Security and Privacy in Mobile Crowdsourcing Networks: Challenges and Opportunities

Kan Yang, Kuan Zhang, Ju Ren, and Xuemin (Sherman) Shen

The authors are with the University of Waterloo.

ABSTRACT

The mobile crowdsourcing network (MCN) is a promising network architecture that applies the principles of crowdsourcing to perform tasks with human involvement and powerful mobile devices. However, it also raises some critical security and privacy issues that impede the application of MCNs. In this article, in order to better understand these critical security and privacy challenges, we first propose a general architecture for a mobile crowdsourcing network comprising both crowdsourcing sensing and crowdsourcing computing. After that, we set forth several critical security and privacy challenges that essentially capture the characteristics of MCNs. We also formulate some research problems leading to possible research directions. We expect this work will bring more attention to further investigation on security and privacy solutions for mobile crowdsourcing networks.

INTRODUCTION

With the rapid advances in mobile and communication technologies, most mobile devices today are equipped with powerful processors, various sensors, large memories, fast wireless communication modules, and so on. Because of these sophisticated components, mobile devices have become important tools to sense, communicate, and compute data. For instance, smartphones can be used to collect video/image data (with cameras), acoustic data (with microphone), location information (with GPS), and some other useful contextual information (with gyroscopes and accelerometers). They can also transmit data via cellular networks, WiFi, Bluetooth, NFC, and so on. According to the forecasts from Canalys, worldwide mobile device shipments, including notebook PCs, tablet PCs, smart phones, and phones, will reach 2.6 billion units by 2016.

Mobile devices (e.g., smartphones, tablets, wearable devices) are usually carried by humans, so they are more applicable in a crowdsourcing environment. Crowdsourcing is the combination of two words, "crowd" and "outsourcing", coined by Jeff Howe in 2006 [1], and defined as the "*act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call.*" Although there are many advantages of crowdsourcing, the most attractive one should be that it can bring massive intelligence to solve problems at an affordable price. Some tasks that are difficult for computers or individuals can be solved efficiently by crowdsourcing to a massive group of people, including image tagging, audio translation, and so on.

The mobile crowdsourcing network (MCN) is an emerging network paradigm that captures the advantages of powerful mobile devices and crowdsourcing. It applies the principles of crowdsourcing to perform tasks with human involvement and powerful mobile devices. In MCNs, crowdsourcing is involved in both data collection and data processing: *crowdsourcing sensing* [2–4] and *crowdsourcing computing* [5, 6]. Human mobility offers unprecedented opportunities for both data sensing and transmission with mobile devices. Mobile devices can sense the surroundings wherever their holders arrive, and the storage capability of a mobile device enables it to transmit data in a store-carry-and-forward way. Moreover, human capabilities also offer intelligent human computation with their devices. Different crowdsourcing applications may utilize different human capabilities, including human perception (understanding, feeling, intuition), intelligence, cognition, knowledge, visual recognition, common sense, experiences, and so on.

Due to human involvement and crowdsourcing, several challenging security and privacy concerns are raised in MCNs. For example, some sensed data may contain location information, which may implicitly reveal a mobile user's movement. Compared to traditional networks, security and privacy issues in MCNs are more critical and challenging due to the following characteristics:

