Crowdsensing Multimedia Data: Security and Privacy Issues

Yan Li  
Inha University

Young-Sik Jeong  
Dongguk University

Byeong-Seok Shin  
Inha University

Jong Hyuk Park  
Seoul National University of Science and Technology

Crowdsensing—the crowdsourcing of sensor data—allows for real-time data gathering with greater reach and accessibility than traditional crowdsourcing, because it leverages personal mobile devices as its sensor nodes (see Figure 1). This makes large-scale participatory sensing viable with little or no infrastructure cost, and because mobile device users can freely move around, coverage of sensor nodes is easily scalable.

Some crowdsensing applications also exploit the spatial properties of a problem space. In these spatial crowdsensing applications, a requester can ask for resources—which always include multimedia data—related to a specific location, and mobile users who accept the task travel to that location to obtain the multimedia data. With the ubiquity of smartphones equipped with various sensors and powerful networking capacities, such spatial multimedia crowdsensing has become increasingly prominent, but understanding the related safety and privacy concerns is extremely important when promoting participation and ensuring safe use of the data collected.

Multimedia crowdsensing activities often take place on participants’ personal devices, so sensitive information can be inadvertently revealed (such as home and work locations and the routes used when commuting). Furthermore, the crowdsourcers—that is, the companies or institutions initiating the crowdsensing—cannot entirely trust data provided by the volunteers, because personal devices can be exposed to unsafe third-party access points (such as Wi-Fi) or contaminated with malicious code (viruses and malware), which could pose challenges to the veracity of the transmitted information. Here, we identify security and privacy issues in multimedia crowdsensing and look into existing works that aim to overcome these issues.

**Security and Privacy Challenges**

Crowdsensing lets individuals with sensing and computing devices gather multimedia data to extract information that collectively forms knowledge. Multimedia crowdsensing (also called participatory sensing or mobile sensing) can be used in a wide variety of applications, including community mapping services; healthcare monitoring; and information retrieval about ambient air quality, the weather, and urban traffic patterns. These applications are broadly classified into two types: human- and environment-centric.

Human-centric applications focus on sensing information of an individual. Ubiquitous healthcare systems that gather and monitor biometric and biomechanical data of an individual are a typical example of human-centric crowdsensing applications. These systems rely on a wireless body area network (WBAN), a wireless network of wearable computing devices (sensors). WBAN is one part of the ubiquitous healthcare system, and it can collect biometric data and send it ubiquitously to cover our daily life. In a WBAN system, the sensors send collected data to the gateway node or coordinator node, which then filters, samples, and aggregates the data. The cleaned data is then sent to the service provider and medical experts.
Environment-centric crowdsensing applications exploit users’ mobile devices (such as the user’s sensor-equipped mobile phone or vehicle) to collect dynamic information about environmental trends or context information. Each mobile node gathers and processes sensor readings locally before delivering them to a central portal, where the data is stored in a database for further analysis and visualization. Examples of such systems include CarTel, GreenGPS, and VTrack.

Crowdsensing applications can also be categorized according to the types of data gathered by the sensors. Pervasive applications that require time- and location-constrained sensing collect sensor data such as the participants’ location (through the GPS) and movement (through the accelerometer) at any given time. Some crowdsensing applications gather sound information using a mobile device’s microphone and extract meaningful information by analyzing the gathered sound data. Other crowdsensing applications integrate numerous individuals’ image data (pictures and videos) of a specific location or event to gain a more comprehensive view or further refine analysis results. There are also crowdsensing applications that gather satellite images or street views of a constantly changing target to build living maps that capture a world in motion. Ubiquitous healthcare monitoring systems that collect, process, and report biometric data are also a form of crowdsensing applications. In other words, all crowdsensing applications involve multimedia (so for the rest of the article, the term “crowdsensing” indicates “multimedia crowdsensing”).

Security in crowdsensing involves ensuring the reliability of crowdsourced sensor data and protecting the privacy of participants. It’s important to guarantee that participants control when and what kind of data is released, including sensitive information. This includes the protection of data that can be inferred from both the sensor readings themselves and from user interactions in participatory sensing systems. Here, we focus on the following three challenges.

