks contain sensitive information. For instance, an adversary can derive the trajectories of vehicular drivers to infer their social behaviors, or analyze the video clips embedded in the multimedia blogs to derive users' lifestyles. Such vulnerabilities are significantly threatening the deployment of applications that utilize the large-scale data sets collected by wireless mobile networks. Therefore, while the wireless data provides abundant opportunities for developing new applications, it could also be dangerous if not handled appropriately and misused by adversaries. Thus, secure data storage must be achieved before widespread adoption of distributed data storage. One of the main challenges in utilizing the distributed wireless data is to develop effective mechanisms that control the access of data so that the right information is shared with the right party at the right time.

Traditional encryption-based access control approaches employ an individual or a group of centralized certification authorities for key management [1], [32]. However, these approaches face the difficulty of scaling with the increasing size and the mobility of devices in wireless networks. Even worse, the centralized authorities in these approaches can become a single point of failure. In this paper, we propose a *fully decentralized* key management framework by utilizing the cryptography-based secret sharing method. The secret sharing approach has been very useful in developing decentralized security protocols [17], [20]. In our decentralized framework, the data is encrypted and the decryption key is divided and shared among mobile devices in the network.

However, the mobility of devices introduces environmental dynamics and makes it hard to reconstruct the key. To cope with mobility, we propose to distribute the key shares among devices that travel together with the collector node through neighborhood prediction. Indeed, in our daily life, people usually travel together to common destinations or areas, e.g., commuting along the same train lines or visiting a museum together. This co-movement phenomenon makes our neighborhood prediction feasible. We further develop the *Transitive Prediction (TRAP)* protocol that helps to maximize the chances of successful key share reconstruction and minimize the communication overhead, and in the meanwhile avoiding the degradation of the security guarantee of data access. Further, we theoretically analyze the communication overhead of key distribution and key reconstruction involved in the TRAP protocol.

Inside *TRAP*, we design three key distribution schemes. These three key distribution schemes can be classified into two categories, the one that does not respect the relationships between moving patterns of different devices, and the one that does. For the first type, we develop a scheme named *random selection*, while for the second type, we develop two schemes, namely *association-probability-based*, and *association-rule-based*. In addition, we derive the theoretical analysis of the robustness and security of our mechanism, and provide discussions on our analytical results.

To evaluate the feasibility of our framework, we use simulated mobile wireless networks in a city environment [7] with different moving speeds: walking speed and vehicular traveling speed. Furthermore, we conduct simulations through NS-2 simulator with RPGM mobility model [16] to validate our analysis on communication overhead, and evaluate the effectiveness of key distribution schemes. Our results show that our key distribution schemes are both effective and efficient to achieve successful key reconstruction in mobile and decentralized environments. These results provide strong evidence of the feasibility of applying our decentralized key management scheme in mobile wireless networks.

The remainder of the paper is organized as follows. We first put our work into the broader context of the current research in Section 2. We then present our decentralized key management framework for mobile wireless networks, and analyze the incurred communication overhead in Section 3. We provide the robustness and security analysis of our approach in Section 4. In Section 5, we describe our key distribution schemes for efficient key reconstruction. We present our simulation methodology and results using various data sets generated from simulated mobile wireless networks and NS-2 simulator in Section 6. Finally, we conclude our work in Section 7.

2 Related Work

Key management is a key component of encryption-based access control system. Recent work has focused on eliminating the need of centralized authentication management in wireless networks. In particular, to address mobility, [4], [30] make use of privileged side channels when mobile users are in the vicinity of each other. The secure side channel is used to set up security associations between nodes by exchanging cryptographic materials. However, the availability of the privileged side channels is not guaranteed.

On the other hand, the secret sharing method has been actively studied in the field of cryptography [8]–[12], [15], [25], [28], [32]. The advantage of using the secret sharing method is that the possibility of a single point of failure is significantly reduced. [25], [32] develop threshold secret sharing. [28] proposes the concept of proactive secret sharing by renewing the key shares periodically. [10] employs verifiable secret sharing to avoid invalid shares provided by any shareholder. Moreover, the secret sharing method has been applied in mobile ad hoc networks [17], [20], [32]. [32] proposes a distributed public-key management scheme based on threshold secret sharing in which the certificate authority (CA) services are divided into a certain number of specialized servers. The drawback is that it assumes some nodes must behave as servers. When moving towards fully distributed infrastructure, a decentralized authentication protocol is developed to distribute the authentication of a CA by utilizing secret sharing [20]. However, it does not consider the mobility of nodes, and thus making it inapplicable to mobile environments.

The work that is most closely related to ours is [17]. By taking into the consideration of mobility, [17] introduces a redundancy-based key distribution scheme in secret sharing to achieve a decentralized CA. Basically, more than one key share are distributed to each node in order to increase the probability of successful key reconstruction in mobile networks. However, the security level of the system can be degraded due to having multiple redundant key shares on nodes. Our work is novel in that our proposed decentralized key management framework employing secret sharing maintains the security guarantee of the data access through neighborhood prediction and distributes key shares only to those nodes that travel together.

3 Decentralized Key Management Framework

We present the framework of our decentralized key management approach in this section. We first describe the network model and adversary model. We then present our approach of decentralized key management protocol. Next, we analyze the communication overhead involved in our proposed protocol.

3.1 Network model

We consider mobile wireless networks, which contain a large number of wireless devices (e.g., mobile phones, laptops, or on board sensing units on vehicles). Each device has a unique ID and may perform different functionalities in the network. For subsequent discussions, we use the term device and node interchangeably. Nodes may freely roam in the network, and the number of nodes in a network may be dynamically changing due to its capability of mobility, i.e., mobile nodes may join, leave, or fail over time. In this work, we target our solutions to a category of mobile wireless networks with the following characteristics.

Node Placement. We make the assumption that the wireless nodes are randomly deployed in the network, with the node distribution following a homogeneous Poisson point process with a density of ρ nodes per unit area [5], [24]. This assumption is reasonable and has been widely used in analyzing multi-hop mobile wireless networks [6], [18], [22].

Mobility. Each node moves randomly or follows some patterns in a large well-defined area. However, we assume that the nodes are not aware of their moving patterns, if there is any. There are no predefined trajectories for each node. We assume there exists a *co-movement* pattern within nodes, i.e., group of nodes may travel together to common destinations. For example, a group of tourists in New York City may travel to visit the Metropolitan Museum together and each of them can use their mobile phones to take pictures, shoot videos, and write multimedia blogs on the way.

Neighbor-Aware. Each node has a communication range and can communicate only with nodes within its transmission range. We call the nodes in the transmission range the *neighbors*. Mobility of nodes may result in the change of the neighborhood. However, we assume that for every node, it has a comparatively stable neighborhood within a period of time.

Location-Aware. Each node knows their physical locations at all time points during moving. This is a reasonable assumption as most of wireless devices (e.g., mobile phones or vehicles) are equipped with GPS or some other approximate but less burdensome localization algorithms [19]. In many cases the location of the collected data is important. For example, knowing that a traffic accident occurred, which requires to inform the neighboring nodes, but without knowing where it occurred is useless.

Distributed Data Storage. Each node stores the data it has collected. The data will be stored within the network at each collector node (e.g., mobile phones or vehicles) unless it is required to be sent to a centralized storage space for backup. By uploading data in a lazy fashion (i.e., on-demand only), distributed data storage enables real-time query evaluation and avoids frequent data transfer from the wireless devices to the centralized storage, and consequently reduces battery power consumption and decreases the communication overhead of the network.

3.2 Adversary Model

In this work, we consider adversaries that can compromise any wireless devices to obtain the key shares. Once a node is compromised, an adversary can get the key share stored on the node if any, however, it cannot decrypt the data stored on the compromised node. An adversary needs to compromise up to m nodes in order to reconstruct the key to decrypt the data on a collector node. Further, once a node is compromised, an adversary may generate fake data and then distribute key shares of the fake data in the network. However, this behavior will not affect the secure access of the legitimate data cached in the network. Thus, it is not the focus of our work.

3.3 Distributed Key Management Model

3.3.1 Node Authentication

There has been sufficient work [17], [20], [32] that we can employ to perform node authentication. [32] proposed a partially distributed certificate authority scheme that supports authority services to be shared by multiple servers. [20] proposed a distributed cryptography-based authentication solution that distributes a certificate key to each node. [17] extended [20] by providing a redundancy-based solution for node authentication. Thus, we can adopt these existing works for node authentication in our network and mainly focus on studying decentralized key management for secure data access. In our work, whenever a node enters the network, it has to pass the authentication procedure. When a node in the network tries to access data, the node needs to collect m key pieces. Thus, an attacker node has to compromise up to m nodes, which means that it has to succeed for m trials to hack the system with complex overhead. This highly increases the security level of our system compared with the system that uses a centralized authority for data access, so that the attacker node only has to hack one node, that is, the centralized authority node.

3.3.2 Secret Sharing Based Key Management

To prevent the misuse of the data and protect the privacy of mobile users, the data is encrypted in our framework. Further, we propose to use the secret sharing scheme to achieve decentralized key management in dynamic wireless environments.

Secret sharing, also named threshold secret sharing, is originated from [25]. Specifically, in a (m, n) secret sharing scheme, a secret is distributed among n participants; only by collecting $m(1 < m \leq n)$ secret shares can re-construct the secret. The decision of values for m and n controls the strength of the system.