0163-6804/15/$25.00 © 2015 IEEE

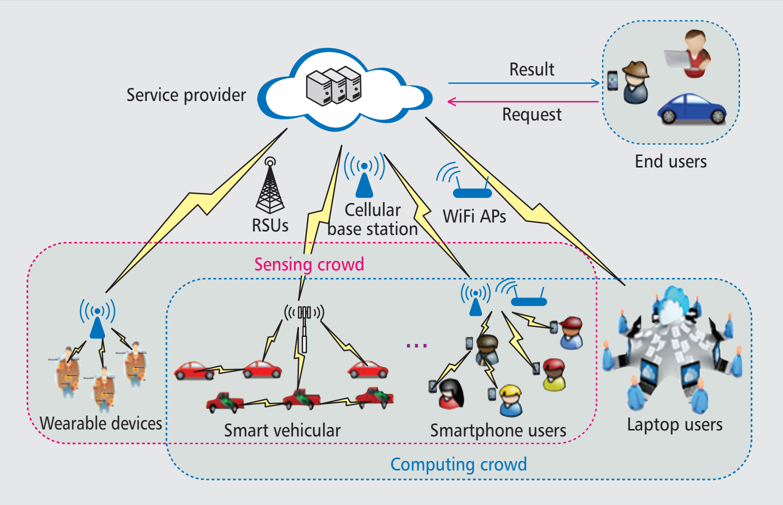
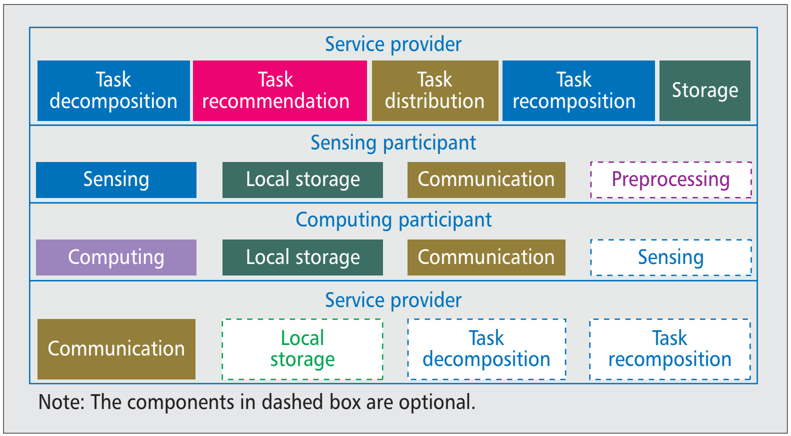


Figure 1. General architecture of mobile crowdsourcing networks.



Service provider

Task decomposition | Task recommendation | Task distribution | Task recomposition | Storage

Sensing participant

Sensing | Local storage | Communication | Preprocessing

Computing participant

Computing | Local storage | Communication | Sensing

Service provider

Communication | Local storage | Task decomposition | Task recomposition

Note: The components in dashed box are optional.

Figure 2. Components of each entity in mobile crowdsourcing networks.

Human involvement: The human is involved in both crowdsourcing sensing and crowdsourcing computing. The sensed data may contain not only sensitive information of mobile devices, but also private information of crowdsourcing participants. Moreover, mobile devices may be controlled by malicious holders to launch attacks.

Task crowdsourcing: Task crowdsourcing can raise big security concerns, especially when crowdsourced tasks themselves contain sensitive information. When crowdsourcing tasks to a dynamic group of people, it is more difficult to protect the private information than only outsourcing tasks to a single server, as the size of the group cannot be predetermined.

Dynamic topology: Mobile users may accept crowdsourced tasks based on their interests, locations, or device conditions (residual battery, available sensors, etc.). The network topology may change over time due to human mobility and dynamic user join/leave, which may also increase the difficulty of exploring security and privacy solutions.

Heterogeneity: Various communication networks may be involved in MCNs, such as wireless sensor networks, cellular networks, WiFi, Bluetooth, and vehicular ad hoc networks (VANETs). Besides, there are also a diversity of mobile devices participating in MCNs, which may produce heterogeneous data.

When participating in crowdsourcing sensing or crowdsourcing computing, mobile users consume their own resources (e.g., battery, cellular data, memory) and may suffer potential security and privacy threats. Although some incentive mechanisms [7, 8] are proposed to provide participants with enough rewards, if security and privacy cannot be guaranteed, many mobile users are still not willing to participate in and contribute to MCNs.

Aiming to address security and privacy challenges in MCNs well, in this article, we first describe several critical security and privacy challenges that capture the characteristics of MCNs. Then we point out some research problems that may lead to some possible research directions. We expect that this work will promote further investigation on security and privacy solutions for mobile crowdsourcing networks.

System Architecture of MCNs

Figure 1 illustrates a general architecture of MCNs, which includes four basic types of entities: service provider, end users, sensing crowd, and computing crowd.

Service Provider

The service provider is a crowdsourcing platform that provides crowdsourcing services to both end users and public crowds. Generally, the service provider accepts service requests from end users, and partitions these tasks into several small tasks that can be crowdsourced. It then publishes these crowdsourced tasks on its service board and waits for the crowds to finish them. Upon receiving results from those crowdsourced tasks, the service provider performs the final process and sends the final results back to end users. In some scenarios, the service provider is only responsible for publishing decomposed crowdsourcing tasks from end users if end users choose to decompose and recompose tasks by themselves.