**Data Reliability**

The crowdsourcers must ensure that the sensor data provided by volunteers is reliable, although they might acknowledge some inherent uncertainties in crowdsensing. The personal device of a volunteer who takes part in crowdsensing might be inadvertently exposed to malicious code.

**Participant Privacy**

People who participate in crowdsensing have no control over the programs that are responsible for performing various activities in the crowdsourcing stages of collecting, storing, and uploading data. They are not aware of exactly what kind of information is collected from their personal device before it is uploaded to the server. Because the reports uploaded by a participant usually include...
the time and location of the sensor reading, they could reveal the participant’s location at a particular time, which could be an invasion of privacy.

**Inadvertent Data**

In crowdsensing, the data is often sensed, collected, and shared automatically without participant intervention and, in some cases, without the participant’s explicit knowledge. In particular, pervasive applications that sense the physical world can disclose sensitive personal information inadvertently. For example, a citizen’s face can be included in the sensed image data (photos and videos) in a crowdsensing application that harvests image information from an urban environment. Various techniques that filter or anonymize such sensitive information from the data before it is sent to a third party have been studied to protect privacy.

**Potential Solutions**

Here, we consider how best to address the three main challenges—that is, how to secure the safety of the crowdsensing network, protect the privacy of participants, and protect the privacy of the third party of crowdsensing networks.

**Enhancing Data Reliability**

There are many crowdsensing platforms. Some allow collaborative mapping with a social group or team, such as Ushahidi (Crowdmap), Mappler, and FindFixStreet. Platforms for web mapping that quickly and easily design maps for the web using custom data include Google Maps Engine (Google Fusion Tables), TileMill (an open source project of Mapbox), ArcGIS Online, CartoDB, GeoCommons, and GeoNode. Amazon Mechanical Turk and Microtask are software platforms for crowdsourcing and distributed work. For easier implementation, it’s best when the community mapping tool adds the authoring feature of a web-mapping tool and the task creation and allocation features of a crowdsourcing platform.

Figure 2 shows the workflow of a crowdsensing system. In the crowdsensing of spatial data, the system models sensing tasks based on a map and creates and allocates location-based tasks. In addition, the system defines the regions, time, queries, and compensation associated with the tasks and distributes the information across participating mobile devices. The participants carrying a mobile sensing device (a smartphone, tablet, or wearable device) go to the required location and perform the

---

**FIGURE 2.** A workflow of a crowdsensing system. Tasks in green are done before sending tasks to participants, tasks in orange are done after receiving the sensing data, and tasks in blue are sharing tasks.
location-based sensing tasks. They prepare a sensor data report and submit it to the server. The participants might have access to all the tasks without any restriction, or they might need to fulfill certain requirements to be able to undertake certain tasks. After performing the tasks, the mobile device or crowdsensing system processes and analyzes the sensor data to extract meaningful information, and the refined results are submitted to the server.

AnonySense is a privacy-aware framework for realizing crowdsensing applications.³ Mobile nodes that take part in crowdsensing are likely to have diverse platforms. A problem in designing a large-scale crowdsensing application is that it assumes that the crowdsourced sensor data coming from a large number of participating mobile devices is reliable and accurate. To guarantee the reliability of crowdsourced sensing tasks performed by voluntary participants, AnonySense employs four system services: Registration Authority (RA), Task Service (TS), Report Service (RS), and Mix Network (MIX).

The RA is responsible for registering nodes that wish to participate and for issuing certificates to the TS and RS so that applications and nodes can verify the authenticity of the services. The TS receives task descriptions from applications, performs some consistency checking (related to the carriers' privacy requirements and the feasibility of the task), and distributes the current tasks to mobile nodes when they ask to download new tasks. The RS receives reports from mobile nodes, aggregates them internally to provide additional privacy, and responds to queries from applications that present a token to collect their task's sensor data. The MIX serves as an anonymizing channel between mobile nodes and the report service, thus letting users send messages anonymously.