Key Distribution. More formally, in mobile wireless networks, a data decryption key S is shared among n devices. To share the S among the n devices, $\{c_1, c_2, \ldots, c_n\}$, we pick a polynomial of order $m-1$: $f(x) = S + a_1x + a_2x^2 + \cdots + a_{m-1}x^{m-1}$. Then the key share S_i to be distributed to device i is $S_i = f(c_i) mod\ p$, where p is a big prime number, and c_i denotes the ith device among n.

We develop the secret sharing method in a fully distributed manner: Each collector node acts as the dealer node as defined in the secret sharing scheme [25] and is responsible to distribute the decryption key of its own data. Furthermore, since each collector node can encrypt its data at different time periods, there can be multiple keys associated with each node in our network. Thus, in order to identify the key shares that belong to the same key, the collector node will generate a unique key ID to append to each key share. The unique key ID will help to identify the key shares that belong to the same decryption key. The collector node will destruct the decryption key after it distributed the key shares.

Key Reconstruction. At a later time, the secret key S can be reconstructed by using Lagrange interpolation $S = f(0) = \sum_{i=1}^{m} S_{c_i} * l_{c_i}(x)(mod\ p)$, where p is a big prime number, and $l_{c_i}(x)$ is the lagrange coefficient of the ith device and is defined as $l_i(x) = \prod_{j=1, j\neq i}^{m} \frac{c_j}{c_j - c_i}$. Any subsets of m key shares could reconstruct the decryption key and each wireless device is unaware of others' shares. Further, only the authorized node by the authentication protocol, e.g., [20], which owns the certificate key, can reconstruct secret key S. Note that the collector node is not responsible for key reconstruction; it collects data not key pieces. **Yingying, I moved the clarification of responsibility of the collector nodes here instead of in Section 3.1. Please feel free to move it.**

Key Updating. Given sufficiently long time, an adversary could compromise m nodes and reconstruct the decryption key of the data. To make our secret sharing based key management more robust, the key shares will be updated periodically. We apply proactive secret sharing [28] in which the key shares will be expired after a specified time period controlled by the collector node. The collector node will re-distribute a set of key shares once the key shares in the previous distribution have expired. The new key share $f_{new}(x)$ can be generated as $f_{new}(x) = S + (a_1 + b_1)x + \cdots + (a_{m-1} + b_{m-1})x^{m-1}(mod\ p)$. Periodically, the collector node will distribute the n newly generated key shares to n wireless devices. The old keys are expired and thus are discarded.

3.3.3 Handling Mobility via Neighborhood Prediction

In a mobile wireless network, the devices carrying key shares may move farther away, causing much communication overhead during key reconstruction and even reconstruction failure (e.g., unreachable devices). Thus, it is desirable to distribute key shares to devices that are moving together with the collector node, and consequently increasing the success rate of key reconstruction in dynamic network environments and reducing the communication overhead and energy consumption during the reconstruction process. However, this brings in a new challenge of how to determine the devices that are traveling together with the collector node. To address this issue, we propose to use neighborhood prediction. In particular, we developed an array of key distribution schemes, which explore correlations embedded in the moving patterns of wireless devices, to predict devices that are traveling together for efficient key distribution. The detailed schemes will be presented in Section 5. During the key distribution phase, the collector node utilizes these schemes to pick the top n wireless devices that are most likely traveling together with it, and distributes the n key shares to these devices.

Further, as stated in our network model each mobile wireless device only keeps the information of its 1-hop neighbors (i.e., devices within its transmission range). During the key distribution phase, it is possible that there are not enough devices within the 1-hop range to share the key, i.e., the devices within the 1-hop

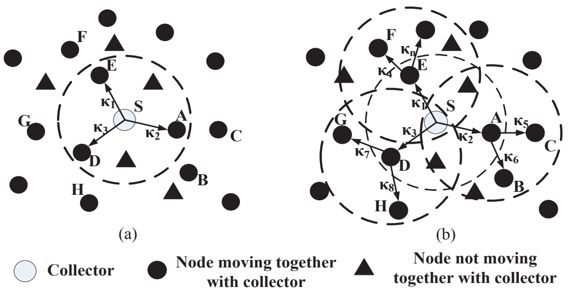

F
E
S
G
A
C
D
H
B
(a)
(b)
Collector
Node moving together with collector
Node not moving together with collector

Fig. 1. Illustration of TRAP in a 2-hop scenario

range of the collector node are less than n. To address this problem, there are two possible solutions:

Solution 1: The collector could request its 1-hop neighbors to send the information of their respective 1-hop neighbors back to it as candidates. Under the scenario that the returned number of candidates is still less than n, the collector will make iterative requests to the neighbors of neighbors to collect more candidate devices, until it collects at least n candidates. Then it will run the key distribution scheme on these candidates and choose the top n devices from the results as the key share holders.

Solution 2: The idea behind the second solution is that the co-movement is *transitive* in practice. For instance, if a mobile user A is traveling together with user B, meanwhile B is traveling together with C, it is highly likely that A is also traveling together with C. Thus, the collector node can utilize this property and distribute the prediction responsibility of key distribution to its neighbors for further prediction of the devices traveling together when there are less than n devices within the 1-hop neighborhood for key share distribution. The prediction of key distribution (i.e, the key distribution scheme) can be successively invoked by the neighbors of the neighbors until enough candidates are found. The predicted results at each neighboring node during each round of invocation will be sent back to the collector node as candidates for choosing the top n devices.

Transitive Prediction (TRAP) Protocol. We note that Solution 1 may incur high computational cost and expensive energy consumption at the collector node. Thus, in this work, we take Solution 2 and develop a fully distributed prediction protocol called *Transitive Prediction (TRAP)* that builds on top of our key distribution schemes. We utilize a layered approach (i.e., we call 1-hop neighbors of a node as one layer) to successively find enough devices that are traveling together with the collector node for resilient key distribution in multi-hop mobile environments. In *TRAP*, the *k-hop neighbors* of the collector node is defined as the 1-hop neighbors of the $(k-1)$-hop neighbors of the collector node with $k > 1$. Figure 1 depicts TRAP of finding n traveling together devices with the collector node in a 2-hop scenario for key distribution.

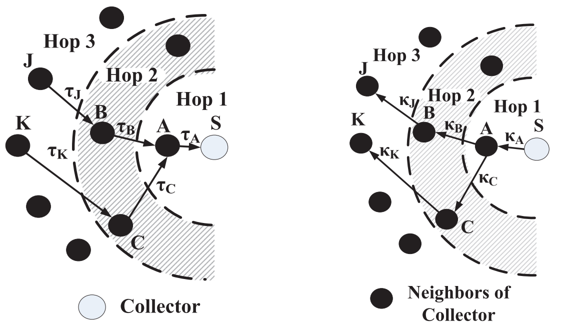
At every round of TRAP, each involved neighboring node will run the key distribution scheme to predict top x devices from its 1-hop neighbors and send the prediction results as candidates back to the collector node. To ensure returning the sufficient number of candidates, we choose $x = n$ in TRAP. The collector node will then choose the top n devices from the returned candidates based on the prediction criteria (e.g., the association rule in *Association-rule-based* scheme in Section 5) in our key distribution schemes to share the key. Thus, in TRAP the computation of successive prediction is distributed at the neighbors that are traveling together, and consequently the computational cost and energy consumption at the collector nodes is significantly reduced.

3.4 Communication Overhead

Key Distribution. We first examine the communication overhead of the key distribution phase in the TRAP protocol in terms of number of transmitted packets. The overhead during this phase consists of two parts: (1) the overhead incurred by a collector node collecting the trajectory information from its neighborhood, and (2) the overhead incurred by the collector node distributing n key pieces to its neighbors. Next, we discuss how to measure these two types of overhead.

Let L be the length of a transmission packet. We assume that one record of trajectory at one time point consists of a pair of (x, y) coordinates and its corresponding time stamp. Let R be the size of one trajectory record (e.g., $R = 12$ bytes when the (x, y) coordinates and the time stamp are of float type). Assume each device records its trajectories every t time units. Then the trajectory data of the time window of t' units can be stored in $\lceil \frac{t'}{t} \cdot \frac{R}{L} \rceil$ packets. Assume each collector node has N neighbors at average within its transmission range r. Then for a particular collector node, it has $N \cdot (k^2 - (k-1)^2) = N \cdot (2k-1)$ neighbors at average in the circular area between the k-hop and $(k-1)$-hop transmission.

Figure 2 illustrates the packet transmission of collecting the trajectories from a collector node S's neighborhood and distributing key pieces to its neighborhood in a 3-hop scenario. The shaded circular area shows the neighbors of S when $k = 2$ hops. The τ in Figure 2 (a) indicates packets containing trajectory information from k-hop neighborhood returning back to S, while κ in Figure 2 (b) indicates packets containing key pieces distributed from S to its neighborhood. Figure 2 (a) depicts the scenario that the number of nodes in 1-hop (node A) and 2-hop neighborhood (node B and C) are less than the n value of a (m, n) secret sharing scheme, thus the 3-hop neighborhood (node J and K) also need to send trajectory information back to S. Figure 2 (b) shows the scenario that multiple key shares are distributed by S to its multi-hop neighborhood.



(a) Collection of trajectory packets (b) Distribution of key shares

Fig. 2. Illustration of packets transmission in a 3-hop scenario.