As illustrated in Fig. 2, the service provider is equipped with several fundamental components, including the task decomposition component, the task recomposition component, the task recommendation component, the task distribution component, and the data storage component. The task requested by the end user is first decomposed by the task decomposition component and then distributed to the crowds through the task distribution component, as shown in Fig. 3. The task recomposition component is responsible for performing the final process of the results from those crowdsourced tasks. The task recommendation component enables mobile users to submit subscription trapdoors (which are constructed under subscription policies defined by mobile users) to indicate their preferences on crowdsourced tasks. Once there is a crowdsourced task, the task recommendation component will check whether this task matches these subscription trapdoors. If the task matches the subscription trapdoor of a mobile user, the service provider will send an alert to this user.

There are two basic types of crowdsourced tasks: sensing and computing. Sensing tasks are designed to collect data from a crowd of mobile users who carry sensor-enabled mobile devices. Sensing tasks return the sensed data from sensing participators to the service provider, which may be stored in storage systems managed by the service provider or sent back to end users, depending on different applications. Computing tasks are designed to crowdsource the computation to a multitude of participants with their mobile computing devices, such as mobile phones, tablets, and smart cars. Usually, computing tasks also require human intelligence and capabilities, which are denoted as human computation [9].

The service provider is usually honest but curious in the sense that it may refer some personal information from sensed data (e.g., location information, identity, interests) and may also be interested in published tasks, computing results, as well as final results sent back to end users.

End Users

End users are the customers who purchase or rent crowdsourcing services at certain costs. They send service requests to the service provider and receive results from it. As illustrated in Fig. 2, devices of end users should have basic communication components.

Under some circumstances, end users may decompose and recompose tasks by themselves and only rely on the service provider to publish their crowdsourced tasks. Together with decomposed crowdsourcing tasks, they may also provide some input data from their local storage systems or other purchased databases. In this setting, devices of end users may also be equipped with local storage components, task decomposition components, and task recomposition components, which are described by dashed boxes in Fig. 2.

Similar to the service provider, end users are usually assumed to be honest but curious when they request crowdsourcing services from the service provider. Sometimes, the end user may also participate in crowdsourced tasks published by other end users. In this setting, the security assumption of the end user should be the same as the following assumption of crowdsourcing participants.

Sensing Crowd

The sensing crowd is a crowd of mobile users who accept and participate in crowdsourced sensing tasks. In order to perform sensing tasks, as shown in Fig. 2, the sensing devices of crowdsourcing participants should have sensing components, local storage components, and communication components. The sensing components may be cameras, microphones, GPS, gyroscopes, accelerometers, and so on, and the communication component may support cellular networks, WiFi, Bluetooth, NFC, and others. Besides, some of sensing devices should also be equipped with preprocessing components if they need to conduct some preprocessing on the sensed data.

There are a large number of crowdsourced sensing tasks published every day, but mobile users may be only interested in some of tasks. In order to filter sensing tasks, mobile users can provide *sensing subscriptions*, which indicate their preferences on sensing tasks. The preferences may be affected by conditions of mobile devices, and the interests and activities of mobile users. The sensing crowd is not alway trusted. For instance, some sensing participants are malicious in the sense that they may report invalid data to the service provider or launch a distributed denial of service (DDoS) attack by accepting all the tasks without giving any results back.

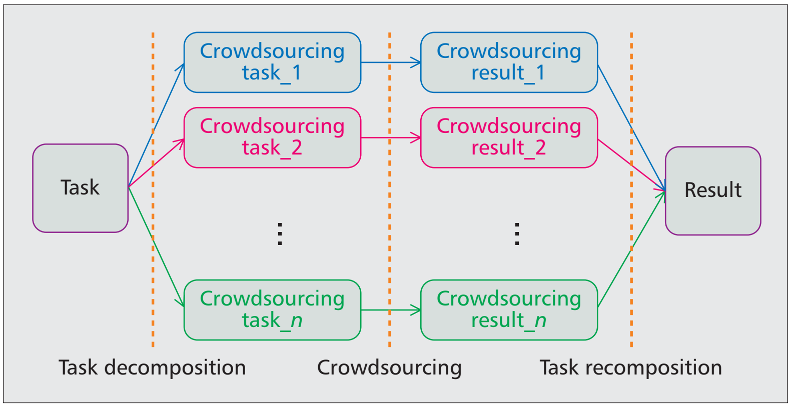


Figure 3. Task decomposition, crowdsourcing, and recomposition.