The Privacy-Enhanced Participatory Sensing Infrastructure (PEPSI) crowdsensing platform uses encryption methods⁴ to protect the sensing reports of participants and queries from users. The encrypted data is then stored in a separate area to avoid attacks from the sensing network. Only the service provider knows the decryption key, so the service provider must decrypt the text from sensing participants and encrypted queries from users and then match the users to the correct participants.

In this method, the label of sensed data is defined as the keywords of the Identity-Based Encryption (IBE) encryption method, and these keywords are then used to create a public key and decryption key. The mobile node encrypts the sensed data using the public key that was created from the report label of the sensed data. When the PEPSI infrastructure receives a query, it reconstructs the query using the private decryption key related to the query keywords. This process occurs during user authorization when the user registers the queries.

Another security challenge in crowdsensing is that the participants must upload their sensor data to third-party cloud services. A few solutions have been proposed to address this issue. One such solution proposes a middleware service that protects participants' sensor data on cloud platforms by providing resource-efficient context inferences.

However, a crowdsensing system might need a general encryption method that can accommodate a variety of multimedia data. Current research only focuses on specific data types. Different kinds of multimedia data use different kinds of encryption and decryption methods. Because many crowdsensing services are used for diverse multimedia data in mobile devices, we need to find ways to manage the security issues for the wide variety of multimedia data in a mobile environment.

Protecting Participant Privacy

In the crowdsensing platform, participants must transmit the sensed data to the service provider, or the crowdsensing network must directly collect the data. Most of the time, the sensed data is automatically transmitted to the server or network users, and the participants don't know when and what kinds of data are transmitted, although they have agreed to the automatic transmission of data.

During this process, the participants' personal information can be exposed to the data user—such as his or her phone number or location. Although the infrastructure's encryption method can protect some information—or filter out data that's easy to detect—other
data must remain open to network users. Location information is the main data that must be consistently transmitted to the service provider or network users, so in a privacy-aware crowdsensing system, location protection is a key challenge.

In particular, in addition to participant locations, the current location of users should be anonymized, and users' query privacy should be protected as well to avoid an adversary inferring the users' interests from the query content. Furthermore, when publishing trajectory data to the public for a specific service—such as for wide-range location-based data mining—individual privacy requirements of the participants should be guaranteed.

Recently, researchers have studied location privacy protection in different ways: using privacy policies, false locations, space transformations, and spatial cloaking. Several algorithms have been proposed to relax the anonymization server and use system-level encryption to guarantee location privacy. Private information retrieval (PIR) can prevent all kinds of location-based attacks, but it incurs significant computation overhead on the server side and imposes stringent requirements on the server task assignment. The Dummy-Q method generates dummy queries with different keywords from the same location, but the generated queries are sometimes unreasonable and thus easily recognized by adversaries.

To relax the reliable third-party assumption and reduce communication costs, Chi-Yin Chow and his colleagues proposed a new scheme that leverages the concept of peer-to-peer. However, peers are assumed to be reliable, and management of reliable relationships among autonomous peers in a crowdsensing system is still an open issue.

Many research works have also focused on anonymity and obfuscation-based techniques for location privacy. The $k$-anonymity and $l$-diversity are generally used in these studies. The $k$-anonymity-based location privacy methods extend a cloaking region when $k-1$ other users are included, while $l$-diversity-based location privacy methods only consider $l-1$ different locations. A spatial cloaking technique depends on $k$-anonymity and cloaking. It blurs a user's location into a cloaked spatial area so that it can satisfy the user's privacy requirements. Existing works on spatial cloaking methods are similar; they all try to blur the user's location into a specific cloaking region.

The Casper system is built on $k$-anonymity, which resides on a reliable server. It proposed using an incomplete pyramid structure to protect a user's location information, lowering both location update costs and cloaking costs. CacheCloak is another method that relies on reliable anonymization of the server architecture. It can achieve real-time location privacy protection and can guarantee the location accuracy from the location service provider.

