PIF: A Personalized Fine-Grained Spam Filtering Scheme With Privacy Preservation in Mobile Social Networks

Kuan Zhang, *Student Member, IEEE*, Xiaohui Liang, *Member, IEEE*, Rongxing Lu, *Senior Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Mobile social network (MSN) emerges as a promising social network paradigm that enables mobile users' information sharing in the proximity and facilitates their cyber–physical–social interactions. As the advertisements, rumors, and spams spread in MSNs, it is necessary to filter spams before they arrive at the recipients to make the MSN energy efficient. To this end, we propose a personalized fine-grained filtering scheme (PIF) with privacy preservation in MSNs. Specifically, we first develop a social-assisted filter distribution scheme, where the filter creators send filters to their social friends (i.e., filter holders). These filter holders store filters and decide to block spams or relay the desired packets through coarse-grained and fine-grained keyword filtering schemes. Meanwhile, the developed cryptographic filtering schemes protect creator's private information (i.e., keyword) embedded in the filters from directly disclosing to other users. In addition, we establish a Merkle Hash tree to store filters as leaf nodes where filter creators can check if the distributed filters need to be updated by retrieving the value of root node. It is demonstrated that the PIF can protect users' private keywords included in the filter from disclosure to others and detect forged filters. We also conduct the trace-driven simulations to show that the PIF can not only filter spams efficiently but also achieve high delivery ratio and low latency with acceptable resource consumption.**

***Index Terms*—Fine-grained, mobile social network (MSN), personalized, privacy preservation, spam filter.**

I. Introduction

MOBILE social network (MSN) has become a promising social networking platform that enables group chat, social gaming, media sharing, and ubiquitous interaction among nearby users [1]. An MSN can easily be established by smartphone users in a local area. These users connect to each other through Bluetooth, WiFi, and device-to-device communications, and form an opportunistic network for a long span of years or a temporary period (e.g., several hours). For example, MSNs create rich interaction opportunities for residents in an urban neighborhood, students in a campus area, businessmen in a conference, tourists visiting a museum or scenic site, and customers in a shopping mall. In MSNs, users' interactions are enabled anytime and anywhere without any concern of the Internet access and wireless data charge. According to a recent report from comScore, Instagram users in the United States spend 98% of time with their mobile devices instead of desktop, while this percentage for Twitter users is over 86%. As we can imagine, users will have rich and quality service experiences from MSN [2], since it helps users to obtain the desired information from others (e.g., crowdsourcing) [3] rapidly, efficiently, and pervasively.

MSN users receive various types of information, such as newsletters, personal posts, rumors, and advertisements, most of which are of great value to users. For example, in Fig. 1, local stores or restaurants repeatedly disseminate their service information, flyers, and advertisements to the nearby users. A saving mom may like to have coupons, baby stuffs, and grocery sale information, while a tourist is interested in tour instructions and handicrafts. On the other hand, user's interests may change over time. Although users could quickly exchange useful information in MSNs, they may still receive a portion of the useless information, which is considered as spams [4]. However, the communications among MSN users mainly relies on users' smartphones, and happens with their opportunistic contacts. The communication overhead is much expensive due to the limited battery of smartphones and opportunistic contacts among users. Therefore, it is crucial to make the communication meaningful in MSNs, i.e., transmit desired information to users and filter spams as early as possible.

According to an investigation by Nexgate, spams over social media have increased about 355% during only the first half of 2013, while they are rapidly spreading in social networks such that every 1 of 200 social media posts is recognized as spam. Extensive research and industry efforts have been put on spam filtering in various applications. Several schemes rely on blacklist [5] or whitelist to either block spammers or admit legitimate senders. An alternative way of filtering is to check the content by matching the keyword associated with the packet [6], [7] or using machine learning techniques [8] to detect spams. Social graph and relevant characteristics are also investigated for spam filtering [9], [10]. Most of these schemes are performed by a centralized server or trusted authority, and require historical information to detect spams. When spammers are shifting to MSNs, they have more chances of going undetected [11], since

Manuscript received June 15, 2015; revised January 12, 2016; accepted January 14, 2016. Date of current version February 24, 2016. This work was supported by a Research Grant from the Natural Science and Engineering Research Council (NSERC), Canada.

K. Zhang and X. Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: k52zhang@bbcr.uwaterloo.ca; xshen@bbcr.uwaterloo.ca).

X. Liang is with the Department of Computer Science, University of Massachusetts at Boston, Boston, MA 02125 USA (e-mail: Xiaohui.Liang@umb.edu).

R. Lu is with the School of Electrical and Electronics Engineering, Nanyang Technological University, 639798 Singapore (e-mail: rxlu@ntu.edu.sg).

Digital Object Identifier 10.1109/TCSS.2016.2519819

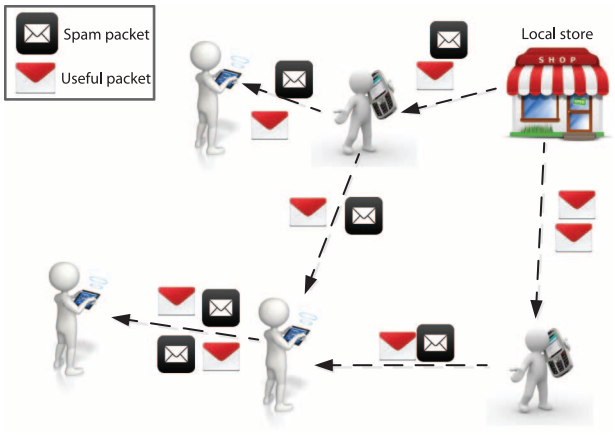


Fig. 1. Information dissemination in MSNs.

MSNs have no centralized and trusted servers and lack historical information. To this end, we propose a distributed filtering scheme where MSN users (i.e., filter creators) to personalize their spam filters, send them to others (i.e., filter holders), and allow filter holders to filter spams as early as possible. However, there are still many challenging issues for this type of filtering scheme in MSNs. The first challenge is how to distribute filters with the consideration of both distribution costs and filtering accuracy. Second, security and privacy concerns are raised, since the distributed filters may contain some sensitive information of filter creators. If the sensitive information within filters is directly exposed to others, it may violate the filter creator's privacy, such as health condition, lifestyles, and preferences [12]. The third one is how to resist malicious attackers, since the original filters may be forged to block some useful information. In addition, if a user greedily distributes his filters to all the other users in the network, it may also consume many network bandwidth and resources of others although it can benefit this individual user. These challenging issues motivate us to further improve the filtering accuracy and preserve users' privacy at the same time.

In this paper, we extend our previous conference version on spam filtering [7] and propose a personalized fine-grained spam filtering scheme (PIF) with privacy preservation in MSNs. The PIF exploits personalized filters with social-assisted filter distribution, privacy-preserving coarse-grained and fine-grained filtering, and efficient filter update. Specifically, the new contributions of this paper are threefold.

1) We develop a personalized filtering scheme with privacy preservation. The filter creator defines his filters in both coarse-grained and fine-grained manners. The keyword embedded in the coarse-grained filter enables filter holders to forward the packets including the same keyword to the filter creator. The PIF also leverages a variant of hidden vector encryption to achieve efficient fine-grained filtering. Both schemes prevent keywords in the filters from directly disclosing to others.
2) We investigate MSN users' social relationship and mobility in MSNs. Then, we exploit the opportunistic contacts among users to analyze the packet delivery process in MSNs. Based on the analysis, we propose a social-assisted filter distribution scheme that enables the filter creator to send filters to his social friends who have high probability to meet him. By doing so, the PIF can reduce the filter distribution overhead and maintain the filtering accuracy.
3) We conduct extensive simulations to show that the PIF can significantly reduce the storage and communication costs and deliver the useful packets in a low delay. Meanwhile, the security property analysis demonstrates that the PIF protect user's private keyword from directly disclosing to inside curious attackers and detect forged filters.

The remainder of this paper is organized as follows. In Section II, we review the related works on spam filtering. Then, we present the network and security models with design goals in Section III. We propose the detailed PIF scheme in Section IV, followed by the security property analysis and the simulations in Sections V and VI, respectively. Finally, we conclude this paper in Section VII.

II. Related Work

Spam filtering has attracted numerous attentions and been widely investigated recently [13]–[15]. Intuitively, some traditional filtering schemes exploit blacklist [5], whitelist, or graph [16] to block illegal senders or bypass legitimate ones. To achieve blacklist-based spam filtering, Soldo *et al.* [5] focus on predictive blacklisting to forecast attack sources according to shared historical attack logs. With a multilevel prediction, an implicit recommendation system is formulated to resist spam. Focusing on spam filtering by using keyword, Lu *et al.* [6] propose a decentralized keyword-based filtering scheme (PReFilter) to match and detect spam packets via keyword list in delay tolerant networks (DTNs). The PReFilter allows relays to have some filters generated by others. It can detect and block spams before they are transmitted to the receivers. Meanwhile, the filters with sensitive keywords are encrypted to protect user's privacy leakage. However, the PReFilter misses to consider the problems of filter distribution and update.