Computing Crowd

The computing crowd is a crowd of users who accept and participate in crowdsourced computing tasks. Based on the input data, computing tasks can be divided into two types.

Sensing-Based Computing Tasks: Sensing-based computing tasks take as inputs the current sensed data (e.g., the current GPS information) and the data provided by the service provider or end users (if applicable). These tasks are usually designed for people with sensor-enabled mobile devices. Different from crowdsourcing sensing, the sensed data here are directly used as computation inputs instead of being sent to the service provider.

Pure Computing Tasks: Pure computing tasks only take the data provided by the service provider (if applicable) as input. Thus, pure computing tasks can be accepted by people with any computing devices, such as mobile phones, laptops, and tablets.

As shown in Fig. 2, devices of computing participants should be equipped with computing components, local storage components, and communication components. Sensing components are also required if they accept sensing-based computing tasks. Actually, any mobile user with proper devices can be part of either sensing or computing crowds. Similar to a sensing crowd, a computing crowd may also provide *computing subscriptions*, which indicate their preferences on computing tasks. However, computing participants cannot be fully trusted either, as they may cheat and send back wrong results. Moreover, they may also be interested in the input data provided by the service provider or end users.

Due to the advantages of human involvement and powerful mobile devices, the architecture of

Application examples	Descriptions
Air pollution	Detect air pollution emitted by factories, cars, and farms.
Water quality	Monitor the water quality and study its eligibility for drinking.
Levels	Measurement of the energy radiated by cell stations and WiFi routers.
Smart navigation	Plan route according to weather conditions, accidents, and traffic jams.
Smart parking	Monitor parking space availability in the city and recommend with charges.
Smart traffic light	Control traffic lights according to traffic load and emerging events.
Health monitoring	Monitor health status from heart rate, electrocardiography, blood pressure, etc.
Disease diagnosis	Diagnose the disease from personal health parameters, and other cases.
Food recommendation	Recommend food or drinks according to personal health conditions.

Table 1. Application examples of mobile crowdsourcing networks.

MCNs can be well applied in many applications. Table 1 provides some mobile crowdsourcing application examples.

Security and Privacy Challenges in MCNs

Inspired by the characteristics of MCNs, we point out some security and privacy challenges, including privacy threats, reliability threats, and availability threats.

Privacy Threats

In MCNs, the privacy may be leaked out from both the data and the task. Here, we discuss two privacy threats: *privacy threats from data* and *privacy threats from tasks*.

Privacy Threats from Data: We first describe privacy threats by analyzing data flows in MCNs. The data flows in sensing tasks are different from the ones in computing tasks in MCNs: In sensing tasks, the data are first sensed by the sensing crowd, transmitted to the service provider, then stored in the storage system managed by the service provider or sent back to end users. In computing tasks, the input data may come from various sources, for example, the storage system managed by the service provider, local storage systems from end users, cloud storage systems purchased by end users/the service provider, sensor-enabled mobile devices, and so on. The output data of computing tasks are first sent back to the service provider and finally to the end user after the task recomposition. Based on different data flows in MCNs, the privacy threats can be summarized as follows.

- *Privacy of Sensed Data:* The sensed data may contain sensitive information of sensing participants, such as identities, location information, biometric information, and so on. For example, the location information can be easily obtained either from GPS receivers embedded in mobile devices or triangulation based on WiFi or cellular networks. Moreover, some environmental data (e.g., precise air temperature, the light, the noise, etc.) may also reveal the location information. The disclosure of location information may leak the privacy of participants, such as home and workplace locations, routines, habits, etc.
- *Privacy of Computing Inputs:* The input data of crowdsourced computing tasks may also contain sensitive information, such as business financial records, proprietary research data, or personal health information. When sending the data to the computing crowd, it may leak out the private information of data contributors, data owners or other related people.
- *Privacy of Computing Results:* The output results of crowdsourced computing tasks may be sensitive or private. End users do not want the service provider to know the contents of the results or obtain some sensitive information from the output results. In some scenarios, even computing participants are not allowed to know the contents of the computing results.

Privacy Threats from Tasks: Besides the privacy leakage from data, the task itself may also reveal some private information of both end users and crowdsourcing participants, denoted as *Task Privacy of End Users* and *Task Privacy of Participants*.