CliqueCloak studied a personalized $k$-anonymity model, so users can adjust their privacy anonymity level, mapping it to a predefined cloaking region. Similarly, a feeling-based cloaking method uses the entropy and quad-tree to calculate a specific cloaking region. P2P-IS-HL-CA is currently the only peer-to-peer cloaking method. Its name comes from the fact that it can reduce communication overhead in P2P sensing networks using an information sharing (IS) scheme. It can also conquer the network partition problem that occurred in the historical location (HL) scheme, and it can prevent center-of-cloaked-area privacy attacks that can occur in the cloaked area (CA) adjustment scheme. However, semantic locations are not considered, so P2P-IS-HL-CA cannot prevent similarity location attacks. All of these current cloaking methods are studied to generate a cloaking area based on users' real locations, but they lack an optimal cloaking processing time.

Most research deals with anonymizing the location, provided as a geometric viewpoint (predefined map grid). SemGraph is the first research that deals with semantic cloaking with a graph-based EMD (Earth Mover’s Distance). Approaches seldom consider the problem of semantic locations, which is the nature of location-based crowdsensing.

One study, by Kuan Lun Huang, Salil Kanhere, and Wen Hu, compared micro-aggregation, a technique used for protecting privacy in databases, with tessellation. The authors proposed a hybrid scheme called \textit{hybrid variable-size maximum distance to...}
average vector (V-MDAV), which combines the positive aspects of the two techniques. They also showed that perturbing user locations with random Gaussian noise can provide users with an extra layer of protection with very little impact on system performance. In their study, when network users registered the query, the location information would change to the grid ID and would be separately stored in a map grid structure. To supply quick answers to a sensor network query, the authors used the quad-tree method to index the location data. The moving participants could thus protect their location information through the network agent, transmitting only the encrypted location data to the network users, who would receive the aggregated data but not the participants’ specific location data.

To test this method, we set up an experiment that randomly generated 10,000 virtual users moving in a closed one-floor space that was 200 × 200 m. When the virtual users randomly moved in the test space, we used a peer-to-peer method, the Casper method, the CacheCloak method, and CliqueCloak to anonymize the users’ location data and evaluate these methods’ processing time and memory usage. In a crowdsensing environment, the processing time is an important indicator for real-time location-based services. As Figure 3 shows, when the privacy level was low, performance for the four methods was similar. However, when the privacy level was high, CliqueCloak and CacheCloak were a little faster than the peer-to-peer and Casper methods.

However, cloaking methods are weak in protecting against certain attacks, such as a center of cloaked area attack. In our experiment, we measured the distance from the center of cloaked area to the real query position to evaluate the effectiveness of protecting. The query area is different from time to time, so we divided the measured distance into a query area and normalized it as a value between 0 and 100. As Figure 4 shows, the peer-to-peer method, CliqueCloak, and Casper successfully adjusted the query location away from the cloaking center. However, the peer-to-peer method was used for a wide area, so it was always far from the cloaking center.

Protecting Inadvertent Data
One of the major goals of crowdsensing is to acquire image data using camera sensors in mobile devices. However, data gathered for environmental information might contain private information, such as the face of another person or a car’s license plate number. Some companies that provide map services, such as Google Maps, have been sued because their images—captured for their street-view service—contained the faces of people. Data sharing between participants can cause unwanted exposure of third-party information. To avoid the leakage of personal information, companies must remove faces from street-view images using face-recognition techniques.

However, in crowdsensing, data is gathered and shared in real time, so critical personal information, such as a person’s face and shape, should be detected and removed in real time too. Much work has been done for real-time face recognition and face-region replacement for specific images. Face recognition in security areas contains a procedure to extract the human face region from images using several face-detection algorithms. In general, face-recognition techniques can identify the face region using features such as skin color and object symmetry and using anatomical landmarks, such as the eyes, nose, and mouth.
Research for face-region detection is categorized into one of two approaches: image-based or feature-based.\(^{14}\)

Well-known image-based face-recognition methods exploit neural networks.\(^{15}\) Using a tremendous number of face images as a training set, such methods model high-dimensional spatial distribution of important features and uses those models as criteria of determination. However, it is very difficult to apply these methods to real-time image analysis due to heavy computation. Feature-based methods detect the face region using an object’s shape information (or features).\(^{16}\) These methods create learning data in the form of an XML file through a specific classifier. Then they detect the face region by identifying similarity information in a target image with the learning data file. However, because these text-based features in an image can easily be modified, incorrect face detection is possible.