Social network, i.e., the social graph formed by users in the network, is another helpful methodology to detect and filter spams. Lahmadi *et al.* [17] utilize social network to collaboratively filter the short message services-based spam via the Bloom filters and content hashing filters. This collaborative filtering scheme also relies on a centralized server to build the social network among users. Hameed *et al.* [18] study the e-mail recipient's social network and mitigate spam outside of the social circle, which can also reduce the Internet bandwidth consumption by spams. To resist spam, malware and phishing via URLs, Thomas *et al.* [19] develop a real-time system, including URL aggregation, feature collection, feature extraction, and classification. The proposed system visits every URL and collects its features that are stored a centralized server for extraction in the training phase and real-time decision making. Meanwhile, some social features, such as social interests, closeness, personal preferences, and trust, are also adopted to facilitate the spam filtering. Li *et al.* [9] develop a

social network-based spam filtering framework. It can detect junk e-mails with the consideration of social features of users and network [20], so that the regular and junk e-mails can be differentiated. In [10], social trust is exploited to collaboratively filter spams. The spam reporter's trustworthiness is used to collect the correct spam reports and detect Sybil attacks at the same time. Li *et al.* [21] also exploit collaborative and privacy-preserving anti-spam system to resist a wide range of camouflage attacks. The proposed ALPACAS framework controls the amount of shared information among the collaborated entities to achieve the confidentiality of e-mails.

In addition, Fan *et al.* [22] investigate the least cost rumor blocking problem to limit the negative rumor diffusion in social network. The community feature is utilized to minimize the total number of so-called rumor protectors and protect bridge ends, as known as the boundary individuals within the neighbor communities of rumor source. Based on a susceptible-infectious model, Shah *et al.* [23] propose a systematic framework to estimate and detect the rumor source. It is formulated as a maximum-likelihood estimator for a class of graphs. Similarly, Wang *et al.* [24] detect the source of rumors with multiple observations based on the susceptible-infectious model. The multiple observations in a tree network is exploited to improve the rumor source detection probability. Different from most of the existing filtering schemes, Stringhini *et al.* [25] propose a new approach to detect spams by looking at the way how e-mails are sent instead of content and origin of e-mails. For example, it can detect the IP address from which the message is sent, and the geographical distance between the sender and the receiver. They investigate the SMTP communication between the e-mail sender and receiving mail server. The introduced concept of SMTP dialects captures small variations in the ways to carry out the SMTP protocol, so that they can distinguish the between normal e-mail senders and spam bots.

However, there are still many challenging issues for spam filtering in MSNs. First, most of the social network-based filtering schemes are based on centralized trusted authority to perform the detection, which leaves a gap of filtering schemes between online social network and MSN. Second, the decentralized schemes, e.g., PreFilter [6] and SAFE [7], are limited due to the lack of knowledge about the packet recipients (i.e., filter creator). The SAFE offers spam filtering based on keyword matching, which is a coarse-grained approach. Filter creators only select the keyword from the keyword space of the network. The coarse-grained keyword filter may not reflect the sufficient features of the delivered packets. To this end, we propose a personalized fine-grained spam filtering scheme to allow the filter creators to generate filters with different features in multiple dimensions. The proposed PIF scheme can allow creators to personalize his filters. Both coarse-grained and fine-grained filtering are integrated in the PIF.

III. Problem Definition

In this section, we present the network model and design goals including the efficiency of spam filtering and privacy preservation.

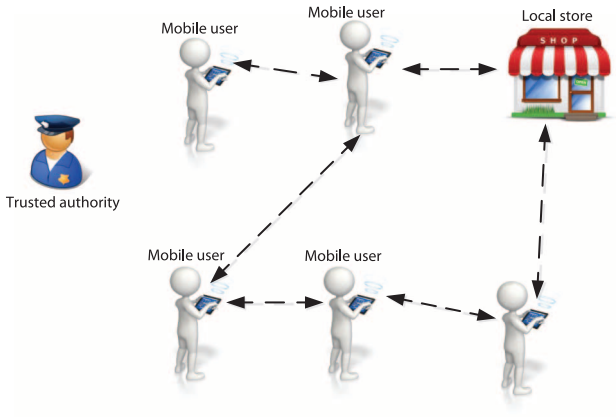


Fig. 2. Network model.

A. Network Model

We consider an MSN including a trust authority (TA) and N users (including mobile users and local stores), as shown in Fig. 2.

1) *TA* is trusted by users, and bootstraps the whole system during the initialization phase. TA can generate secret master keys used for legitimate users to generate the session keys. TA also issues certificates to legitimate users when they register. TA does not participate in the communication and filtering. However, the TA can revoke the reported malicious users.
2) *Users* include mobile users and local stores having smartphones or wearable devices to communicate with each other in the local area. They are denoted by $\mathbb{U} = \{u_1, u_2, \ldots, u_N\}$. The power and storage occupancies of each user's smartphone are limited. Each legitimate user first registers to the TA to build user's profiles and obtain key materials, e.g., unique identity, certificate, and secret keys, which should be securely kept for session key generation. In packet delivery and spam filtering phases, users can authenticate their identities and filters, and verify other user's information.

B. Security Model

Malicious users may participate in MSNs and launch attacks in the phases of packet delivery and spam filtering. We define two types of attacks: 1) inside curious attack (ICA) and 2) outside forgery attack (OFA). First, some of the filter holders may be curious about other user's preferences and personal profiles. ICA aims to violate and disclose other user's sensitive and private information. The privacy (i.e., keyword within the filters) may be leaked during filter distribution, storage, packet delivery, and filtering phases. Second, some outside adversaries cannot directly obtain other user's private information included in the filters. However, it is possible to forge other user's filters such that the useful packets may be blocked or spams can be delivered in MSNs. It would consume a large number of communication and storage overheads.

C. Design Goals

In this paper, our design goal is to develop a personalized fine-grained filtering scheme with user's privacy preservation.

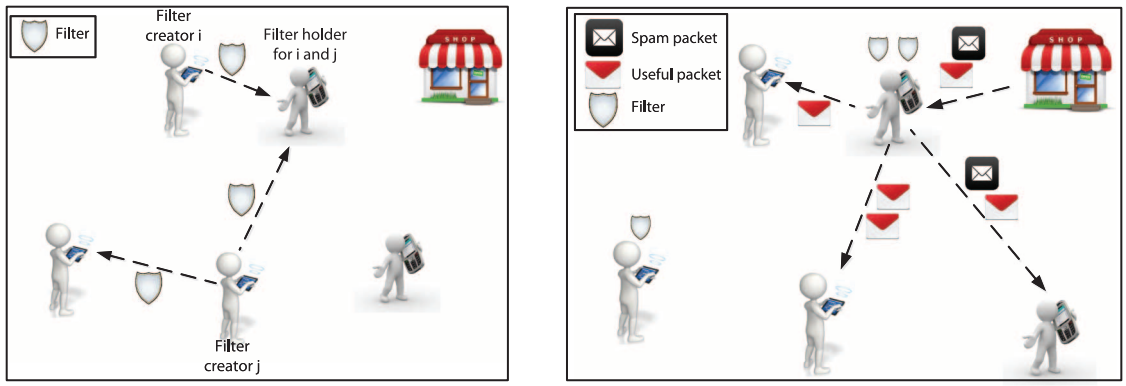


Fig. 3. PIF scheme (In filter distribution phase, filter creator sends his filters to his social friends. In filtering phase, filter holders block spams to the filter creator with his filters.). (a) Filter distribution. (b) Filtering.

1) Efficiency Goals: Due to the opportunistic contact (i.e., intermitted end-to-end connectivity) and limited smartphone battery, our goal is to develop an efficient spam filtering scheme to detect and block the spams in MSNs as early as possible. The proposed scheme should efficiently filter the spams and cost few extra storage, communication, and computational overheads. Meanwhile, it should be able to bypass useful packets without any delayed delivery of them. In addition, the distributed filters should be personalized by filter creators and updated timely.

2) Security Goals: Our security goal is to preserve the user's privacy against ICA and detect the forged filters from OFA. First, the proposed spam filtering scheme should be able to preserve the filter creator's privacy from directly disclosing. The keyword included in the distributed filters cannot appear in plaintext to others. During the filtering, the keyword should also be invisible to others and kept in the ciphertext. Second, the OFA should not be able to forge legitimate user's filters. If any filter is forged, the filter creator and other users are able to detect it efficiently.

IV. Proposed PIF Scheme

In this section, we propose the PIF scheme as shown in Fig. 3. First, the users (i.e., filter creators) build their personalized filters embedding the keywords and degree. Then, the filter creator sends his filters to his social friends (i.e., filter holders). When meeting a sender who wants to send a packet to the filter creator, filter holders use these filters to check if this packet is desired by the filter creator, and block spams in the early stage of the packet delivery. The PIF consists of social-assisted filter distribution, coarse-grained and fine-grained filters, and Merkle Hash tree-based filter authentication and update.

A. Social-Assisted Filtering Distribution

To find a proper approach to distribute filters, we first formulate the packet delivery process to understand the effective way (or relay selection) to deliver the packet in MSNs. Some frequent used notations are listed in Table I.