- *Task Privacy of End Users:* The content of the task may reveal sensitive information of end users to the service provider. For example, if an end user publishes crowdsourcing tasks that can only be accepted by psychologists, the service provider may infer that this end user may suffer from some psychological diseases.
- *Task Privacy of Participants:* Some tasks may also leak out private information about crowdsourcing participants. For example, if a crowdsourcing participant accepts a temperature measurement task at a particular location X at time Y, it may reveal that this participant will be at location X at time Y. However, participants may not want to leak their current location when they are tasked. In this example, the participant cannot hide the exact location information and the time when executing the task. One countermeasure is to hide their identities when taking the task and reporting results, such that the service provider only knows that some participants are at location X at time Y, but it does not know exactly who.

Reliability Threats

In MCNs, any one with a mobile device or a computing device can accept crowdsourced tasks. Due to task crowdsourcing and human involvement, it is difficult to guarantee every participant to provide reliable data or computing results.

Reliability of Sensed Data: Some malicious sensing participants may report incorrect or invalid sensed data, which is referred to as a pollution attack. Although pollution attacks exist in traditional sensor networks, it is much more challenging to detect and resist pollution attacks in MCNs due to the following reasons:

- Any adversary can be a participant in MCNs and provide pollution data.
- Powerful mobile devices can be configured by the adversary to jam specific pollution data.
- The anonymous mechanism of privacy protection also increases the difficulty of pollution attack detection.

Reliability of Computing Results: The adversary may also be a computing participant; thus, not all computing participants are honest. Malicious computing participants may provide invalid or incorrect results in order to save their computing resources. Sometimes, honest computing participants may also provide wrong results due to many reasons, for example, misunderstanding directions, or making mistakes due to personal bias or lack of experience. Because end users can only receive final results from the service provider, they can only verify whether or not final results are correct. However, it is difficult for them to identify the dishonest computing participants who provide the wrong results when final results are incorrect. Thus, the verification of the crowdsourcing result should also be performed by the service provider, which is a challenging issue as the service provider does not know the contents of the results due to privacy requirements.

Reliability of Transmission: The sensed data may be transmitted to the service provider via various channels, including 3G/4G, WiFi, VANET, or relayed by other devices. The reliability of transmission channels in these networks has been well studied in the literature. However, in MCNs, data may be tampered with by intermediated devices due to human involvement in data transmission. For example, some malicious participants may selectively discard certain data packages or modify data content.

Availability Threats

Extensive studies have been done on some denial of service (DoS) issues for traditional networks in the literature, such as network congestion due to message floods. In MCNs, however, several unique DDoS issues are raised due to task crowdsourcing:

- *DDoS by malicious participants:* Some malicious participants may accept all the crowdsourced tasks but refuse to give valid results or even ignore the tasks. This may cause the DDoS where valid crowdsourcing participants cannot get any tasks since the total number of crowdsourced tasks are usually fixed.
- *DDoS by honest but selfish participants:* A DDoS attack may also happen even when all the participants are honest. For example, a participant who is selfish and hopes to receive more rewards may accept all the computing tasks and complete them over a long time period due to its limited computation capabilities. This will also eliminate the advantage of diverse contributors in crowdsourcing as the results are from the same person.

It is challenging to prevent these DDoS attacks in MCNs because of the characteristics of MCNs, including task crowdsourcing, human involvement, and dynamic topology. To deal with these availability threats, a series of methods may be required. First, all the users should be authenticated before joining the mobile crowdsourcing network. Second, a reputation mechanism is necessary to evaluate the reliability of crowdsourcing participants and detect unreliable crowdsourcing participants. Third, novel incentive mechanisms may also be required to provide fairness in MCNs. Figure 4 briefly summarizes the security and privacy threats that capture the characteristics of MCNs.