First, we discuss the overhead incurred by a collector node collecting the trajectory information from its neighborhood. The collector first sends request messages to its neighborhood asking for trajectory information. We assume the request message can fit into one packet. The overhead H_{req} incurred by transmitting packets of request messages can be represented by the number of request packets transmitted within h-hop:

$$\begin{aligned} H_{req}(h) &= \sum_{k=1}^{h} k(2k-1)\cdot N \\ &= (h^3 - \frac{h^2}{2} + \frac{h}{2})N. \end{aligned} \tag{1}$$

Then, the packets containing trajectory information from the k-hop nodes need to be transmitted via its $(k-1)$-hop neighbors back to the collector node, which means the $N \cdot (2k-1)$ nodes, in the circular area between k-hop and $(k-1)$-hop, will transmit its trajectory information back to the collector node. Thus, the overhead H_{traj} incurred by sending the trajectories information can be bounded by the number of transmitted packets containing the trajectory collection:

$$\begin{aligned} H_{traj}(h) &= \sum_{k=1}^{h} k(2k-1)N \cdot \lceil \frac{t'}{t} \cdot \frac{R}{L} \rceil \\ &= (h^3 - \frac{h^2}{2} + \frac{h}{2})N \cdot \lceil \frac{t'}{t} \cdot \frac{R}{L} \rceil. \end{aligned} \tag{2}$$

Next, we discuss the overhead incurred by the collector node distributing n key pieces to its neighbors. The collector node applies our key distribution schemes to predict the top n $(n < N)$ neighbors as key distributees within the network. Recall that each key piece can fit into one packet. Thus the collector node sends n packets, each containing a key piece. The overhead H_{key} of number of packets transmitted for distributing key pieces is,

$$H_{key}(h) = h \cdot n. \tag{3}$$

Therefore, the total number of packets transmitted for key distribution within h-hop transmission is

$$\begin{aligned} H_{dist}(h) &= H_{req}(h) + H_{traj}(h) + H_{key}(h) \\ &= (h^3 - \frac{h^2}{2} + \frac{h}{2})N \cdot (\lceil \frac{t'}{t} \cdot \frac{R}{L} \rceil + 1) + h \cdot n. \end{aligned} \tag{4}$$

From Equation 4, we found that the communication overhead of the key distribution is cubically proportional to the number of hops. When $h = 1$, it is the one-hop communication overhead scenario.

Key Reconstruction. We next study the communication overhead during the key reconstruction phase, which is made of two parts: (1) the overhead by a legitimate user sending the key piece request messages to its neighborhood, and (2) the overhead by transferring key pieces back from nodes containing the information in the neighborhood. In a (m, n) secret sharing scheme, in order to reconstruct the key, the user needs to collect at least m key pieces, where $m < n$.

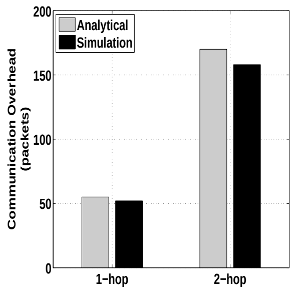
We first discuss the overhead by a legitimate user sending the key piece request messages to its neighborhood. Within one-hop transmission, the user first sends the request messages to its N neighbors. We assume the request message can fit into one packet. The key piece holder, who is within the communication range, sends the key pieces to the user upon receiving the request message. Therefore, the transmitted packets are $N + l$, where l is the collected key pieces in total. If $m \leq l \leq n$, the key reconstruction is successful; if not, the collector may send requests to its multi-hop neighbors. For the k-hop transmission, it is similar to the key distribution phase that the packets need to be transmitted via the $(k-1)$-hop neighbors. Thus, the overhead H_{recon} in terms of the number of packets transmitted during key reconstruction is,

$$\begin{aligned} H_{recon}(h) &= \sum_{k=1}^{h} k(2k-1)\cdot N + h \cdot l \\ &= (h^3 - \frac{h^2}{2} + \frac{h}{2}) \cdot N + h \cdot l. \end{aligned} \tag{5}$$

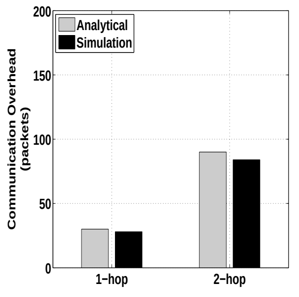
From Equation 5, we found that the communication overhead during the key reconstruction is also cubically proportional to the number of hops, which has the similar trend to the key distribution phase.

Simulation Results. To verify our analysis of communication overhead, we conducted simulation through the network simulator NS-2. The mobility model is applied by using Random way Point Group Mobility (RPGM) [16]. We used the IEEE 802.11 b with RTS/CTS with packet size setting to 512 bytes. The average number of packets for key distribution and reconstruction is measured over 50 simulation runs.

We examined both 1-hop and 2-hop scenarios. In the 1-hop scenario, we set the average number of neighbors $N = 20$, while in the 2-hop scenario the



(a) Key Distribution



(b) Key Reconstruction

Fig. 3. Communication overhead comparison between analytical and simulated results with $(6, 15)$ scheme, and $N = 20$ in the 1-hop scenario and $N = 10$ in the 2-hop scenario.

average number of neighbors is set to $N = 10$. Figure 3 presents the comparison of communication overhead in terms of number of packets between the analytical and simulated results. We found that our analytical and simulated results are quantitatively the same. The small difference is caused by the varying number of neighbors N due to the mobile environments in our simulation, while in our theoretical analysis the number of neighbors is fixed.

4 Robustness and Security Analysis

In this section, we formally analyze the robustness and security of our TRAP protocol in mobile wireless networks.

4.1 Robustness Analysis

The (m, n) secret sharing scheme splits the decryption key into n shares and distributes the n shares to n devices. However, due to the mobility of the network, it is possible that these key shares may not be accompanied together while time goes. Thus in the following, we analyze the robustness of the protocol via the probability that legitimate users can successfully reconstruct the key.

One-hop scenario. This scenario considers the case that there are sufficient m key shares available in the 1-hop neighborhood of the node for key reconstruction. Assume that each node has a transmission range r; thus it covers an area $A = \pi r^2$. Since the number of nodes N in the area A follows a Poisson distribution, the probability that a node has i nodes in its 1-hop neighborhood is $Pr(N = i)= \frac{\gamma^i}{i!}e^{-\gamma}$, where the expected node degree $\gamma = \rho\pi r^2$. Further, we define p_1, the percentage of nodes in the 1-hop neighborhood of the collector node that hold key shares, where $0 < p_1 < 1$, and we assume that there exists at least one key share within its neighborhood. Let i_1 be the total number of nodes in the 1-hop neighborhood. Since as each legitimate user possesses a key share already, it needs to collect another $m - 1$ key shares to reconstruct the key. It is straightforward that $i_1 p_1$ must be at least $m - 1$. Thus we have:

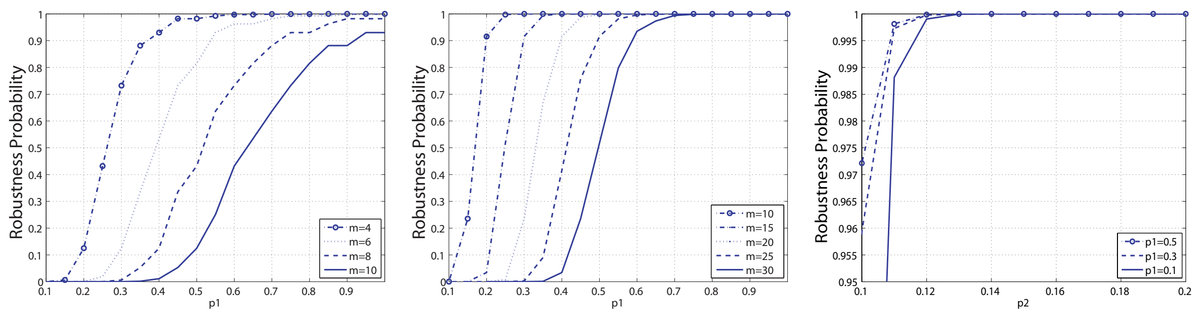
$$\begin{aligned}Pr(i_1 p_1 \geq m-1) &= 1 - Pr(i_1 < \frac{m-1}{p_1}) \\ &= 1 - \sum_{j=1}^{\left\lfloor \frac{m-1}{p_1} \right\rfloor - 1} Pr(N = j) \\ &= 1 - \sum_{j=1}^{\left\lfloor \frac{m-1}{p_1} \right\rfloor - 1} \frac{\gamma^j}{j!} e^{-\gamma}, \qquad (6)\end{aligned}$$

where $\gamma = \rho\pi r^2$.