The packet delivery in MSNs relies on users' opportunistic contacts. Suppose the contact between two users u_i and u_j follows a Poisson distribution [26], [27] with the pairwise contact rate $\lambda_{i,j}$. A binary random variable $C_{i,j}$ is defined as

$$C_{i,j} = \begin{cases} 1, & \text{if } u_i \text{ and } u_j \text{ meet within time period } T \\ 0, & \text{otherwise.} \end{cases}$$

TABLE I
Frequently Used Notations

Notation	Description
$\lambda_{i,j}$	Contact rate between u_i and u_j
$C_{i,j}$	Contact between u_i and u_j within period T
$\overline{C_{i,j}}$	Expectation of contact times between u_i and u_j within period T
$P_i(t \leqslant T)$	Probability that u_i meets another user in T
$P^r_{s,d}(t \leqslant T)$	Forwarding probability that u_i forwards a packet to u_d through a relay u_r within T
$W_{i,x}$	u_i's xth keyword
$\mathcal{F}_i$	u_i's keyword filter set
TH	Number of common communities

Let λ_i be the average contact rate that u_i meets any other user. We have

$$\overline{C_{i,j}} = 1 \cdot \int_0^T \lambda_i e^{-\lambda_i t} dt + 0 \cdot \int_T^\infty \lambda_i e^{-\lambda_i t} dt. \quad (1)$$

Therefore, $C_{i,j}$ follows the Bernoulli distribution. As the contacts between each two users are independent [27], the probability that u_i meets another user in T is

$$P_i(t \leq T) = 1 - \prod_{\substack{u_j \in \mathbb{U} \\ j \neq i}} (1 - \overline{C_{i,j}}) = 1 - e^{-\sum_{\substack{u_j \in \mathbb{U} \\ j \neq i}} \lambda_{i,j} T}. \quad (2)$$

Let

$$\lambda_i = \sum_{\substack{u_j \in \mathbb{U} \\ j \neq i}} \lambda_{i,j},$$

then $P_i(t \le T) = 1 - e^{-\lambda_i T}$. Thus, t follows power-law distribution. The probability distribution function (pdf) is $f_i(t) = \lambda_i e^{-\lambda_i t}$ $(t \ge 0)$. We have the average contact interval of u_i as

$$\overline{E_i(t)} = \int_0^\infty t f_i(t) dt = \int_0^\infty t\lambda_i e^{-\lambda_i t} dt = \frac{1}{\lambda_i}. \quad (3)$$

According to [28]–[30], users in the same social community may have a higher probability to meet each other, since social community indicates users' personal interests. Consider the packet delivery within one community (u_s, u_r, and u_d are in the same community), if the sender u_s meets a relay u_r at t_1 and u_r meets the destination u_d at t_2, the forwarding probability $P^r_{s,d}(t = t_1 + t_2 \le T)$ is

$$\begin{aligned} P^r_{s,d}(t \le T) &= \int_0^{t_1} \lambda_{s,r} e^{-\lambda_{s,r} t} dt \cdot \int_{t_1}^{T} \lambda_{r,d} e^{-\lambda_{r,d} t} dt \\ &= \int_0^T f_{s,r}(t) \otimes f_{r,d}(t) dt \\ &= \int_{t=0}^{T} \left(\int_{\tau=0}^{t} f_{s,r}(\tau) \cdot f_{r,d}(t-\tau) d\tau \right) dt. \end{aligned} \quad (4)$$

Note that $\otimes$ is the convolution. Because u_r knows $t_{s,r}$

$$P^r_{s,d}(t = t_1 + t_2 \le T) \ge P_r(t_1 \le t_{s,r}) \cdot P_r(t_2 \le t_{s,r}). \quad (5)$$

Thus, we have

$$\begin{aligned} P^r_{s,d}(t \le T) &= \int_{t=0}^{T} \left(\int_{\tau=0}^{t} f_{s,r}(\tau) \cdot f_{r,d}(t-\tau) d\tau \right) dt \\ &\ge \int_{\tau_1=0}^{t_{s,r}} f_{s,r}(\tau_1) d\tau_1 \cdot \int_{\tau_2=0}^{T-t_{s,r}} f_{r,d}(\tau_2) d\tau_2 \\ &= \left(1 - e^{-\lambda_{s,r} t_{s,r}}\right) \cdot \left(1 - e^{-\lambda_{r,d}(T - t_{s,r})}\right). \end{aligned} \quad (6)$$

With the consideration of both direct and indirect contacts between u_s and u_d, the probability of forwarding a packet from u_s to u_d is

$$p_{s,d}(t \le T) = 1 - (1 - P_{s,d}(t \le T)) \prod_{\substack{u_r \in \mathbb{U} \\ r \ne s,d}} \left(1 - P^r_{s,d}(t \le T)\right). \quad (7)$$

Then, we have

$$p_{s,d}(t \le T) \ge 1 - e^{-\lambda_{s,d} T} \cdot \prod_{\substack{u_r \in \mathbb{U} \\ r \ne s,d}} \left(1 - p^r_{s,d}\right) \quad (8)$$

where $p^r_{s,d} = \left(1 - e^{-\lambda_{s,r} t_{s,r}}\right) \cdot \left(1 - e^{-\lambda_{r,d}(T - t_{s,r})}\right)$. Since $0 \le 1 - p^r_{s,d} \le 1$, where $u_r \in \mathbb{U}$ and $r \ne s, d$, $p_{s,d}$ become smaller when multiplied by more items, such as $1 - p^r_{s,d}$.

If multiple relay users are selected for the packet forwarding, the probability of multihop packet delivery in time period T can be

$$P^{r \ldots r'}_{s,d}(t \le T) = \int_0^T f_{s,r}(t) \otimes \ldots \otimes f_{r',d}(t) dt. \quad (9)$$

With multiple communities, the probability that u_s forward the packet to u_d can be calculated as

$$\begin{aligned} \mathcal{P}_{s,d}(t \le T) &= 1 - \prod_{i \in \mathbb{CC}_{s,d}} (1 - p_{s,d}(t \le T, i)) \\ &\ge \max_{i \in \mathbb{CC}_{s,d}} \{p_{s,d}(t \le T, i)\}. \end{aligned} \quad (10)$$

It is larger than the probability that u_s forward the packets within only one community. Therefore, the PIF selects the filter holders as the users who have large number of common communities with the filter creator.

B. Coarse-Grained Filtering

To achieve the security goals, the coarse-grained filtering for PIF consists of initialization, filter generation, filter distribution, and filtering as follows.

1) *Initialization*: TA bootstraps the system and assigns secret keys to individual users. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two additive cyclic groups. They have the same order q, and $\mathbb{G}$'s generator is P. Note that q a large prime. A bilinear pairing [31] exists between $\mathbb{G}$ and $\mathbb{G}_T$ is e: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. We have $e(xP, yP) = e(P, P)^{xy}$, where x and y are randomly selected from $\mathbb{Z}^*_q$. A key generation algorithm $\mathcal{G}$ takes as input a security parameter $\mathbf{k}$, and outputs (q, $\mathbb{G}$, $\mathbb{G}_T$, P, e, H_1), where H_1 is a trapdoor hash function $\mathsf{H}_1 : \{0,1\}^* \longrightarrow \mathbb{Z}^*_q$.
2) *Filter generation*: The filter creator u_i selects his keywords $W_{i,1}, \ldots, W_{i,K}$, where $1 \le k \le K$, and establishes a keyword list $\mathcal{W}_i$. Note that $K \subseteq \mathbf{K}$ which is the keyword space of the whole MSN. Every keyword is semantically defined by the TA. Then, u_i randomly picks $x_i \in \mathbb{Z}^*_q$, and computes $\mathsf{PK}_i = \frac{1}{x_i} P$ as his public key. x_i is his secret key SK_i. For a specific keyword $W_{i,k}$ (e.g., "Health"), the filter $\mathcal{F}_{i,k} =< \varphi_0, \mathcal{W}_{i,k} >$. Here, $\varphi_0 = e(P, \mathsf{PK}_i)$ and $\mathcal{W}_i = \frac{\mathsf{H}_1(W_{i,k})}{x_i + \mathsf{H}_1(W_{i,k})} P$. The keyword filter set for u_i is $\mathcal{F}_i = (\mathcal{F}_{i,1}, \ldots, \mathcal{F}_{i,k})$.
3) *Filter Distribution*: If u_i meets another user u_j, they first authenticate each other and privately compare with their profiles to determine the number of their common communities (as discussed in Section IV-A). We adopt privacy-preserving profile matching scheme in [32] to enable users to learn their common communities. If the number of their common communities is larger than a threshold TH, u_i can send his filter $\mathcal{F}_i$ to u_j as the filter holder.
4) *Filtering*: A packet sender u_s wants to delivers a packet including keywords $(W_{s,1}, \ldots, W_{s,x})$ to u_i. When u_s meets u_j, u_j helps u_i to determine if the packet from u_s can be delivered or not.
   For the keyword $W_{s,x}$, u_s sends u_j $\varphi_s = \mathsf{PK}_i + \varphi_1$. Note that $\varphi_1 = \frac{1}{\mathsf{H}_1(W_{s,x})} P$. Then, u_j checks if $\varphi_0 = e(\varphi_s, \mathcal{W}_i)$ holds. If it holds, the keyword $W_{s,x}$ matches u_i's filter such that this packet should be forwarded. When there are multiple keywords in the packet from u_s to u_i, u_j discards u_s's packet if none of the keywords associated with u_s matches u_i's filter.