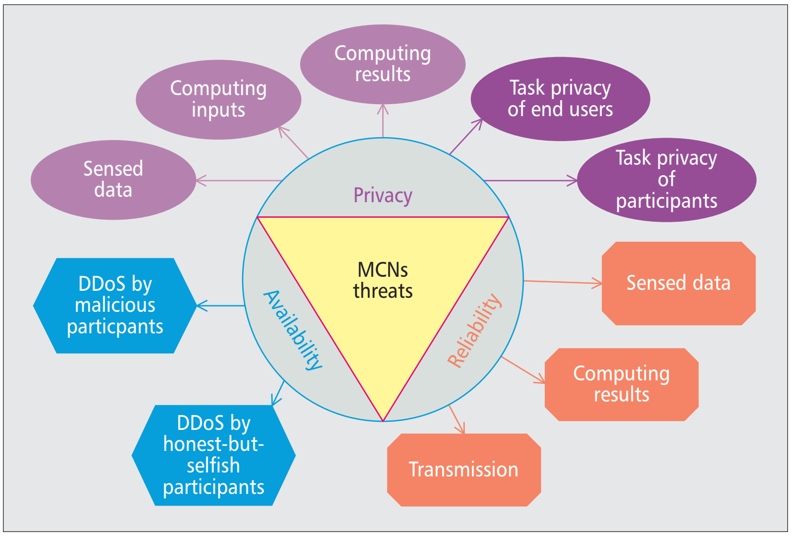


Figure 4. Security and privacy threats in MCNs.

Security and Privacy Opportunities in MCNs

There are many potential research problems from privacy threats, reliability threats, and availability threats. Some traditional user privacy concerns have been discussed in [10]. In this section, we formulate several new research problems that may provide some possible research directions.

Authentication of Crowdsourcing Participants

In order to cope with availability and reliability threats, it is necessary to authenticate crowdsourcing participants before they join the network. Normally, the authentication is conducted by the service provider. However, in some applications, the service provider is just a platform for task recommendation and distribution, and does not have the capability or privilege to authenticate crowdsourcing participants. Under this circumstance, crowdsourcing participants should be authenticated by end users. However, due to the large number of crowdsourcing participants, it is not efficient to let end users perform authentication. Therefore, a distributed authentication mechanism is desired to provide efficient and reliable authentication service for crowdsourcing participants.

One possible approach is to let the existing crowdsourcing participants authenticate new mobile users. However, some of the existing crowdsourcing participants may be malicious and collude to authenticate invalid mobile users, which means that they may let unauthorized users join the system. To avoid collusion authentication, we can utilize a threshold-based group authentication method. In a (t, n) threshold group authentication scheme, at least t existing crowdsourcing participants can act as a group authenticator, while any less than t existing crowdsourcing participants cannot authenticate new mobile users successfully. Thus, this can tolerate at most $t - 1$ malicious existing crowdsourcing participants in MCNs. However, how to efficiently initialize the system remains a challenging issue in a threshold group authentication scheme.

Another challenging problem in a (t, n) threshold group authentication scheme is the equality of authentication privilege. In other words, the newly joined crowdsourcing participant should also be able to provide group authentication to other new mobile users, together with existing crowdsourcing participants in the system. Specifically, suppose the authentication secret s of a registered mobile user is shared among n existing crowdsourcing participants. When a new mobile user becomes a crowdsourcing participant, this secret s should be shared among the new $n + 1$ crowdsourcing participants. Here comes another critical problem: how to reshare the secret from n crowdsourcing participants to the new $n + 1$ crowdsourcing participants. To solve this problem, we can refer to the method in [11], which lets each existing participant further share its secret piece into secret sectors. The new crowdsourcing participants can then reconstruct the secret sectors and obtain the new shared secret piece. However, this may incur heavy communication overhead among the crowdsourcing participants. Therefore, how to reduce the communication overhead for distributed threshold-based group authentication becomes a promising research direction.

Privacy-Preserving Task Recommendation

The task recommendation component provides a platform for task publication and subscription. Mobile users subscribe the tasks of their interests from the publishers by submitting subscription trapdoors. Due to the honest but curious service provider, the privacy issues become much more critical in task publication and subscription. There are two major concerns:

- *Task privacy*. When publishing crowdsourced tasks, end users do not want the service provider and other unauthorized participants to access their published tasks (including the corresponding input data).
- *Subscription privacy*. Mobile users also do not want the service provider to know what types of tasks that are of interest to them. Moreover, end users may hope to define the access policy of crowdsourced tasks themselves, and mobile users also hope to define the subscription policy themselves.

To protect task privacy, end users usually encrypt tasks and the input data before publishing to the honest but curious service provider, such that the service provider cannot know the contents of tasks and their data without decryption keys. The content of the subscription trapdoor, such as interests and status of mobile devices, should also be encrypted to prevent the service provider from knowing the type of tasks in which mobile users are interested. However, it is difficult to encrypt tasks or trapdoors in MCNs because of the large number of crowdsourcing participants. For example, traditional public key encryption methods require a publisher to encrypt tasks with different keys for different users. It may produce many copies of encrypted tasks in the system, the number of which is proportional to the number of mobile users. Moreover, the publisher needs to know public keys of mobile users beforehand, which is impossible in MCNs. Toward symmetric key encryption methods, the publisher needs to be always online to distribute keys. Similar problems hold for the encryption of trapdoors.