Multi-hop scenario. This scenario considers the case that there are less than $m - 1$ key shares available in the $(k-1)$-hop $(k \geq 2)$ neighborhood, but at least $m-1$ key shares in the k-hop neighborhood of the collector node for key re-construction. The $(k - 1)$-hop neighborhood covers an area $A_{k-1} = \pi((k - 1)r)^2$, while the k-hop neighborhood of a node (with transmission range r) covers an area $A_k = \pi(kr)^2$. Let i_{k-1} and i_k be the number of neighbors in the $(k - 1)$-hop and k-hop neighborhood. Similar to the 1-hop scenario, we define p_k as the percentage of nodes in k-hop neighborhood that carry key shares, where $0 < p_k < 1$. Since the legitimate user (holding a key share already) can collect $t \in [m-1, n-1]$ key shares from the k-hop neighborhood but less than $m - 1$ neighbors from the $(k - 1)$-hop neighborhood, we have:

$$\begin{aligned}&Pr(m-1 \leq i_k p_k \leq n-1 | i_{k-1} p_{k-1} < m-1) \\ &= \frac{Pr(\frac{m-1}{p_k} \leq i_k \leq \frac{n-1}{p_k}) + Pr(i_{k-1} < \frac{m-1}{p_{k-1}})}{Pr(i_{k-1} < \frac{m-1}{p_{k-1}})} \\ &- \frac{Pr(\frac{m-1}{p_k} \leq i_k \leq \frac{n-1}{p_k} \cup i_{k-1} < \frac{m-1}{p_{k-1}})}{Pr(i_{k-1} < \frac{m-1}{p_{k-1}})} \\ &= \frac{Pr(\frac{m-1}{p_k} \leq i_k \leq \frac{n-1}{p_k}) + Pr(i_{k-1} < \frac{m-1}{p_{k-1}})}{Pr(i_{k-1} < \frac{m-1}{p_{k-1}})} \\ &- \frac{1 - Pr((i_k \leq \frac{m-1}{p_k} \cup i_k \geq \frac{n-1}{p_k}) \cap i_{k-1} \geq \frac{m-1}{p_{k-1}})}{Pr(i_{k-1} < \frac{m-1}{p_{k-1}})}, \qquad (7)\end{aligned}$$

In Equation 7, since it is impossible that there are more key shares in the $(k-1)$-hop neighborhood than in the k-hop neighborhood, $Pr(i_k \leq \frac{m-1}{p_k} \cap i_{k-1} \geq \frac{m-1}{p_{k-1}}) = 0$. Furthermore, since $i_k p_k \leq n - 1$, it must be true that $i_k \leq \frac{n-1}{p_k}$. Thus $Pr(i_k \geq \frac{n-1}{p_k} \cap i_{k-1} \geq \frac{m-1}{p_{k-1}}) = 0$. So we have $Pr((i_k \leq \frac{m-1}{p_k} \cup i_k \geq \frac{n-1}{p_k}) \cap i_{k-1} \geq \frac{m-1}{p_{k-1}}) = 0$.



(a) 1-hop scenario, n = 15 (b) 1-hop scenario, n = 50 (c) 2-hop scenario, n = 15

Fig. 4. Robustness probability for 1-hop scenario with (a) m = 4, 6, 8, and 10, (b) m = 10, 15, 20, 25, and 30 in the secret sharing method, and for 2-hop scenario with (c) (10, 15) secret sharing scheme.

Then continuing from Equation 7, we have:

$$\begin{aligned} &Pr(m-1 \leq i_k p_k \leq n-1 | i_{k-1} p_{k-1} < m-1) \\ &= \frac{Pr(\frac{m-1}{p_k} \leq i_k \leq \frac{n-1}{p_k}) + Pr(i_{k-1} < \frac{m-1}{p_{k-1}}) - 1}{Pr(i_{k-1} < \frac{m-1}{p_{k-1}})} \\ &= 1 - \frac{1 - Pr(\frac{m-1}{p_k} \leq i_k \leq \frac{n-1}{p_k})}{Pr(i_{k-1} < \frac{m-1}{p_{k-1}})} \\ &= 1 - \frac{Pr(i_k < \frac{m-1}{p_k} \cup i_k > \frac{n-1}{p_k})}{Pr(i_{k-1} < \frac{m-1}{p_{k-1}})} \end{aligned} \tag{8}$$

Since $i_k p_k \leq n-1$, it is impossible that $i_k > \frac{n-1}{p_k}$. Thus continuing from Equation 8,

$$\begin{aligned} &Pr(m-1 \leq i_k p_k \leq n-1 | i_{k-1} p_{k-1} < m-1) \\ &= 1 - \frac{Pr(i_k < \frac{m-1}{p_k})}{Pr(i_{k-1} < \frac{m-1}{p_{k-1}})} \\ &= 1 - \frac{\sum_{j=1}^{\lfloor \frac{m-1}{p_k} \rfloor - 1} \frac{\gamma^j}{j!} e^{-\gamma}}{\sum_{j=1}^{\lfloor \frac{m-1}{p_{k-1}} \rfloor - 1} \frac{\gamma'^j}{j!} e^{-\gamma'}}, \end{aligned} \tag{9}$$

where the expected node degree $\gamma' = \rho\pi(k-1)^2 r^2$ and $\gamma = \rho\pi k^2 r^2$.

Discussion. Based on the theoretical analysis, we choose different parameter setup to measure the probability of robustness in our approach.

We fix the value of γ (e.g., γ = 15), the expected number of nodes in 1-hop neighborhood, and vary the value of p_1, the ratio of the nodes in the 1-hop neighborhood that holds key shares. Figure 4 (a) presents the robustness probability of key reconstruction in the 1-hop neighborhood with $n = 15$ and $m = 4, 6, 8$ and 10 for the setup of the (m, n) secret sharing scheme. We observed that for all the m values, the robustness probability increases with increasing p_1. This is straightforward as the more key shares moving together, they make better chance for key reconstruction. Figure 4 (b) shows the results when we change to $n = 50$ and $m = 10, 15, 20, 25$ and 30 for the setup of the (m, n) secret sharing scheme. It has a similar trend as Figure 4 (a). Furthermore, we observed that the smaller m value is, the smaller p_1 is needed to achieve a robustness probability threshold, since fewer number of key shares are needed for key reconstruction.

Figure 4 (c) presents the robustness probability in a 2-hop scenario with the (10, 15) secret sharing method. Similar to the 1-hop neighborhood scenario, we vary the value of p_2, the ratio of the nodes holding key shares in the 2-hop neighborhood. Meanwhile, we set the value of p_1, the percentage of nodes holding key shares in the 1-hop neighborhood, as 0.1, 0.3 and 0.5 respectively. In general, the analytical robustness probability is as high as near to 1 for most of the cases. This indicates the feasibility of applying TRAP using the secret sharing method to mobile wireless networks. To ensure the robustness, we can choose appropriate m, n and γ values to ensure that the robustness probability is no less than a user-defined threshold.

4.2 Security Analysis

To possess the decryption key, an adversary will try to compromise at least m nodes to obtain their key shares for key reconstruction. Next, we compare the probability that a legitimate user and an attacker can successfully reconstruct the keys from the k-hop neighborhood. Note here we do not have to consider the $(k-1)$-hop neighborhood, since the security analysis only reasons on whether the legitimate users (and the attacker) can collect sufficient number of key pieces from the k-hop neighborhood.

Legitimate User. We use the same notations as in Section 4.1, where k ($k \geq 1$) is the number of hops, i_k is the number of nodes in the k-hop neighborhood, and p_k is the percentage of nodes in the k-hop neighborhood that carry key shares. The probability that the legitimate user can successfully reconstruct the key is:

$$Pr(m-1 \leq i_k p_k \leq n-1) = Pr(\frac{m-1}{p_k} \leq i_k \leq \frac{n-1}{p_k}) = \sum_{j=\lfloor \frac{m-1}{p_k} \rfloor+1}^{\lceil \frac{n-1}{p_k} \rceil-1} Pr(n=j) = \sum_{j=\lfloor \frac{m-1}{p_k} \rfloor+1}^{\lceil \frac{n-1}{p_k} \rceil-1} \frac{\gamma^j}{j!} e^{-\gamma} \quad (10)$$

where $\gamma = \pi \rho k^2 r^2$.

Attacker. The only difference between the legitimate users and the attacker is that the legitimate users only need to collect at least $m-1$ key shares (and at most $n-1$ key shares), while the attacker needs to collect at least m key shares (and at most n key shares). Thus the probability that the attacker can successfully reconstruct the key is:

$$Pr(m \leq i_k p_k \leq n) = Pr(\frac{m}{p_k} \leq i_k \leq \frac{n}{p_k}) = \sum_{j=\lfloor \frac{m}{p_k} \rfloor+1}^{\lceil \frac{n}{p_k} \rceil-1} Pr(n=j) = \sum_{j=\lfloor \frac{m}{p_k} \rfloor+1}^{\lceil \frac{n}{p_k} \rceil-1} \frac{\gamma^j}{j!} e^{-\gamma} \quad (11)$$

where $\gamma = \pi \rho k^2 r^2$.

The comparison of Equation 10 and 11 leads to an important observation:

Theorem 4.1: In the k-hop neighborhood, a legitimate user has the *advantage probability* of $\sum_{j=\lfloor \frac{m-1}{p_k} \rfloor+1}^{j=\lfloor \frac{m}{p_k} \rfloor} \frac{\gamma^j}{j!} e^{-\gamma}$ - $\sum_{j=\lceil \frac{n-1}{p_k} \rceil}^{j=\lceil \frac{n}{p_k} \rceil-1} \frac{\gamma^j}{j!} e^{-\gamma}$ for key reconstruction than an attacker, where $\gamma = \pi \rho k^2 r^2$.

Following Theorem 4.1, with fixed n and γ, we can control the amount of legitimate users' advantage over the attacker by choosing appropriate m values. We can also incorporate additional techniques such as authentication in [20] to our approach to increase the difficulty that the attacker can obtain all m key shares, which will as well increase the advantage of successful key reconstruction of the legitimate users over the attacker.