Note that these steps can help the filter holders to check the packet's keyword matching in a coarse-grained manner

Algorithm 1. Social-assisted Coarse-grained Filtering

Procedure: Social-assisted Filtering
u_s wants to send a packet including keyword $W_{s,x}$ via u_j to u_i
if u_j has u_i's filters **then**
  u_j checks if the keyword in the packet is valid or not
  u_s sends $\varphi_s = \mathsf{PK}_i + \varphi_1$ to u_j
  u_j computes $e(\varphi_s, \mathcal{W}_i)$
  if $e(\varphi_s, \mathcal{W}_i) = \varphi_0$ **then**
    u_s forwards the packet to u_j
  else
    u_j discards u_s's the packet, and informs u_s
  end if
else
  u_s forwards this packet to u_j
end if
end procedure

(i.e., coarse-grained filtering). The details of the coarse-grained filtering scheme are illustrated in Algorithm 1.

Since $e(P, \mathsf{PK}_i) = e(P, \frac{1}{x_i}P) = e(P,P)^{\frac{1}{x_i}}$, and

$$\begin{aligned} &e(\widetilde{\varphi}_s, \mathcal{W}_i) \\ &= e\left(\mathsf{PK}_i + \varphi_1, \frac{\mathsf{H}_1(W_{i,k})}{x_i + \mathsf{H}_1(W_{i,k})}P\right) \\ &= e\left(\frac{1}{x_i}P + \frac{1}{\mathsf{H}_1(W_{s,x})}P, \frac{\mathsf{H}_1(W_{i,k})}{x_i + \mathsf{H}_1(W_{i,k})}P\right) \\ &= e\left(\frac{x_i + \mathsf{H}_1(W_{s,x})}{x_i\mathsf{H}_1(W_{s,x})}P, \frac{\mathsf{H}_1(W_{i,k})}{x_i + \mathsf{H}_1(W_{i,k})}P\right) \\ &= \begin{cases} e(P,P)^{\frac{1}{x_i}}, & \text{if } W_{i,k} = W_{s,x} \\ \text{random}, & \text{otherwise} \end{cases} \end{aligned}$$

when two keywords match, $\varphi_0 = e(\widetilde{\varphi}_s, \mathcal{W}_i)$. If the keywords are not the same, $e(\widetilde{\varphi}_s, \mathcal{W}_i)$ is random.

C. Fine-Grained Filtering

Although the coarse-grained keyword-based filter can block a portion of packets when matching keywords, users may want to personalize their filters due to their own preferences. It is necessary to provide a fine-grained filtering solution. The filter creator can define various levels of his interests corresponding to the specific keyword, and allow the filter holders to fine-grained filter the packets. To this end, we develop a variant of hidden vector encryption technique [33], [34] in the PIF scheme to achieve the fine-grained [35] spam filtering.

The filter creator u_i generates his fine-grained keyword filter as a vector $\mathbf{w} = (w_1, \ldots, w_l) \in \{1, \ldots, n\}^l$ to indicate his interest degree in specific keyword. A high w_l means that u_i is likely interested in the lth keyword. Denote $\sigma^*(\mathbf{w}) = \sigma^*_{a,b} \in \{1, *\}^{nl}$ by

$$\sigma^*_{a,b} = \begin{cases} 1, & \text{if } w_a = b \\ *, & \text{otherwise.} \end{cases}$$

Let $f(\sigma^*(\mathbf{w}))$ be the set of all index k such that $\sigma^*_k \neq *$, where $k \in \{1, \ldots, nl\}$.

When the sender u_s wants to send a packet with keyword information W', u_s builds the encryption vector $\sigma(\mathbf{w}') = \sigma_{a,b} \in \{0,1\}^{nl}$ for $\mathbf{w}' = (w'_1, \ldots, w'_l) \in \{1, \ldots, n\}^l$ as

$$\sigma_{a,b} = \begin{cases} 1, & \text{if } w'_a \geq b, \\ 0, & \text{otherwise.} \end{cases}$$

Here, $a \in \{1, \ldots, l\}$ and $b \in \{1, \ldots, n\}$.

For example, let $l = 3, n = 4$, and $\mathbf{w} = (1, 3, 1)$. The vector $\sigma^*(\mathbf{w}) = (1 * **, * * 1*, 1 * **)$, indicating that the matching condition with another vector $\mathbf{w}'$ is $P = (w'_1 \geq 1) \bigwedge (w'_2 \geq 3) \bigwedge (w'_3 \geq 1)$. When the encryption vector $\mathbf{w}' = (2, 3, 1)$. $\sigma(\mathbf{w}') = (1100, 1110, 1000)$. Therefore, the two vectors are matched.

Define $P(\sigma^*(\mathbf{w}), \sigma(\mathbf{w}'))$ as the predicate function as

$$P(\sigma^*(\mathbf{w}), \sigma(\mathbf{w}')) = \begin{cases} 1, & \text{if for all } i \in f(\sigma^*(v)) \\ & \sigma^*(w_a) = \sigma(w'_a) \\ 0, & \text{otherwise.} \end{cases}$$

If $P(\sigma^*(\mathbf{w}), \sigma(\mathbf{w}')) = 1$, u_j can forward the packet to u_s. We consider "$\geq$" predicate in this paper. The proposed scheme can be easily extended to "$\leq$" and some other predicates. It is also possible to combine different predicates.

Based on the above predicate, we propose a fine-grained filtering scheme to preserve the sender's keyword vector.

1) Initialization: Define $\mathbb{G}_1$ and $\mathbb{G}_2$ as the two multiplicative cyclic groups having the same order q, which is also a large prime. $\mathbb{G}_1$'s generator is g. Let e: $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be a bilinear pairing such that $e(g^a, g^b) = e(g,g)^{ab}$ for any random numbers $a, b \in \mathbb{Z}^*_q$. A bilinear key generation algorithm $\mathcal{G}$ takes as input the security parameter $\mathbf{k}$, and outputs $(q, \mathbb{G}_1, \mathbb{G}_2, g, e)$.

TA selects random elements g_1, g_2, $(h_1, u_1, \psi_1), \ldots, (h_{nl}, u_{nl}, \psi_{nl}) \in \mathbb{G}_1$, and some random numbers $y_1, y_2, v_1, \ldots, v_{nl}, t_1, \ldots, t_{nl} \in \mathbb{Z}^*_q$. Let $Y_1 = g^{y_1}$, $Y_2 = g^{y_2}$, $V_k = g^{v_k} \in \mathbb{G}_1$, and $T_k = g^{t_k} \in \mathbb{G}_1$ for $t, k \in (1, \ldots, nl)$. $\Gamma = e(g_1, Y_1)e(g_2, Y_2) \in \mathbb{G}_2$.

The public and secret keys are (PK, SK) as

$$\begin{aligned} \text{PK} &= (g, Y_1, Y_2, (h_1, u_1, \psi_1, V_1, T_1), \ldots, \\ &\quad (h_{nl}, u_{nl}, \psi_{nl}, V_{nl}, T_{nl})) \\ \text{SK} &= (g_1, g_2, y_1, y_2, v_1, \ldots, v_{nl}, t_1, \ldots, t_{nl}). \end{aligned}$$

2) Filter Generation: u_i builds his fine-grained filter $\mathbf{w} = (w_1, \ldots, w_l) \in \{1, \ldots, n\}^l$, and maps it to vector $\sigma^*(\mathbf{w})$. Then, $\sigma^*(\mathbf{w})$ is sent to u_j with the encryption of AES, when they are encountered.

u_j decrypts $\sigma^*(\mathbf{w})$ from u_i and secretly keeps it. Then, u_j selects two random numbers $\alpha, \beta \in \mathbb{Z}^*_q$, and picks random tuples $< \mu_a, \phi_a, \theta_a, \delta_a > \in \mathbb{Z}^*_q$, such that $\mu_a y_1 + \phi_a y_2 = \alpha$ and $\theta_a y_1 + \delta_a y_2 = \beta$ for all $a \in f(\sigma^*(\mathbf{w}))$.

Then, u_j computes the filter $\mathsf{F}(\sigma^*(\mathbf{w}))$ as

$$F_1 = g_1 \prod_{a \in f(\sigma^*(\mathbf{w}))} \left(h_i u_i^{\sigma^*(w_a)}\right)^{\mu_a} \psi^{\theta_a}$$

$$F_2 = g_2 \prod_{a \in f(\sigma^*(\mathbf{w}))} \left(h_i u_i^{\sigma^*(w_a)}\right)^{\phi_a} \psi^{\delta_a}$$

$$F_3 = g^{\alpha}, F_4 = g^{\beta}, F_5 = g^{-\sum_{a \in f(\sigma^*(\mathbf{w}))} (v_i \alpha + t_i \beta)}.$$

3) Filtering: u_s first generates ciphertext with the keyword related vector $\sigma(\mathbf{w}'_s)$. Then, u_s encrypts $\sigma(\mathbf{w}'_s)$ by using u_j's public key PK. u_s also picks two random numbers ρ_1 and $\rho_2 \in \mathbb{Z}_q^*$, and sends the ciphertext CT=$(C_1, C_2, C_{3,1}, \ldots, C_{3,nl}, C_{4,1}, C_{4,nl}, C_5, C_6)$, where

$$\begin{aligned}
&C_1 = Y_1^{\rho_1}, C_2 = Y_2^{\rho_1}\\
&C_{3,1} = \left(h_1 u_1^{w'_a}\right)^{\rho_1} V_1^{\rho_2}\\
&\cdots\\
&C_{3,nl} = \left(h_{nl} u_{nl}^{w'_a}\right)^{\rho_1} V_{nl}^{\rho_2}\\
&C_{4,1} = \psi_1^{\rho_1} T_1^{\rho_2}\\
&\cdots\\
&C_{4,nl} = \psi_{nl}^{\rho_1} T_{nl}^{\rho_2}\\
&C_5 = g^{\rho_2}, C_6 = \Gamma^{\rho_1}.
\end{aligned}$$

Having CT from u_s, u_j aggregates $C'_3 = \prod_{a \in f(\sigma^*(\mathbf{w}))} C_{3,a}$ and $C'_4 = \prod_{a \in f(\sigma^*(\mathbf{w}))} C_{4,a}$. u_j collects the indexes of keyword passed the coarse-grained keyword filtering, and checks

$$\frac{e(F_1, C_1)e(F_2, C_2)}{C_6} \stackrel{?}{=} e(F_3, C'_3)e(F_4, C'_4)e(F_5, C_5). \quad (11)$$

If (11) holds, u_j forward the packet to u_i; otherwise, u_j discards it.