To overcome this problem, a novel face detection algorithm has been proposed that identifies the face region using an overall body silhouette rather than facial features.\(^{17}\) This method detects the face region using a new set of features, such as silhouettes of the head, neck, and shoulders, and extracting contours in a variety of directions. Even though the overall detection is comparatively high when it is applied to images for moving objects, the rate might be degraded when applied to non-static images.

Another proposal is an image-based face region detection method that uses a deep neural network to identify facial patterns.\(^{14}\) The method extracts private regions using deep visual features, such as Scale Invariant Feature Transform (SIFT) and GIST (named for getting the “gist” of a scene), and by applying deep image tags and user image tags to the image training phase. However, real-time protection of private information for acquired images is needed during the crowdsensing process. Also real-time protection should be done on mobile devices such as smartphones. Other researchers have explored fast detections of the face region\(^{16,18}\) by exploiting a minimum number of features (such as skin color) to reduce the processing time.

Xiao Hu and his colleagues proposed a method that can quickly detect the face region using only the skin color feature.\(^{19}\) The method identifies the face region using simple chrominance \(Cr\) data rather than using chrominance \(Cr/Cb\) data simultaneously on a specific position of the image. Then it detects the face region by applying the AdaBoost algorithm to determine whether a face is located in that part of the image.
Unfortunately, there have been few attempts to devise a method that can process real-time images or videos with small latency as well as be operated on mobile devices with low computing power. Furthermore, research must move beyond the facial region to explore license plates, street signs, and company logos. Deep learning could be useful in this area.

As enterprises increasingly turn to crowdsensing to efficiently and cost-effectively collect reliable data, they’ll be looking to increase participation. However, when volunteers consider whether to participate, security is a main factor. Many researchers have thus studied how to encrypt participants’ privacy and location data and users’ queries, but few have studied how to protect the multimedia data of third parties. In future work, we plan to study the development of a security-aware crowdsensing system that supports real-time protection of privacy—for participants, users, and other people whose privacy could be violated. We also hope to find better ways to support crowdsensing service providers as they detect and protect various types of multimedia objects. Furthermore, we plan to use deep learning methods to better support object detection in mobile systems.

Acknowledgments
This research was supported by the Next-Generation Information Computing Development Program through the National Research Foundation of Korea, funded by the Ministry of Science, ICT & Future Planning (no. 2012M3C4A7032781). This research was also supported by the Ministry of Science, ICT and Future Planning (MSIP), Korea, under the Information Technology Research Center support program (IITP-2016-H8601-16-1009), supervised by the Institute for Information & Communications Technology Promotion.

References


Yan Li is an assistant professor in the Department of Computer Engineering at Inha University. Her research interests include cloud computing, crowdsensing, and big data analytics. Yan has a PhD in computer science and engineering from Inha University, Korea. Contact her at leeyeon@inha.ac.kr.

Young-Sik Jeong is a professor in the Department of Multimedia Engineering at Dongguk University in Seoul. Contact him at yjeong@dongguk.edu.

Byeong-Seok Shin is a professor in the Department of Computer Engineering at Inha University, Korea. His research interests include medical imaging, volume visualization, and real-time rendering. Shin received his PhD in computer engineering from Seoul National University. He is a member of IEEE and ACM. Contact him at bsshin@inha.ac.kr.

Jong Hyuk Park (corresponding author) is a professor at the Department of Computer Science and Engineering, Seoul National University of Science and Technology (SeoulTech), Korea. His research interests include security, ubiquitous and pervasive computing, and multimedia service. Park received his PhD from the Graduate School of Information Security at Korea University. He is a member of IEEE, the IEEE Computer Society, and the Korea Information Processing Society (KIPS). Contact him at jhpark1@seoultech.ac.kr.