Then, we analyze how the advantage probability is affected by the secret sharing scheme with respect to the varying m, and fixed the other parameters. Let's consider,

$$\sum_{j=\lfloor \frac{m-1}{p_k} \rfloor+1}^{j=\lfloor \frac{m}{p_k} \rfloor} \frac{\gamma^j}{j!} e^{-\gamma} - \sum_{j=\lceil \frac{n-1}{p_k} \rceil}^{j=\lceil \frac{n}{p_k} \rceil-1} \frac{\gamma^j}{j!} e^{-\gamma} > \alpha, \quad (12)$$

where α is a given threshold of the advantage probability. We have,

$$\sum_{j=\lfloor \frac{m-1}{p_k} \rfloor+1}^{j=\lfloor \frac{m}{p_k} \rfloor} \frac{\gamma^j}{j!} > \sum_{j=\lceil \frac{n-1}{p_k} \rceil}^{j=\lceil \frac{n}{p_k} \rceil-1} \frac{\gamma^j}{j!} + \frac{\alpha}{e^{-\gamma}}. \quad (13)$$

From the sterling approximation $j! \approx \sqrt{2\pi j}(\frac{j}{e})^j$, we can approximate the LHS of the Equation 13. Then, we further replace the j in $\frac{\gamma e}{j}$ and $\sqrt{2\pi j}$ by $\lfloor \frac{m-1}{p_k} \rfloor+1$, which is the lower region of j. Thus, we can have the LHS of the Equation 13 as,

$$\sum_{j=\lfloor \frac{m-1}{p_k} \rfloor+1}^{j=\lfloor \frac{m}{p_k} \rfloor} \frac{\gamma^j}{j!} \approx \sum_{j=\lfloor \frac{m-1}{p_k} \rfloor+1}^{j=\lfloor \frac{m}{p_k} \rfloor} (\frac{\gamma e}{j})^j (2\pi j)^{-\frac{1}{2}} < \sum_{j=\lfloor \frac{m-1}{p_k} \rfloor+1}^{j=\lfloor \frac{m}{p_k} \rfloor} (\frac{\gamma e}{\lfloor \frac{m-1}{p_k} \rfloor+1})^j (2\pi(\lfloor \frac{m-1}{p_k} \rfloor+1))^{-\frac{1}{2}}. \quad (14)$$

Thus, we can rewrite Equation 13 by inserting Equation 14 as

$$\sum_{j=\lfloor \frac{m-1}{p_k} \rfloor+1}^{j=\lfloor \frac{m}{p_k} \rfloor} (\frac{\gamma e}{\lfloor \frac{m-1}{p_k} \rfloor+1})^j (2\pi(\lfloor \frac{m-1}{p_k} \rfloor+1))^{-\frac{1}{2}} > \sum_{j=\lceil \frac{n-1}{p_k} \rceil}^{j=\lceil \frac{n}{p_k} \rceil-1} \frac{\gamma^j}{j!} + \frac{\alpha}{e^{-\gamma}}. \quad (15)$$

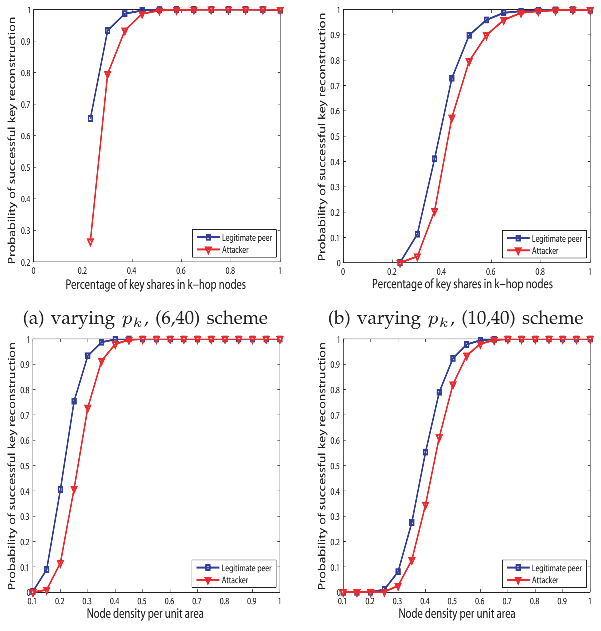
Let $M = \frac{\gamma e}{\lfloor \frac{m-1}{p_k} \rfloor+1}$, the constant $C = \sum_{j=\lceil \frac{n-1}{p_k} \rceil}^{j=\lceil \frac{n}{p_k} \rceil-1} \frac{\gamma^j}{j!} + \frac{\alpha}{e^{-\gamma}}$, and replace it in Equation 15. Then we have,

$$\sum_{j=\lfloor \frac{m-1}{p_k} \rfloor+1}^{j=\lfloor \frac{m}{p_k} \rfloor} (M)^j (\frac{2\pi\gamma e}{M})^{-\frac{1}{2}} > C. \quad (16)$$

If $M = 1$, we then have,

$$\sum_{j=\lfloor \frac{m-1}{p_k} \rfloor+1}^{j=\lfloor \frac{m}{p_k} \rfloor} (M)^j (\frac{2\pi\gamma e}{M})^{-\frac{1}{2}} = (2\pi\gamma e)^{-\frac{1}{2}} (\lfloor \frac{m}{p_k} \rfloor - \lfloor \frac{m-1}{p_k} \rfloor) > C, \quad (17)$$

where $\lfloor \frac{m-1}{p_k} \rfloor = \gamma e - 1$. In this case, we can determine m from the known parameters γ, e, and p_k. Thus, the threshold of advantage probability α in C should follow $\alpha < e^{-\gamma}(2\pi\gamma e)^{-\frac{1}{2}}(\lfloor \frac{m}{p_k} \rfloor - \lfloor \frac{m-1}{p_k} \rfloor) - e^{-\gamma} \sum_{j=\lceil \frac{n-1}{p_k} \rceil}^{j=\lceil \frac{n}{p_k} \rceil-1} \frac{\gamma^j}{j!}$.



(a) varying p_k, (6,40) scheme (b) varying p_k, (10,40) scheme

(c) varying ρ, (6,40) scheme (d) varying ρ, (10,40) scheme

Fig. 5. (a) and (b) are the probability of successful key reconstruction vs. percentage of nodes in the k-hop neighborhood that carry key shares p_k with (a) (6,40) scheme, (b) (10,40) scheme; (c) and (d) are probability of successful key reconstruction vs. node density per unit area ρ with (c) (6,40) scheme, (d) (10,40) scheme.

If $M \neq 1$, we then have,

$$\begin{aligned}
&\sum_{j=\lfloor \frac{m-1}{p_k} \rfloor+1}^{j=\lfloor \frac{m}{p_k} \rfloor} (M)^j \left(\frac{2\pi\gamma e}{M}\right)^{-\frac{1}{2}} \\
&= \left(\frac{M}{2\pi\gamma e}\right)^{\frac{1}{2}} M^{\lfloor \frac{m-1}{p_k} \rfloor+1} \frac{M^{(\lfloor \frac{m}{p_k} \rfloor - \lfloor \frac{m-1}{p_k} \rfloor)} - 1}{M-1} \\
&= (2\pi\gamma e)^{-\frac{1}{2}} \frac{M^{(\lfloor \frac{m}{p_k} \rfloor + \frac{3}{2})} - M^{(\lfloor \frac{m-1}{p_k} \rfloor + \frac{3}{2})}}{M-1} \\
&> C. \qquad (18)
\end{aligned}$$

Equation 18 can guide us to pick appropriate m and n values (i.e., different secret sharing schemes) to satisfy the α-advantage probability constraint. More specifically, from Equation 18, $M^{(\lfloor \frac{m}{p_k} \rfloor + \frac{3}{2})}$ in the numerator is the dominant term. Thus, if the threshold of advantage probability α in C increases, the value of m needs to decrease accordingly when other parameters (γ, e, p_k, and n) are fixed.

Discussion. In Figure 5 (a) and (b), we measure the probability of successful key reconstruction of legitimate users and the attacker for both $(6, 40)$ and $(10, 40)$ secret sharing schemes. The value of γ is set to 15. We observed an increasing trend of successful key reconstruction probability for both legitimate users and the attacker when p_k increases. However, the attacker always has worse chance to successfully reconstruct the key than the legitimate users.

We then vary the value of ρ, the node density per unit area, and measure the probability of successful key reconstruction of legitimate users and the attacker again. The results in Figure 5 (c) and (d) indicate that larger ρ values bring larger key reconstruction probability for both legitimate users and the attacker. We also observe that larger ρ values, the node density per unit area, bring larger key reconstruction probability for both legitimate users and the attacker, as there will be more nodes in the neighborhood for key reconstruction with the increasing values of ρ. The reason is similar to the discussion of increasing p_k above.

Figure 5 provides insights about the probability of successful key reconstruction of a legitimate user and an attacker separately. We next present the advantage probability of a legitimate user in Table 1 by examining the increasing values of p_k and m while fixing the parameters of $\rho = 0.5$, $\gamma = 15$, and $n = 40$. We found that the advantage probability sustains a high value above 0.84 when p_k is less than 0.9. The advantage probability decreases to 0.56 when p_k increases to 0.9 with m=11. This is because when the percentage of nodes carrying key shares p_k increases, the attacker's probability of obtaining the key shares from neighborhood increases accordingly. Based on these observations, we can control the security guarantee of our framework by carefully choosing parameters in our scheme.

ρ	p_k	m	Advantage probability
0.5	0.2	3	0.9483
0.5	0.3	4	0.8459
0.5	0.4	6	0.9483
0.5	0.5	7	0.8459
0.5	0.6	8	0.8459
0.5	0.7	10	0.9483
0.5	0.8	11	0.8459
0.5	0.9	11	0.5674

TABLE 1
Advantage Probability of the corresponding m value with ρ=0.5, γ=15, n=40, and various $p_k = 0.2 \sim 0.9$.