Fortunately, attribute-based encryption (ABE) [12] is a new encryption technique that only produces one copy of the ciphertext. Specifically, the task is encrypted under an access policy defined by the encryptor. The secret key of the decryptor is associated with credentials/attributes of the decryptor. The ciphertext can be decrypted only when attributes of the decryptor can satisfy the access policy in the ciphertext. Thus, we can apply attribute-based access control schemes [13, 14] to protect task privacy.

To protect subscription privacy and allow subscribers to define subscription policy, a straightforward method is to construct a subscription trapdoor by using ABE with another set of parameters. However, this requires the authority responsible for attribute management and key generation in an ABE system to generate tags for each task or subscription policy for each query. In MCNs, the tags should be generated by end users, and the subscription policy should be generated by mobile users. Therefore, efficient methods are desired to protect both task privacy and subscription privacy of task recommendation in MCNs.

In MCNs, the tags should be generated by end users, and the subscription policy should be generated by mobile users. Therefore, efficient methods are desired to protect both task privacy and subscription privacy of task recommendation in MCNs.

Privacy-Preserving Verifiable Computation Outsourcing

To protect the privacy of input data and output results of computing tasks, a straightforward method is to encrypt the input data and do the computation directly on encrypted data by using homomorphic encryption methods such that the results, when decrypted, match computation carried on unencrypted data. However, the high overhead of homomorphic encryption makes it far from applicable in practice. Thus, new methods are required to achieve computation outsourcing in MCNs.

Toward the reliability of the results from crowdsourced computing tasks, both end users and the service provider should be able to verify the correctness of computing results. But end users may not have sufficient computation resources. Thus, the ultimate goal is to design efficient verifiable computation outsourcing protocols that can minimize the computational overhead of end users. Of course, the overhead incurred by the correctness verification of the computation should be substantially smaller than running the computation itself. Another ambitious goal is to design protocols that minimize the communication overhead between end users/service provider and crowdsourcing participants.

However, some crowdsourced computing tasks contain human computation associated with human intelligence. In this case, computing results may be subjective to individual sensitivity and experience, and thus these results cannot be easily verified by verifiable computing protocols. Sometimes, crowdsourcing participants may also act honestly, but misunderstand task directions or make mistakes due to personal bias or lack of experience. Other methods may also be applied to ensure the reliability of results, such as building a reputation system, redundancy tasks, and statistical filtering. When end users are malicious, neither crowdsourcing participants nor end users can give unbiased verification results. In this context, similar to third party data integrity checking [15], a trusted third party may be employed to verify computing results.

Some crowdsourced computing tasks contain human computation associated with human intelligence. In this case, computing results may be subjective to individual sensitivity and experience, and thus these results cannot be easily verified by verifiable computing protocols.

Conclusion

The mobile crowdsourcing network is a promising paradigm in a ubiquitous computing era. In this article, we have proposed a general architecture of MCNs containing crowdsourcing sensing and crowdsourcing computing. We have outlined several critical security and privacy threats that capture the characteristics of MCNs. We have also formulated several research problems that lead to future research directions on security and privacy solutions for MCNs.

Acknowledgment

This research has been supported by a research grant from the Natural Science and Engineering Research Council (NSERC), Canada.

References

[1] J. Howe, "The Rise of Crowdsourcing," *Wired*, vol. 14, no. 6, 2006, pp. 1–4.

[2] R. K. Ganti *et al.*, "Mobile Crowdsensing: Current State and Future Challenges," *IEEE Commun. Mag.*, Nov. 2011, vol. 49, no. 11, pp. 32–39.

[3] D. Christin *et al.*, "A Survey on Privacy In Mobile Participatory Sensing Applications," *J. Systems and Software*, vol. 84, no. 11, 2011, pp. 1928–46.

[4] H. Ma, D. Zhao, and P. Yuan, "Opportunities in Mobile Crowd Sensing," *IEEE Commun. Mag.*, vol. 52, no. 8, 2014, pp. 29–35.