The correctness of fine-grained filtering is as follows:

$$\begin{aligned}
&e(K_1, C_1)e(K_2, C_2)\\
&= e\left(g_1 \prod_{a \in f(\sigma*(\mathbf{w}))} (h_a u_a^{w_a})^{\mu_a} \psi^{\theta_a}, g^{y_1 \rho_1}\right)\\
&\quad \cdot e\left(g_2 \prod_{a \in f(\sigma*(\mathbf{w}))} (h_a u_a^{w_a})^{\phi_a} \psi^{\delta_a}, g^{y_2 \rho_1}\right)\\
&= \Gamma^{\rho_1} \prod_{a \in f(\sigma*(\mathbf{w}))} \left[e((h_a u_a^{w_a})^{\mu_a}, g^{y_1 \rho_1}) e((h_a u_a^{w_a})^{\phi_a}, g^{y_2 \rho_1})\right]\\
&\quad \prod_{a \in f(\sigma*(\mathbf{w}))} \left[e((\psi^{\theta_a}, g^{y_1 \rho_1}) e(\psi^{\delta_a}, g^{y_2 \rho_1})\right]\\
&= \Gamma^{\rho_1} \prod_{a \in f(\sigma*(\mathbf{w}))} e((h_a u_a^{w_a})^{\rho_1}, g^{\mu_a y_1 + \phi_a y_2})\\
&\quad \prod_{a \in f(\sigma*(\mathbf{w}))} e(\psi^{\rho_1}, g^{\theta_a y_1 + \delta_a y_2})\\
&= \Gamma^{\rho_1} e\left(\prod_{a \in f(\sigma*(\mathbf{w}))} (h_a u_a^{w_a})^{\rho_1}, g^{\alpha}\right) e\left(\prod_{a \in f(\sigma*(\mathbf{w}))} \psi^{\rho_1}, g^{\beta}\right)\\
&\quad \cdot e(K_3, C'_3)e(K_4, C'_4)e(K_5, C_5)\\
&= e\left(g^{\alpha}, \prod_{a \in f(\sigma*(\mathbf{w}))} (h_a u_a^{w'_a})^{\rho_1} g^{v_a \rho_2}\right)\\
&\quad \cdot e\left(g^{\beta}, \prod_{a \in f(\sigma*(\mathbf{w}))} \psi^{\rho_1} g^{t_a \rho_2}\right)
\end{aligned}$$

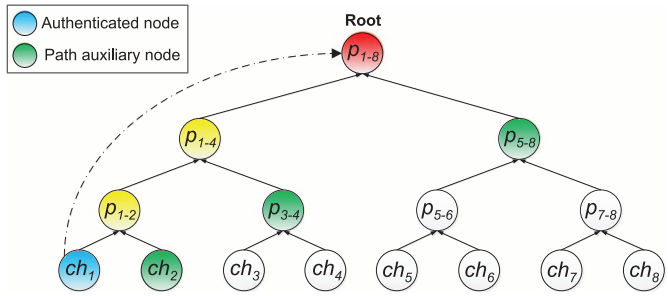


Fig. 4. Merkle Hash tree-based filter authentication.

$$\begin{aligned}
&e\left(g^{-\sum_{a \in f(\sigma*(\mathbf{w}))} (v_a \alpha + t_a \beta)}, g^{\rho_2}\right)\\
&= e\left(g^{\alpha}, \prod_{a \in f(\sigma*(\mathbf{w}))} (h_a u_a^{w'_a})^{\rho_1}\right) e\left(g^{\beta}, \prod_{a \in f(\sigma*(\mathbf{w}))} \psi^{\rho_1}\right)\\
&e(g^{\rho_2}, \prod_{a \in f(\sigma*(\mathbf{w}))} g^{v_a \alpha + t_a \beta}) e\left(g^{-\sum_{a \in f(\sigma*(\mathbf{w}))} (v_a \alpha + t_a \beta)}, g^{\rho_2}\right)\\
&= \begin{cases}
e\left(g^{\alpha}, \prod_{a \in f(\sigma*(\mathbf{w}))} (h_a u_a^{w'_a})^{\rho_1}\right) e\left(g^{\beta}, \prod_{a \in f(\sigma*(\mathbf{w}))} \psi^{\rho_1}\right)\\
\text{if } \mathbf{w} \text{ matches } \mathbf{w}', \text{i.e., } w_i = w'_i \text{ for all } a \in f(\sigma*(\mathbf{w}))\\
\\
\perp\\
\text{Otherwise.}
\end{cases}
\end{aligned}$$

Note that $C_6 = \Gamma^{\rho_1}$. If $\mathbf{w}$ matches $\mathbf{w}'$, it passes the fine-grained filter check, so that the packet from u_s can be forwarded to u_i.

D. Filter Authentication and Update Scheme

In this section, we exploit Merkle Hash tree [36] (i.e., a tree structure of cryptographic Hash functions) to authenticate each filter. We propose the construction of Hash tree for filters with the filter authentication.

Merkle Hash tree has a typical binary tree structure including 2^{N-1} leaf nodes. The depth of Merkle tree is N [37]. A parent node $p_{i-j} = \mathsf{H}(ch_i || ch_j)$ is computed by a one-way Hash function taking as input its children nodes. In Fig. 4, given the leaf nodes ch_1 and ch_2, the parent node $p_{1-2} = \mathsf{H}(ch_1 || ch_2)$ as shown. Similarly, p_{1-4} is computed by concatenating p_{1-2} and p_{3-4}. The root node $r_{1-8} = \mathsf{H}(p_{1-4} || p_{5-8})$. Let $\mathcal{PH}_1 = \{ch_2, p_{3-4}, p_{5-8}\}$ be the path from the leaf node ch_1 to the root r_{1-8}. $\mathcal{PH}_1$ can be used to authenticate the leaf node ch_1.

In the PIF, the filter creator u_i builds his keyword list $\mathcal{W}_i = \{W_{i,1}, \ldots, W_{i,K}\}, 1 \le k \le K$. Each keyword is located in the leaf of Merkle Hash tree $\mathcal{FR}_{u_i}$. In the authentication, the path $\mathcal{PH}_k$ of $W_{i,k}$ is the certificate of the keyword $W_{i,k}$. The verifier checks if the concatenated hash value of $\mathcal{PH}_k$ equals the root R_i or not. If not, the keyword is forged. Suppose there are 2^N leaf nodes in a Merkle Hash tree, users perform N Hash operations to verify each keyword (leaf node). The size of filter's signature is $N \times L$. Note that L denotes the length of each Hash value. For example, in SHA-256, L is 256 bits.

Algorithm 2. Filter Update Check

Procedure: Filter Update Check
u_i changes his keyword $W_{i,k}$, and constructs a new filter tree $\mathcal{FR}'_{u_i}$ with the root node R'_{u_i}.
u_i meets his filter holder u_j.
if u_j has u_i's keyword $W_{i,k}$ **then**
  u_j sends R_{u_i} to u_i for the authentication.
  if R_{u_i} is valid **then**
    if $R_{u_i} \neq R'_{u_i}$
      u_j searches the changed leaf nodes.
      u_i sends the updated $\mathcal{FR}'_{u_i}$ to u_j.
      u_j updates u_i's filter as $\mathcal{FR}'_{u_i}$.
    end if
  else
    u_i reports u_j to the TA since u_j forges u_i's filter.
  end if
end if
end procedure

The properties of Merkle Hash tree can also be used to check the filter's version. We propose a filter update scheme based on this property. As we presented above, the root of Merkle Hash tree changes if any leaf node varies. We do not need to check every leaf node (i.e., keyword) of the distributed filter. The filter creator u_i checks the root value R_{u_i} from his filter holder u_j for filter tree $\mathcal{FR}_{u_i}$. If the root is an existing root value, u_i sends the updated filter tree $\mathcal{FR}'_{u_i}$ to u_j as illustrated in Algorithm 2. The PIF improves the efficiency of filter search during the filter update. The Merkle Hash tree can also be extended to fine-grained filter where each value in the vector is assigned as leaf node.

V. Security Property Analysis

In this section, we discuss security properties of the PIF. We analyze the resistance to the presented attacks in Section III.