5 Key Distribution Schemes

Careless key distribution will result in key shares scattered across the whole network and thus degrade the performance of key reconstruction. Therefore, how the key shares are distributed is of utmost importance. Based on our theoretical analysis in Section 4, ideally the key shares should be distributed to the nodes that move together in the network. In this section, we investigate three schemes to decide which nodes should be assigned the key shares in the TRAP protocol. The three key distribution schemes can be classified into two types, the one that does not respect the relationships between moving patterns of different nodes (i.e., correlation-blind scheme), and

Time Point	Neighbor ID
T_1	c_1, c_3, c_4
T_2	c_2, c_3, c_5
T_3	c_1, c_2, c_3, c_5
T_4	c_1, c_2, c_5

(a) The neighborhood of c_0

Neighbor Set	Support
$\{c_1\}$	0.75
$\{c_2\}$	0.75
$\{c_3\}$	0.75
$\{c_4\}$	0.25
$\{c_5\}$	0.75

(b) The support of 1-node set

Neighbor Set	Support
$\{c_1, c_2, c_3\}$	0.25
$\{c_1, c_2, c_5\}$	0.5
$\{c_2, c_3, c_5\}$	0.5
$\{c_1, c_3, c_5\}$	0.25

(c) The support of 3-node set

TABLE 2
Illustration of the *Association-rule-based* scheme.

the one that does (i.e., correlation-aware schemes). For the first type, we design a scheme named *random selection*, while for the second type, we design two schemes, namely *association-probability-based*, and *association-rule-based*. As it is possible that there are less than n nodes in the 1-hop neighborhood of the current node, these schemes aim at choosing $x \leq n$ nodes, where x is either the number of available nodes in the current 1-hop neighborhood (when there are less than n such nodes), or n, when there are sufficient n nodes in the 1-hop neighborhood.

5.1 Correlation-blind Scheme

We design the *random selection* scheme that picks x nodes without considering the moving patterns of these nodes. The idea is straightforward: the x nodes are randomly picked from the 1-hop neighborhood, without taking the moving patterns of these nodes into consideration. This is a naive approach to distribute the key shares. It is obvious that this scheme may be suffered from inefficient key reconstruction, as the x nodes that hold key shares are likely to move apart in the future and consequently the collection of x key shares from these x nodes will be costly in terms of communication overhead. Unfortunately, most existing schemes [20], [25], [32] use this random-selection strategy to do the key distribution.

5.2 Correlation-aware Schemes

The key to improve the performance of key reconstruction procedure is to distribute the key shares to the nodes that are moving together, i.e., the nodes that have strong correlations between their moving trajectories. To achieve this goal, we design two schemes, namely *association-probability-based*, and *association-rule-based*, to determine the x nodes for key distribution by considering the correlations between their moving patterns. For these two schemes, we use different mechanisms to measure the correlations/associations between different moving trajectories.

5.2.1 *Association-probability-based Scheme*

In this scheme, we measure the correlation/association between different moving trajectories as probability. In particular, given a current node c and a candidate node c', let T and T' be the trajectories of c and c' within the time window W, then the association probability Pr_a between c and c' is computed as $Pr_a = C/|W|$, where C is the number of time points in W that c' is in the 1-hop neighborhood of c. We rank the probability Pr_a in descending order and pick the top-x nodes in the sorted list as the key distributees.

5.2.2 *Association-rule-based Scheme*

Association rule technique is a well-known machine learning mechanism that can effectively discover hidden associations in the collection of data. In general, an association rule is defined as an expression $X \Rightarrow Y$, where X and Y are value set, with support $s\%$. It indicates the fact that X tends to be associated with Y, with the evidence that $s\%$ of tuples contain both X and Y. We adapt it to our problem for finding x nodes that have associated moving patterns. To be more specific, we try to find the rule $X \Rightarrow Y$ from D of the highest support $s\%$, where $X = \{c\}$, the current node that is looking for candidates from its current 1-hop neighborhood, and Y is a set of x nodes $\{c_1, \ldots, c_x\}$, i.e. an x-node set. The rule indicates the fact that node c is moving together with nodes $c_1, \ldots, c_x$. The support $s\%$ equals to the number of time points at which both $\{c\}$ and $\{c_1, \ldots, c_x\}$ locate in the 1-hop neighborhood.

There has been active research on efficient association rule mining algorithms. However, we cannot directly apply these algorithms to our problem, as they return the association rules whose supports are no less than a given threshold, while in our case, we look for the x-node association rule of the highest support, which is unknown before mining. If we set the threshold as 0, it will result in computing all possible $\binom{t}{x}$ (t : number of candidate nodes) combinations of associations, which will be very expensive. Therefore, our goal is to efficiently discover the x-node association rule that is of the highest support from the trajectory data. If there are multiple such rules, we pick the one of the largest support, and choose the x nodes in the Y side of the rule.

The general principle of most of association rule mining algorithms for efficient mining is to make use of the *monotone* property of the association rules, which refers to the fact that any subset of a frequent itemset (i.e., of large support) must be frequent [2]. Thus generating the candidate itemsets in each pass only needs to use the frequent itemsets found in the previous pass. We utilize this property and design the following algorithm. First, given the current node c that is looking for candidates from its current 1-hop neighborhood, for each candidate node c', we

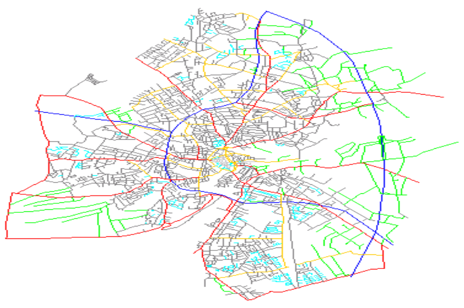


Fig. 6. The simulation data sets are generated based on the city and its vicinity in Germany.

compute the support of 1-node association $\{c\} \Rightarrow \{c'\}$, and rank these supports in descending order. Following the *monotone* property of association rules, the target x-node association of the top-1 support must be chosen from the 1-node associations of the top-x support (i.e., the support of the top x-th item in the sorted item list). Therefore, we pick the 1-node associations of the top-x support. If there are exactly x such nodes, they are the x key distributee nodes that we look for. Otherwise, out of the $x' > x$ nodes, we compute the support for all possible x-node associations, and output the one of the largest support.

Our algorithm only needs at most $\binom{x'}{x}$ passes to find x key distributee nodes. Compared with checking all possible $\binom{t}{x}$ $(t > x')$ choices, where t is the set of all possible candidate nodes, our algorithm is much more efficient.

We use an example to illustrate our algorithm. Consider a collector node c_0 whose neighborhood at various time points is shown in Table 2 (a). Assume x=3. We first calculate the support of all 1-node association, with the result shown in Table 2 (b). There are four nodes c_1, c_2, c_3 and c_5 that are of top-3 support 0.75. Then we calculate the support of $\binom{4}{3}$ = 4 possible 3-node sets. Table 2 (c) shows that out of these four candidates, $\{c_1, c_2, c_5\}$ and $\{c_2, c_3, c_5\}$ both have the same highest support value. We pick one and return it as the final result.

6 Simulation Evaluation

In this section, we describe our simulation methodology and present the results that evaluate the effectiveness of our schemes.

6.1 Methodology

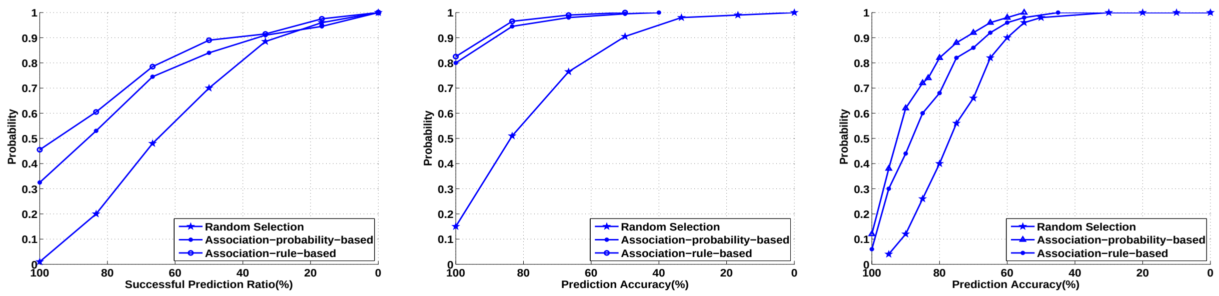
Data sets generation from city environment. We would like to evaluate the feasibility of applying our approaches in real world scenarios (e.g. traffic monitoring in VANETs) using mobile wireless networks. Thus, we conducted simulations based on mobile devices generated from a city environment and its vicinity in Germany [7] as shown in Figure 6. The size of the area is $25000m \times 25000m$. We generated 1000 nodes and placed them randomly inside the city as a real-world network. To further simulate real-world scenarios, during the simulation period some new nodes may move into the city environment and some existing nodes may move out the city environment. We studied two different scenarios with respect to the traveling speed of the node: walking speed (5ft/sec) and vehicular traveling speed (50ft/sec) by randomly choosing multiple subsets of nodes. The scenario using the regular walking speed simulates data collection through mobile phones carried by people, while the scenario with the vehicular traveling speed intends to study the applications enabled by the data collected through VANETs. There are no predefined trajectories for each node. However, group of nodes may travel together to common destinations (e.g. shopping malls or museums in the city).