[5] M. Conti *et al.*, "From Opportunistic Networks to Opportunistic Computing," *IEEE Commun. Mag.*, vol. 48, no. 9, 2010, pp. 126–39.

[6] K. Parshotam, "Crowd Computing: A Literature Review and Definition," *Proc. South African Institute for Computer Scientists and Information Technologists Conference (SAICSIT '13)*; ACM, 2013, pp. 121–30.

[7] A. Singla and A. Krause, "Truthful Incentives in Crowdsourcing Tasks Using Regret Minimization Mechanisms," *Proc. 22nd Int'l. Conf. World Wide Web*, 2013, pp. 1167–78.

[8] H. Zhou *et al.*, "Consub: Incentive-Based Content Subscribing in Selfish Opportunistic Mobile Networks," *IEEE JSAC*, no. 99, 2013, pp. 1–11.

[9] A. J. Quinn and B. B. Bederson, "Human Computation: A Survey and Taxonomy of a Growing Field," *Proc. SIGCHI Conf. Human Factors in Computing Sys.*, ACM, 2011, pp. 1403–12.

[10] Y. Wang, Y. Huang, and C. Louis, "Respecting user Privacy in Mobile Crowdsourcing," *Science*, vol. 2, no. 2, 2013, pp. pp–50.

[11] K. Yang *et al.*, "Threshold Key Redistribution for Dynamic Change of Authentication Group in Wireless Mesh Networks," *IEEE GLOBECOM '10*, 2010, pp. 1–5.

[12] V. Goyal *et al.*, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," *Proc. 13th ACM Conf. Computer and Commun. Security*, New York, NY, USA: ACM, 2006, pp. 89–98.

[13] R. Lu, X. Lin, and X. Shen, "SPOC: A Secure and Privacy-Preserving Opportunistic Computing Framework for Mobilehealthcare Emergency," *IEEE Trans. Parallel Distrib. Sys.*, vol. 24, no. 3, 2013, pp. 614–24.

[14] K. Yang *et al.*, "DAC-MACS: Effective Data Access Control for Multiauthority Cloud Storage Systems," *IEEE Trans. Info. Forensics Security*, vol. 8, no. 11, 2013, pp. 1790–1801.

[15] K. Yang and X. Jia, "Data Storage Auditing Service in Cloud Computing: Challenges, Methods and Opportunities," *World Wide Web*, vol. 15, no. 4, 2012, pp. 409–28.

Biographies

KAN YANG (kan.yang@uwaterloo.ca) received his B.Eng. degree in information security from the University of Science and Technology of China in 2008 and his Ph.D. degree in computer science from the City University of Hong Kong in August 2013. He is currently a postdoctoral fellow with the Broadband Communications Research (BBCR) group in the Department of Electrical and Computer Engineering at the University of Waterloo, Canada. His research interests include cloud security and privacy, big data security and privacy, mobile security and privacy, big data mining, applied cryptography, wireless communication and networks, and distributed systems.

KUAN ZHANG (k52zhang@bbcr.uwaterloo.ca) received his B.Sc. degree in electrical and computer engineering and M.Sc. degree in computer science from Northeastern University, China, in 2009 and 2011, respectively. He is currently working toward a Ph.D. degree with the BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo. His research interests include security and privacy for mobile social networks.

JU REN (ren_ju@csu.edu.cn) received his B.Sc. and M.Sc. degrees in computer science from Central South University, China, in 2009 and 2012, respectively. He is currently a Ph.D. candidate in the Department of Computer Science at Central South University, China. From August 2013 to the present, he is also a visiting Ph.D. student in the Department of Electrical and Computer Engineering, University of Waterloo. His research interests include wireless sensor networks, mobile sensing/computing, and cloud computing.

XUEMIN (SHERMAN) SHEN (xshen@bbcr.uwaterloo.ca) is a professor and University Research Chair, Department of Electrical and Computer Engineering, University of Waterloo. He was the associate chair for Graduate Studies from 2004 to 2008. His research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He served as the Technical Program Committee Chair/Co-Chair for IEEE INFOCOM '14, IEEE VTC '10-Fall, Symposia Chair for IEEE ICC '10, Tutorial Chair for IEEE VTC '11-Spring and IEEE ICC '08, and Technical Program Committee Chair for IEEE GLOBECOM '07. He also serves or has served as Editor-in-Chief of *IEEE Network*, *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, and a Distinguished Lecturer of the IEEE Vehicular Technology and Communications Societies.