A. Resistance to Inside Curious Attack

To resist ICA, each keyword cannot be sent to others in plaintext. The PIF encrypts the creator's filters. The security of encryption is based on the collision resistant hash function and the assumption that Bilinear Diffie–Hellman Problem is computational difficult in $(\mathbb{G}, \mathbb{G}_T, e)$. Specifically, given (P, xP, yP, zP) with x, y, and z randomly selected from $\mathbb{Z}_q^*$, it is computationally infeasible to compute $e(P,P)^{xyz} \in \mathbb{G}_T$ [31]. Due to the security properties of trapdoor hash function, it is infeasible to compute $W_{i,k}$ from $\mathsf{H}_1(W_{i,k})$. Under the honest-but-curious model, the keyword is securely stored, so that the creator u_i's sensitive and private information is preserved. $\mathcal{W}_i = \frac{\mathsf{H}_1(W_{i,k})}{x_i + \mathsf{H}_1(W_{i,k})} P$. Furthermore, the filter holder u_j can efficiently check if the keyword in the packet matches any keyword in u_i's filter without disclosing $W_{i,k}$. u_j only forward the packet with appropriate keywords to u_i. In addition, the keyword index is defined by each filter creator. Different creators randomly sort the filters. If the keyword space is not large enough, u_j can take much time to exhaustively search every keyword in the keyword space. In addition, an expired time can be added into the filter, and the filter creator can update his filters timely. u_j can only guess the keyword before this expired time. After this expired time, the filter is not valid. The guess on an invalid filter cannot match any keyword within the filter, since the time stamp inside the hash function would change the output of hash value. The long guess-time can limit the ICA's attacking capability. Furthermore, the filters for different holders are set with different expired time and keyword index. Thus, multiple holders cannot collude to guess the keyword within the valid period of each filter.

In the fine-grained filtering, the keyword vector from sender is invisible to the filter holders u_j. Assume that the augmented decision linear problem [33] is computationally infeasible, u_s's private vector $\mathbf{w}'$ cannot be guessed by u_j under the selective security model. The fine-grained vector from the filter holder u_i is visible to u_j. It is a tradeoff between the fine-grained privacy of the creator and the filtering capability of holder. Fortunately, u_i can personalize his vector $\mathbf{w} = (w_1, \ldots, w_l) \in \{1, \ldots, n\}^l$. Take $n = 5$, e.g., u_i has interests in "health" with the fine-grained degree $(1, 3, 2)$ in different dimensions. In the vector, u_i can change his original fine-grained degree to build a blurred searching vector and distribute this blurred vector to a specific filter holder. Since the keyword is invisible to the filter holders, they cannot link the blurred fine-grained degree with a specific keyword. Furthermore, the packet is also encrypted by using the filter creator's public key (i.e., the destination of the packet). The filter holder cannot infer the keyword from the forwarded packet. Therefore, the filter creator's fine-grained information cannot be guessed by ICA.

B. Outside Forgery Attack

The PIF can detect the forged filters from OFA. With Merkle Hash tree, the root value is concatenated from its children nodes. Having the path information from the leaf nodes to the root, each leaf node (i.e., keyword) has a unique certificate generated by the filter creator u_i. The path information is verifiable by others. If the existing filters are changed by u_i, the new certificate is updated. However, before the filter update at u_j, the former certificate is still valid. The resilience of OFA is based on the security level of hash function used to construct the Merkle tree.

According to the above analysis, the PIF can preserve user's privacy from directly disclosing to ICA and resist the forgery attack from OFA. Note that the encountered users need to match their profiles to determine the common communities. We follow the security solution from [32] to guarantee the security and privacy requirements during profile matching. In addition, TA can receive the forgery reports from users and revoke the OFA, but does not participate in the communications. Therefore, the PIF operates in a decentralized manner from the perspective of spam filtering and security protections.

VI. Performance Evaluation

To evaluate the performance of the PIF scheme, we conduct the extensive simulation through Infocom06 trace [38].

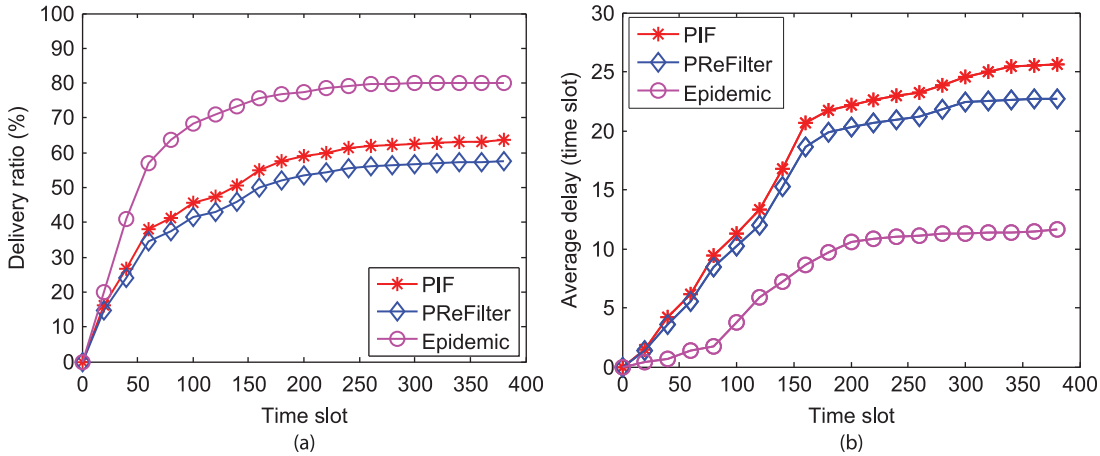


Fig. 5. Packet delivery comparison among different schemes. (a) Delivery ratio. (b) Average delay.

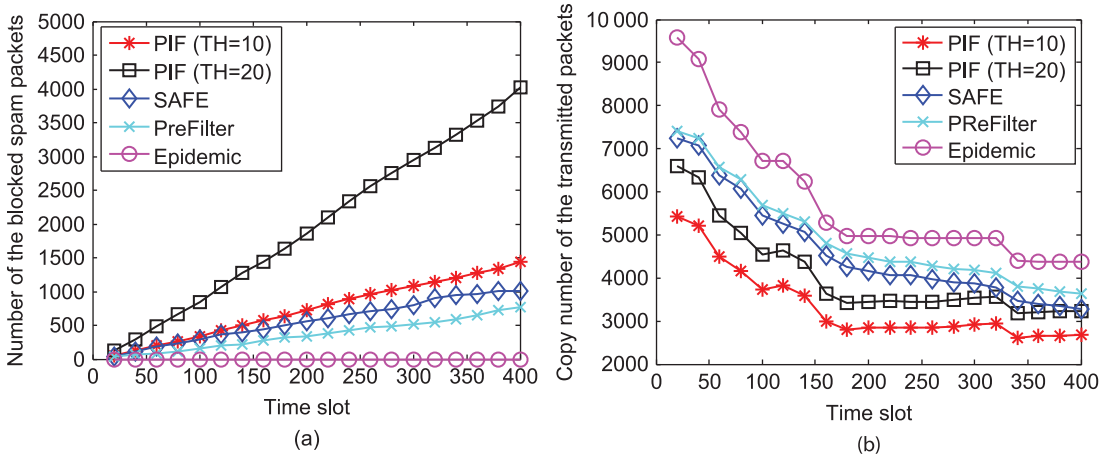


Fig. 6. Filtering comparison among different schemes. (a) Number of the blocked spams. (b) Number of copies.

A. Simulation Setup

The Infocom06 trace [38] consists of 78 mobile users during a four-day conference. Every mobile user takes a dedicated portable device to discover the nearby Bluetooth devices every 120 s. The system log records the mobile users' mobility and contact information. Totally, there are 128 979 contacts available for the simulation. We then divide the data set into two parts: the training set including one-third of the data to produce users' social relations (e.g., communities), and the simulation set including the other two-third of the data. We also leverages maximal clique to assign each user's communities. Finally, 100 communities are selected. Every community consists of a sufficient number of users, while the sum of all the edges within the community is large. In every community, there are at least 28 users. On average, every mobile users' participates in 38 communities. In the simulation, the time is divided into time slots, and each time slot represents 90 s. At the beginning of simulation, we define 100 keywords according to communities, where each user selects keywords that are associated with fine-grained interest values from $[1, 100]$ defined by users. Then, each user generates 78 packets with random keywords and interest values to different destination users every 10 time slots.

B. Simulation Results

We compare the PIF with SAFE [7], PReFilter [6], and Epidemic schemes. The PIF and SAFE have the same delivery ratio and delay, since they do not block any useful packets. Compared with PReFilter, the PIF achieves higher delivery ratio with a reasonable delay, as shown in Fig. 5(a) and (b). Epidemic scheme allows each user to send his packets to any encountered user, so that it achieves the highest delivery ratio with lowest delay. However, it costs many network resources, such as communication and storage. Note that the PIF achieves the same delivery ratio and delay with different THs (i.e., the number of common communities that both encountered users have). It is because the PIF forward packets based on the common communities with the destination. Only the number of distributed filters is impacted by TH. Therefore, the useful packets can pass the filter check and be forwarded.