NS-2 simulation. We then evaluate the effectiveness of our key distribution schemes by controlling the multi-hop wireless networks. We simulated the environment under NS-2 network simulator by applying our proposed TRAP protocol and three key distribution schemes. We used Random way Point Group Mobility (RPGM) [16] as our mobility model. 10 groups of nodes are simulated in our simulation. Individual nodes within a group move slightly different, but their movements are constrained within the group. We studied two different scenarios: 1-hop and 2-hop. In the 1-hop scenario, the average number of neighbors is 20, and there are enough neighbors to distribute key shares. Whereas in the 2-hop scenario, the average number of neighbors is 10, and there are fewer neighbors for key distribution. In our simulation, we configured the transmission range of each node as 50 meters, and the simulation area as 500 by 500 meters. We applied the secret sharing scheme with n=15, m=4, 6, 8, and 10, respectively.

6.2 Metrics

We utilize the following metrics to evaluate the effectiveness of our key distribution schemes using neighborhood prediction:

Prediction Accuracy. We measure the effectiveness of the key distribution through neighborhood prediction. We split our simulation study time into two periods: *past* and *future*. The data in the *past* is used to perform prediction, whereas the data in the *future* is used to validate the prediction accuracy. For a given collector node, we define the *prediction accuracy* as the percentage of the intersection of the predicted devices that will travel together in the future ($\{N_{predict}\}$) and the devices that are indeed traveling together in the future ($\{N_{future}\}$): $\frac{|\{N_{predict}\} \cap \{N_{future}\}|}{|\{N_{predict}\}|}$. We will evaluate the effectiveness of our key distribution schemes by studying the statistical characteristics of the prediction accuracy through calculating its Cumulative

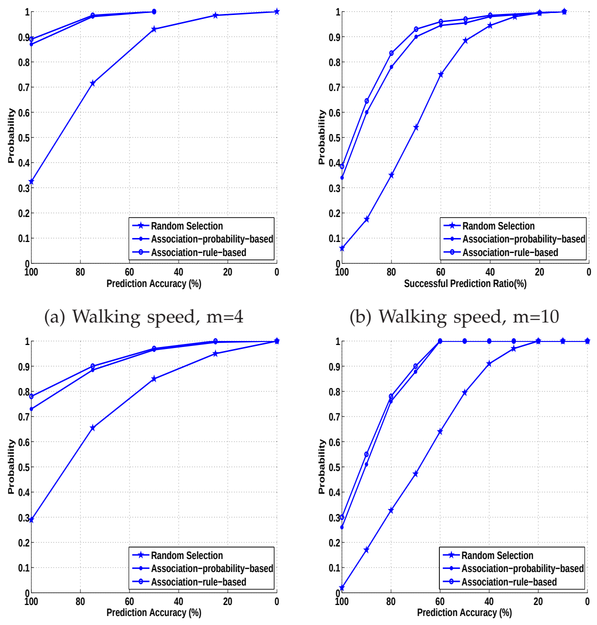


(a) n=15, m=6

(b) n=50, m=6

(c) n=50, m=20

Fig. 7. Cumulative Distribution Function(CDF) of prediction accuracy under the walking speed.



(a) Walking speed, m=4

(b) Walking speed, m=10

(c) Vehicular speed, m=4

(d) Vehicular speed, m=10

Fig. 8. CDF of prediction accuracy under different traveling speed when $n = 50$.

Distribution Function (CDF) and averaged prediction error.

Time Performance. By measuring the time that each scheme needs to perform neighborhood prediction for key distribution, we evaluate the efficiency across different schemes. This metric helps to benchmark our schemes in the simulation environment and further indicates the feasibility of implementing them in real wireless devices.

Percentage of successful key reconstruction. We measure the percentage of successful key reconstruction to evaluate the effectiveness of key distribution. The percentage of successful key reconstruction is defined as the number of times that the keys can be re-constructed successfully over the total number of simulation runs.

6.3 Results

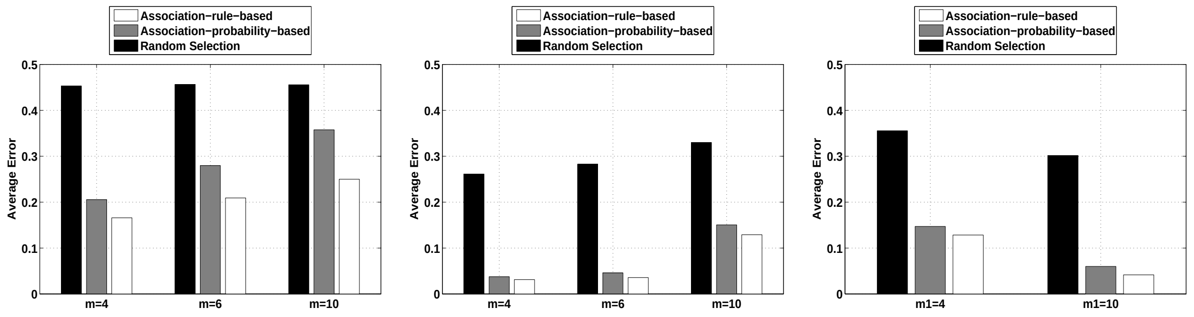
Cumulative Distribution Function (CDF) Measurement: 1-hop Scenario. We are interested in studying what is the probability of different key distribution schemes that can perform neighborhood prediction with 100th, 75th, 50th, and 25th percentile accuracy. Figure 7 presents the CDF of prediction accuracy with respect to different m and n in the secret sharing method under the walking speed. In Figure 7, n is set to 15 and 50 respectively and m is set to 10 and 20 respectively. In this simulation setup, n is the network size and all the nodes are within the transmission range of the collector node. We observed that *Association-rule-based* scheme tops out the performance, whereas *Random Selection* has the worst prediction performance. In general, Correlation-aware schemes outperform the Correlation-blind scheme.

Further, we found that under a fixed m, the larger the n is the higher the prediction accuracy can be achieved. Because under a larger n, there are more nodes that are holding key shares travel together with the collector node, and thus the probability of a successful key reconstruction is increased.

On the other hand, under a fixed n, the smaller the m is the higher the prediction accuracy can be achieved. Because under a smaller m, it requires fewer nodes that are holding key shares travel together in order to achieve successful key reconstruction.

Figure 8 presents the comparison of the Prediction Accuracy CDFs under different traveling speed, walking speed and vehicular speed, with different setups of the secret sharing method, i.e.,(4, 50) and (10, 50). We observed the similar performance trend as in Figure 7: Correlation-aware schemes outperform the Correlation-blind scheme. Further, the performance of our key distribution schemes under the vehicular speed is qualitatively the same as the performance under the walking speed. This indicates that our approach is generic across different device traveling speed.

Averaged Prediction Error: 1-hop scenario. Figure 9 (a) and (b) present the percentage of the prediction error versus different (m, n) setups in the secret sharing method across our key distribution



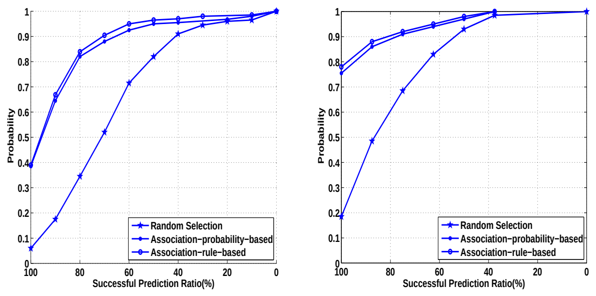
(a) 1-hop scenario, n = 15 (b) 1-hop scenario, n = 50 (c) 2-hop scenario, n = 50

Fig. 9. Averaged prediction error using TRAP under the walking speed.

schemes under the walking speed. We observed that Correlation-aware schemes incur smaller prediction errors (less than 36%) and the *Association-rule-based* scheme presents the smallest prediction errors in all cases. Further, under a fixed n, the prediction error increases with the increasing number of m. Overall, the results of averaged prediction errors are inline with the observations of prediction accuracy in Figure 7. This is encouraging as it indicates that our key distribution schemes are highly effective in distributing the key shares to those devices that are traveling together with the collector node.

TRAP: 2-hop Scenario. We next present the results when there are not enough devices within the transmission range of the collector node and the key distribution will be performed in the multi-hop range of the collector node. Figure 10 presents the CDF of the prediction accuracy for a (20, 50) secret sharing method in a 2-hop scenario. During key reconstruction, there are not enough key shares within the transmission range of the collector node that are traveling together with it, e.g., m_1=4 and 6 in this simulation. The rest of the key shares will be collected through the second hop of the collector node using TRAP. We found that the key distribution schemes have better performance when $m_1 = 10$ as shown in Figure 10 (b) than those when $m_1 = 4$ (Figure 10 (a)). This is because when there are more key shares can be found in the 1-hop range, there will be less nodes carrying key shares need to be found in the 2-hop range, and consequently the prediction accuracy increases. Further, when there are more key shares need to be collected in the 2-hop range, i.e., $m_1 = 4$, the *Association-rule-based* scheme can still reach the probability of 84% to achieve the prediction accuracy of 80% or higher. Additionally, the results of the averaged prediction error shown in Figure 9 (c) are consistent with our prediction accuracy. Thus, these results provide strong evidence of the effectiveness of TRAP.

Time performance. Finally, we study the time efficiency of our key distribution schemes. Table 3 presents the time measurements of our schemes when using various setups of m and n in the secret sharing method. We observed that the time to perform key distribution through neighborhood prediction is in the order of milliseconds for all the schemes by using a DELL desktop with Intel Core2 Q6600 2.4GHz processor. Further, we found that the schemes, e.g., *Association-rule-based* scheme, which provide higher prediction accuracy run slower. Thus, there exists a tradeoff between the prediction accuracy and the computation time. Our results will provide a guidance for choosing different schemes based on application needs in practice.



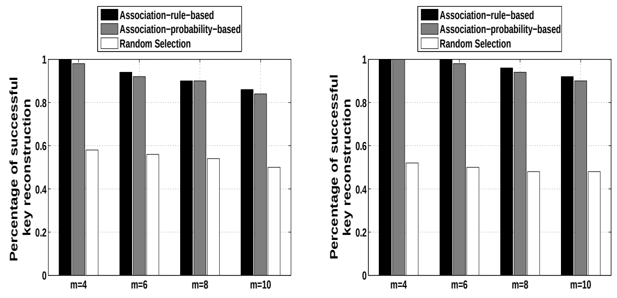
(a) n=50, m=20, m_1=4 (b) n=50, m=20, m_1=10

Fig. 10. CDF of prediction accuracy under different traveling speed when $n = 50$.

Effectiveness of key distribution schemes. Finally, we evaluate the effectiveness on the key distribution schemes in terms of the successful key reconstruction by using the NS-2 simulator with RPGM mobility

Scheme setting	(4, 15)	(6, 15)	(10, 15)
Random Selection	10.2	10.48	11.06
Association-probability-based	27.8	31.4	44.6
Association-rule-based	29.2	33.0	46.2
Scheme setting	(4, 50)	(6, 50)	(10, 50)
Random Selection	13.6	14.08	14.6
Association-probability-based	42.2	41.4	48.6
Association-rule-based	44.3	45.8	49.7

TABLE 3
Time performance (in millisecond) across different key distribution schemes



(a) 1-hop: $N = 20$, (15, m) (b) 2-hop: $N = 10$, (15, m)

Fig. 11. Percentage of successful key reconstruction vs. different settings of m in (n, m).

model. Figure 11 measures the percentage of successful key reconstruction over different m values in the (m, n) secret sharing scheme. We observed that we can obtain high key reconstruction ratio over 0.85 in both 1-hop ($N = 20$) and 2-hop ($N = 10$) scenarios for our proposed Association-rule-based and Association-probability-based methods. We note that in the 2-hop scenario, our framework requests information from 2-hop neighbors to distribute key shares and reconstruct keys. Our methods obviously outperform the Random Selection method. More specifically, we found that both Association-rule-based and Association-probability-based scheme can obtain similarly high percentage of successful key reconstruction. This is consistent with the trend we found from our results of the prediction accuracy in Section 6.3, which is obtained based on the generated data sets from a city environment. These results provide strong evidence of the effectiveness of our proposed TRAP protocol.

7 Conclusion

In this work, we proposed a fully decentralized key management framework to facilitate secure data access in mobile wireless networks, where cryptographic keys are split into multiple shares and are distributed to multiple nodes in the network. The data is cached in the collecting mobile devices within the network to reduce the high communication overhead and excessive energy consumption among wireless devices if continuously forwarding the data to centralized storage nodes.

To handle the node mobility, we further developed the Transitive Prediction (TRAP) protocol that distributes the key shares to the nodes that are moving together through neighborhood prediction for effective key reconstruction in mobile environments. Also, we measured the communication overhead incurred by our proposed TRAP protocol through theoretical analysis and simulation. In addition, as a part of TRAP, we designed three key distribution schemes to choose the distributee nodes that have co-moving patterns by analyzing the correlation relationship embedded in the trajectories of co-moving devices. We further derived the theoretical analysis of the robustness and security of our approach. Our simulation results based on data sets generated from a simulated mobile wireless network in city environment and NS-2 simulator demonstrated that our key distribution schemes are highly effective for key reconstruction. Both of our theoretical analysis and simulation results provide strong evidence of the feasibility and effectiveness of applying the decentralized key management framework to achieve resilient data confidentiality in distributed mobile environments.

Acknowledgments

The preliminary results have been published in "A Decentralized Key Management Scheme via Neighborhood Prediction in Mobile Wireless Networks" [31] in MASS 2010.

References

[1] A. Abdul-Rahman. The pgp trust model. edi-forum. *the Journal of Electronic Commerce*, 1997.
[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. pages 487–499, 1994.
[3] M. Azizyan, I. Constandache, and R. R. Choudhury. SurroundSense: Mobile Phone Localization via Ambience Fingerprinting. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*, June.
[4] D. Balfanz, D. Smetters, P. Stewart, and H. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proceedings of the 9th Annual Network and Distributed System Security Symposium (NDSS)*, 2002.
[5] C. Bettstetter and C. Hartmann. Connectivity of wireless multihop networks in a shadow fading environment. pages 28–32, 2003.
[6] C. Bettstetter and J. Zangl. How to achieve a connected ad hoc network with homogeneous range assignment: an analytical study with consideration of border effects. In *Mobile and Wireless Communications Network, 2002. 4th International Workshop on*, pages 125–129, 2002.
[7] T. Brinkhoff. Generating network-based moving objects. In *Proceedings of the 12th International Conference on Scientific and Statistical Database Management*, 2000.
[8] R. Canetti, S. Halevi, and A. Herzberg. Maintaining authenticated communication in the presence of break-ins. *Journal of Cryptology*, 13:200–0, 1998.
[9] Y. Desmedt and Y. Frankel. Shared generation of authenticators and signatures (extended abstract). In *CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, pages 457–469, London, UK, 1992.
[10] Y. Frankel and Y. G. Desmedt. Parallel reliable threshold multisignature, 1992.
[11] Y. Frankel, O. resilience Proactive, P. key Cryptosystems, P. Gemmell, P. D. Mackenzie, and M. Yung. Optimal-resilience proactive public-key cryptosystems, 1997.
[12] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust and efficient sharing of rsa functions, 1996.
[13] A. Ghose, J. Grossklags, and J. Chuang. Resilient data-centric storage in wireless ad-hoc sensor networks. In *Proceedings of the 4th International Conference on Mobile Data Management*, pages 45–62, 2003.
[14] J. Girao, D. Westhoff, E. Mykletun, and T. Araki. Tinypeds: Tiny persistent encrypted data storage in asynchronous wireless sensor networks. *Ad Hoc Networks, Elsevier*, 5:10731089, 2007.
[15] L. Gong. Increasing availability and security of an authentication service. *IEEE Journal on Selected Areas in Communications*, 11:657–662, 1993.

[16] X. Hong, M. Gerla, G. Pei, and C. chuan Chiang. A group mobility model for ad hoc wireless networks. In *ACM Int. Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 53–60, 1999.

[17] D. Joshi, K. Namuduri, and R. Pendse. Secure, redundant, and fully distributed key management scheme for mobile ad hoc networks: an analysis. *EURASIP J. Wirel. Commun. Netw.*, 2005(4):579–589, 2005.

[18] L. Kleinrock and J. Silvester. Optimum transmission radii for packet radio networks or why six is a magic number. In *National Telecommunications Conference*, volume 1, 1978.

[19] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Comput. Networks*, 43(4):499–518, 2003.

[20] H. Luo and S. Lu. Ubiquitous and robust authentication services for ad hoc wireless networks. Technical report, 2000.

[21] E. Miluzzo, N. D. Lane, K. Fodor, R. A. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell. Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application. In *Proceedings of 6th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November.

[22] D. Miorandi and E. Altman. Coverage and connectivity of ad hoc networks in presence of channel randomness. In *IEEE INFOCOM*, pages 491–502, 2005.

[23] R. D. Pietro, L. V. Mancini, C. Soriente, A. Spognardi, and G. Tsudik. Catch me (if you can): Data survival in unattended sensor networks. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2008.

[24] S. S. K. Quanjun Chen and M. Hassan. Analysis of per-node traffic load in multi-hop wireless sensor networks. *IEEE Transactions on Wireless Communications*, Feb. 2009.

[25] A. Shamir. How to share a secret. *Communication of the ACM*, 1979.

[26] M. Shao, S. Zhu, W. Zhang, and G. Cao. pDCS: Security and privacy support for data-centric sensor networks. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2007.

[27] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin. Data-centric storage in sensornets. *ACM SIGCOMM Computer Communication Review archive*, 2003.

[28] A. H. Stanis, A. Herzberg, H. Krawczyk, and M. Yung. Proactive secret sharing or: How to cope with perpetual leakage. pages 339–352. Springer-Verlag, 1995.

[29] A. Studer, E. Shi, F. Bai, and A. Perrig. Tacking together efficient authetication, revocation, and privacy in vanets. In *Proceedings of the First IEEE International Conference on Sensor and Ad hoc Communcations and Networks (SECON)*, June 2009.

[30] S. Čapkun, J.-P. Hubaux, and L. Buttyán. Mobility helps security in ad hoc networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing (MobiHoc)*, pages 46–56, 2003.

[31] X. Zheng, H. Wang, Y. Chen, H. Liu, and R. Liu. A decentralized key management scheme via neighborhood prediction in mobile wireless networks. In *IEEE 7th International Conference on Mobile Adhoc and Sensor Systems (MASS)*, Nov. 2010.

[32] L. Zhou and Z. J. Haas. Securing ad hoc networks. *IEEE Network Magazine*, 13:24–30, 1999.