In Fig. 6, we compare the PIF with SAFE and PreFilter in terms of filtering performance. From Fig. 6(a), the PIF blocks more spams compared with SAFE and PreFilter schemes, since the PIF employs find-grained filtering to effectively block the useless packets according to filter creator's defined keyword and fine-grained interests. Meanwhile, the PIF (TH = 20) filters

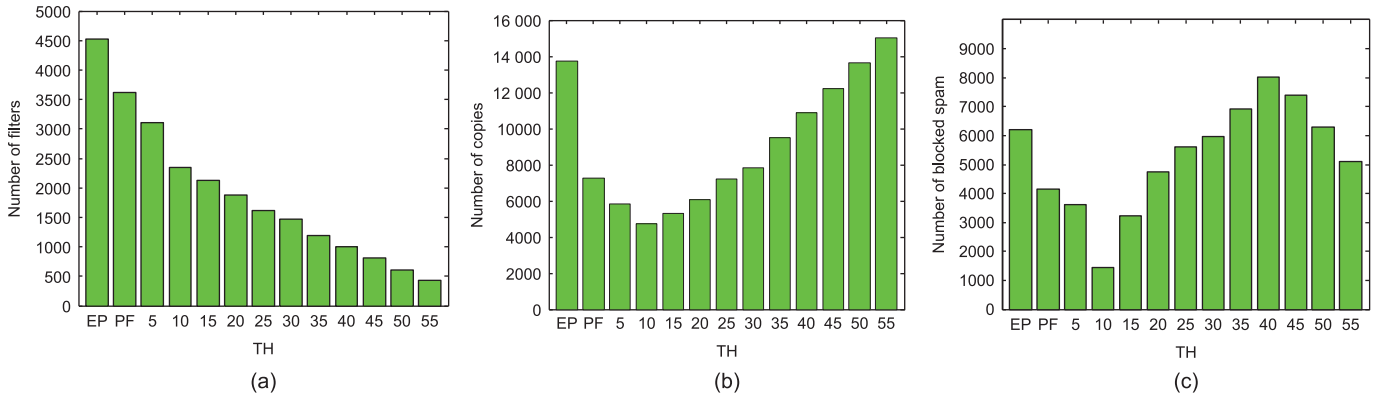


Fig. 7. Performance comparison of PIF with different THs. (a) Number of filters versus TH. (b) Number of copies versus TH. (c) Number of blocked packets versus TH.

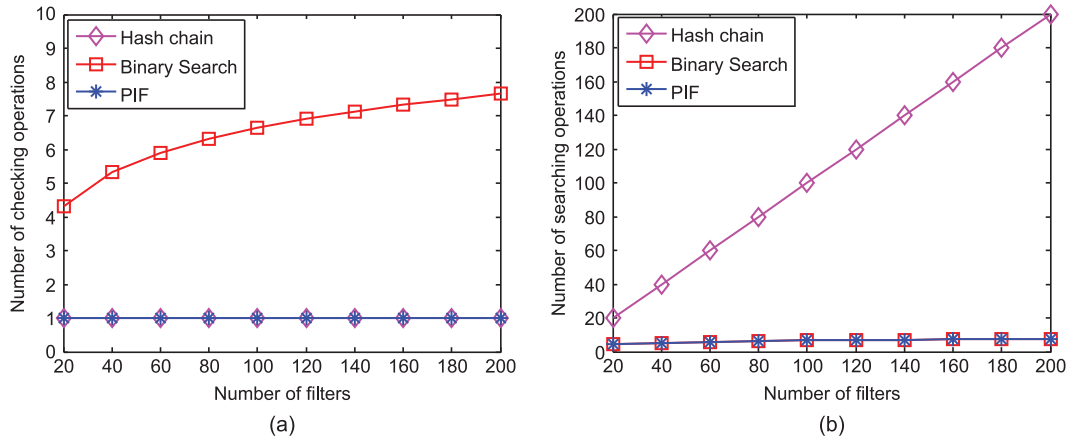


Fig. 8. Update comparison among different schemes. (a) Checking comparison. (b) Searching comparison.

more spams compared with the PIF (TH = 10). In Fig. 6(b), the PIF (TH = 10) significantly reduces the communication overheads. Although the PIF (TH = 20) blocks more spams as shown in Fig. 6(a), it still produces many copies. It is because the fewer filters are distributed in the network when TH = 20, and more users without filters may help to carry-and-forward spams. The PIF (TH = 10) can balance the tradeoff between the number of copies and the number of blocked spam packets compared with other schemes and settings.

In Fig. 7(a), when TH increases, the number of distributed filters decreases. During the filter distribution, a smaller TH leads to a larger number qualified users to hold filters. The PReFilter and Epidemic filtering schemes [i.e., PF and Ep in Fig. 7(a)] distribute too many filters to users. In the PIF, the filter creators purposely distribute their filters to the users who have more than TH common communities with the filter creators. In Fig. 7(b), a higher TH causes more copies during the packet delivery. Since the higher threshold decrease the number of distributed filters in the network, the smaller number of filters cannot filter spams well. From Fig. 7(c), we can see that the PIF with an increased TH can block more spams. When TH is small (e.g., 10 or 15), a sufficient number of users hold filters such that they do not duplicate spams. Under this circumstance, spams are filtered at sender's side. When TH increases, fewer users hold filters. The number of produced spams increases, but the number of blocked spams is also increased. With a larger TH (e.g., 45), fewer users hold filters. The spams keep increasing, but the filtering capability is degraded. In other words, the further increased TH leads to a decreasing number of blocked spams when $\mathrm{TH} > 40$. In summary, the PIF ($\mathrm{TH} = 10$) achieves the better performance to balance the number of distributed filters and copies (i.e., communication overhead), and efficiently blocks spam packets.

C. Computational Overhead

In this section, we evaluate the PIF in terms of computational complexity. Denote C_H as a Hash operation ($\{0,1\} \longrightarrow \mathbb{Z}_q^*$), C_M as a multiplication operation in $\mathbb{G}_1$ and C_p as a pairing operation. In the coarse-grained filtering scheme, the filter generation has $1 \cdot C_H + 1 \cdot C_M + 1 \cdot C_p$ operations; the filter holder checks packet sender's keyword with one pairing operation and packet sender only has one multiplication operation to protect his keyword from direct disclosing to the filter holder. For the fine-grained filtering scheme, we do not calculate the time of multiplication operations, since exponential operations take much more time than multiplication operations. Denote C_e by an exponential operation in $\mathbb{G}_1$, and $C_{e'}$ as an exponential operation in $\mathbb{G}_2$. The filter generation has $(6nl + 3) \cdot C_e$ operations. The packet sender has $(5nl + 1) \cdot C_e$ and $1 \cdot C_{e'}$ operations. Finally, the filter holder has five paring operations to check if the sender's keyword matches the filter creator's filters.

We compare the filter update complexity, as shown in Fig. 8. Filter update includes two steps: 1) check if the filters need to

be updated and 2) search the out-of-date filter. We compare the PIF with a binary search scheme and a Hash chain scheme (i.e., computing every leaf node's Hash value and checking the concatenation of all these Hash values). From Fig. 8(a), both the PIF and Hash chain schemes achieve $O(1)$ checking complexity to find if any filter should be updated. The reason is that the Merkle Hash tree-based update check only needs to check the root of the distributed filters. The binary search scheme requires an increasing number of operations when more filters are distributed, i.e., $O(\log(N))$, where N is the total number of filters. During the searching step, Hash chain scheme requires $O(N)$ searching operations, while both the PIF and binary search schemes only have $O(\log(N))$ searching complexity, as shown in Fig. 8(b). Therefore, the PIF can efficiently update the distributed filters.

VII. Conclusion

In this paper, we have proposed a personalized fine-grained spam filtering scheme with privacy preservation in MSNs. First, we have developed a filter distribution scheme based on users' common communities to efficiently distribute filters and block spams. Then, we have proposed coarse-grained and fine-grained filtering schemes with privacy preservation to enable filter creator to personalize his filters. We have also proposed a Merkle Hash tree-based filter structure, which can not only authenticate the validity of filters but also update the filters to satisfy user's various demands. The security property analysis demonstrates that filter creator's private information included in his filters can be protected from direct disclosing. In addition, we have conducted the extensive simulations to show that the PIF cannot only reduce the delay as well as the communication and storage overhead but also achieve a high filtering accuracy and efficiency. For our future work, we will investigate the self-adaptive filtering with collaboration of filter creator's social friends in MSNs.

References

[1] K. Zhang, X. Liang, R. Lu, and X. Shen, "Exploiting multimedia services in mobile social network from security and privacy perspectives," *IEEE Commun. Mag.*, vol. 52, no. 3, pp. 58–65, Mar. 2014.

[2] K. Wei, M. Dong, K. Ota, and K. Xu, "CAMF: Context-aware message forwarding in mobile social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 8, pp. 2178–2187, Aug. 2015.

[3] A. Azaria, A. Richardson, S. Kraus, and V. Subrahmanian, "Behavioral analysis of insider threat: A survey and bootstrapped prediction in imbalanced data," *IEEE Trans. Comput. Soc. Syst.*, vol. 1, no. 2, pp. 135–155, Jun. 2014.

[4] M. Hardt and S. Nath, "Privacy-aware personalization for mobile advertising," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS'12)*, 2012, pp. 662–673.

[5] F. Soldo, A. Le, and A. Markopoulou, "Blacklisting recommendation system: Using spatio-temporal patterns to predict future attacks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 7, pp. 1423–1437, Aug. 2011.

[6] R. Lu *et al.*, "PReFilter: An efficient privacy-preserving relay filtering scheme for delay tolerant networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM'12)*, 2012, pp. 1395–1403.

[7] K. Zhang, X. Liang, R. Lu, and X. Shen, "SAFE: A social based updatable filtering protocol with privacy-preserving in mobile social networks," in *Proc. IEEE Int. Conf. Commun. (ICC'13)*, 2013, pp. 6045–6049.

[8] B. Agrawal, N. Kumar, and M. Molle, "Controlling spam emails at the routers," in *Proc. IEEE Int. Conf. Commun. (ICC'05)*, 2005, pp. 1588–1592.

[9] Z. Li and H. Shen, "SOAP: A social network aided personalized and effective spam filter to clean your e-mail box," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM'11)*, 2011, pp. 1835–1843.

[10] M. Sirivianos, K. Kim, and X. Yang, "Socialfilter: Introducing social trust to collaborative spam mitigation," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM'11)*, 2011, pp. 2300–2308.

[11] K. Zhang, X. Liang, R. Lu, K. Yang, and X. Shen, "Exploiting mobile social behaviors for sybil detection," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM'15)*, 2015, pp. 271–279.

[12] M. Li *et al.*, "All your location are belong to us: Breaking mobile social networks for automated user location tracking," in *Proc. 15th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (Mobihoc'14)*, 2014, pp. 43–52.

[13] J. Kim, K. Chung, and K. Choi, "Spam filtering with dynamically updated URL statistics," *IEEE Secur. Privacy*, vol. 5, no. 4, pp. 33–39, Jul./Aug. 2007.

[14] R. Henry and I. Goldberg, "Formalizing anonymous blacklisting systems," in *Proc. IEEE Symp. Secur. Privacy*, 2011, pp. 81–95.

[15] P. Heymann, G. Koutrika, and H. Garcia-Molina, "Fighting spam on social web sites: A survey of approaches and future challenges," *IEEE Internet Comput.*, vol. 11, no. 6, pp. 36–45, Nov./Dec. 2007.

[16] A. Ramachandran and N. Feamster, "Understanding the network-level behavior of spammers," in *Proc. ACM Conf. Spec. Interest Group Data Commun. (SIGCOMM'06)*, 2006, pp. 291–302.

[17] A. Lahmadi, L. Delosières, and O. Festor, "Hinky: Defending against text-based message spam on smartphones," in *Proc. IEEE Int. Conf. Commun. (ICC'11)*, 2011, pp. 1–5.

[18] S. Hameed, X. Fu, P. Hui, and N. Sastry, "LENS: Leveraging social networking and trust to prevent spam transmission," in *Proc. Int. Conf. Netw. Protocols (ICNP'11)*, 2011, pp. 13–18.

[19] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time URL spam filtering service," in *Proc. IEEE Symp. Secur. Privacy*, 2011, pp. 447–462.

[20] H. Shen and Z. Li, "Leveraging social networks for effective spam filtering," *IEEE Trans. Comput.*, vol. 63, no. 11, pp. 2743–2759, Nov. 2014.

[21] K. Li, Z. Zhong, and L. Ramaswamy, "Privacy-aware collaborative spam filtering," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 5, pp. 725–739, May 2009.

[22] L. Fan, Z. Lu, W. Wu, B. M. Thuraisingham, H. Ma, and Y. Bi, "Least cost rumor blocking in social networks," in *Proc. IEEE Int. Conf. Devices Circuits Syst. (ICDCS'13)*, 2013, pp. 540–549.

[23] D. Shah and T. Zaman, "Rumors in a network: Who's the culprit?" *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5163–5181, Aug. 2011.

[24] Z. Wang, W. Dong, W. Zhang, and C. Tan, "Rooting our rumor sources in online social networks: The value of diversity from multiple observations," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 663–677, Jun. 2015.

[25] G. Stringhini, M. Egele, A. Zarras, T. Holz, C. Kruegel, and G. Vigna, "B@bel: Leveraging email delivery for spam mitigation," in *Proc. USENIX Secur.*, 2012, pp. 16–32.

[26] A. Balasubramanian, B. Levine, and A. Venkataramani, "Replication routing in DTNs: A resource allocation approach," *IEEE/ACM Trans. Netw.*, vol. 18, no. 2, pp. 596–609, Apr. 2010.

[27] W. Gao, Q. Li, B. Zhao, and G. Cao, "Social-aware multicast in disruption-tolerant networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1553–1566, Oct. 2012.

[28] V. Raghavan, G. Steeg, A. Galstyan, and A. Tartakovsky, "Modeling temporal activity patterns in dynamic social networks," *IEEE Trans. Comput. Soc. Syst.*, vol. 1, no. 1, pp. 89–107, Mar. 2014.

[29] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE rap: Social-based forwarding in delay-tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 11, pp. 1576–1589, Nov. 2011.

[30] K. Zhang, X. Liang, M. Barua, R. Lu, and X. Shen, "PHDA: A priority based health data aggregation with privacy preservation for cloud assisted WBANs," *Inf. Sci.*, vol. 284, pp. 130–141, 2014.

[31] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Proc. Adv. Cryptology (CRYPTO'01)*, 2001, vol. 2139, pp. 213–229.

[32] X. Liang, X. Li, K. Zhang, R. Lu, X. Lin, and X. Shen, "Fully anonymous profile matching in mobile social networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 641–655, Sep. 2013.

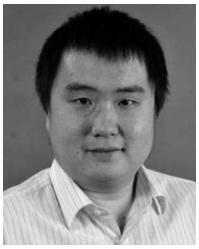
[33] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Proc. Theory Cryptography Conf. (TCC'07)*, 2007, pp. 535–554.

[34] J. Park, "Efficient hidden vector encryption for conjunctive queries on encrypted data," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 10, pp. 1483–1497, Oct. 2011.

[35] M. Wen, R. Lu, K. Zhang, J. Lei, X. Liang, and X. Shen, "PaRQ: A privacy-preserving range query scheme over encrypted metering data for smart grid," *IEEE Trans. Emerging Topics Comput.*, vol. 1, no. 1, pp. 178–191, Jun. 2013.
[36] R. Merkle, "Protocols for public key cryptosystems," in *Proc. IEEE Symp. Secur. Privacy*, 1980, pp. 122–134.
[37] Szydlo, "Merkle tree traversal in log space and time," in *Proc. Adv. Cryptology (EUROCRYPT'04)*, 2004, vol. 3027, pp. 541–554.
[38] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD trace cambridge/haggle/imote/infocom (v. 2006-01-31)," Jan. 2006.

Kuan Zhang (S'13) received the B.Sc. degree in electrical and computer engineering and the M.Sc. degree in computer science from the Northeastern University, Shenyang, China, in 2009 and 2011, respectively. Currently, he is pursuing the Ph.D. degree at the Broadband Communications Research (BBCR) Group, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada.

His research interests include the security and privacy for mobile social networks, e-healthcare system, and cloud computing.



Xiaohui Liang (S'10–M'13) received the bachelor's and master's degrees in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2006 and 2009, respectively. He received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2013.

He was a Postdoctoral Researcher with the Department of Computer Science, Dartmouth College, Hanover, NH, USA. Since 2015, he has been an Assistant Professor with the Department of Computer Science, University of Massachusetts Boston, Boston, MA, USA. His research interests include the security, privacy, and trustworthiness in medical cyber-physical systems, cyber security for mobile social networks, and applied cryptography.

Rongxing Lu (S'09–M'11–SM'15) received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2006, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2012.

Since 2013, he has been an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include the computer, network and communication security, applied cryptography, security and privacy analysis for vehicular network, e-Healthcare system, and smart grid communications.

Dr. Lu was a recipient of the Canada Governor General Gold Medal and the IEEE Communications Society Asia Pacific Outstanding Young Researcher Award in 2013.

Xuemin (Sherman) Shen (M'97–SM'02–F'09) received the B.Sc. degree from Dalian Maritime University, Dalian, China, in 1982, and the M.Sc. and Ph.D. degrees from Rutgers University, Newark, NJ, USA, in 1987 and 1990, respectively, all in electrical engineering.

He has been a Professor and University Research Chair with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He was an Associate Chair for Graduate Studies from 2004 to 2008. His research interests include the resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular *ad hoc* and sensor networks.

Dr. Shen was the Technical Program Committee Chair/Co-Chair for ACM MobiHoc'15, IEEE Infocom'14, IEEE VTC'10 Fall, the Symposia Chair for IEEE ICC'10, the Tutorial Chair for IEEE VTC'11 Spring and IEEE ICC'08, the Technical Program Committee Chair for IEEE Globecom'07. He was also the Editor-in-Chief for IEEE NETWORKS, *Peer-to-Peer Networking and Application*, and *IET Communications*. He is a Registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society. He was a recipient of the Excellent Graduate Supervision Award in 2006 and the Outstanding Performance Award in 2004, 2007, 2010, and 2014 from the University of Waterloo, the Premier's Research Excellence Award in 2003 from the province of ON; and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo.