Towards Secure Edge-assisted Image Sharing for Timely Disaster Situation Awareness

Jing Yao[1,2], Yifeng Zheng[3], Cong Wang[1,2] and Surya Nepal[3]
[1]*City University of Hong Kong, Hong Kong, China*
[2]*City University of Hong Kong, Shenzhen Research Institute, Shenzhen, China*
[3]*CSIRO Data61 & CSCRC, Australia*

Keywords: Edge Computing, Data Security, Image Sharing.

Abstract: To save human lives and reduce injury and property loss in disasters, it is important to collect real-time situation awareness information such as the surroundings, road conditions, resource information, and more. Among others, images carry rich information and can easily provide a comprehensive view of the disaster situations. This is nowadays greatly facilitated with the prevalence of camera-embedded smartphones. However, high redundancy typically exists among the images gathered from different users during disasters. Given that bandwidth is dearer in disaster situations, it would be valuable to detect the image redundancy during transmission so that bandwidth allocation can be prioritized for unique images, enabling the timely delivery of useful information. In light of the above, in this position paper, we propose the design of an image sharing system architecture for timely disaster situation awareness. Our system architecture takes advantage of the emerging edge computing paradigm to perform image redundancy detection and prioritize the transmission of unique images, optimizing the amount of useful information delivered within a certain period of time. Meanwhile, to prevent images from being exposed to the intermediate edge infrastructure, our protocol is devised in a manner that the edge infrastructure can perform image redundancy detection without seeing the images in the clear.

1 INTRODUCTION

Rapid relief is of paramount importance to save human lives and reduce injury and property loss in disasters (earthquakes, fires, tornadoes, etc). To facilitate the rescue workers to effectively and rapidly perform the work, it is essential to collect real-time situation awareness information such as the surroundings and individuals, disaster events, road conditions, resource information, and more (Zuo et al., 2019).

Among others, images are a carrier of rich information and could easily present a comprehensive view and description of the disaster situations. With the prevalence of camera-embedded and cost-effective smartphones (Zheng et al., 2018), it is very easy for smartphone users to report and share visual information about their surroundings through photos. Indeed, it is quite common for users to take photos and share them to help reflect the situation of disasters. For example, in Typhoon Haiyan (2013), a huge amount of images had been shared by users, which had been leveraged by volunteers to explore which place is in the greatest need of help (Zuo et al., 2019). While the images shared during disasters provide valuable information for timely situation awareness, there usually exists large redundancy among the images uploaded by different users. Here, the redundancy refers to images that capture observations on the same objects/events. For example, a research study (Weinsberg et al., 2012) shows that 22% of the images taken by users during the Haiti earthquake (2010) are similar to each other, and the percentage even reaches 52% in the image set regarding the San Diego fire (2007).

Such high redundancy poses challenges on the timely delivery of useful information for situation awareness in disaster. In particular, if all images are treated equally and directly transferred to the disaster management service (DMS), it poses an obstacle for gaining unique information in real-time. A lot of resources and efforts would have to be put into the discovery of unique images. Meanwhile, the transfer of all images at the same time also imposes a heavy load on the network, which would be highly challenging given that network bandwidth is dearer in disasters. It would be much more practical to prioritize the band-

Yao, J., Zheng, Y., Wang, C. and Nepal, S.
Towards Secure Edge-assisted Image Sharing for Timely Disaster Situation Awareness.
DOI: 10.5220/0009801602950301
In *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications (ICETE 2020) - SECRYPT*, pages 295-301
ISBN: 978-989-758-446-6

width allocation on unique images. Therefore, how to properly prioritize the upload of images from different users for the DMS in disaster scenarios is of critical importance for timely situation awareness.

To support this, one promising direction is to take advantage of edge computing, an emerging paradigm which extends the capabilities of cloud computing to the network edge (NetworkWorld, 2017). Among others, one notable advantage of edge computing for disaster response is that it would allow routers, smartphones, and other devices to keep collecting data even without Internet connections (StorageCraft.com, 2020). So, with edge computing deployed between the cloud-based DMS and the users, this opens up the opportunities of giving fast responses to users and performing redundancy detection before images are uploaded to the DMS. As the edge infrastructure and the DMS could be in different trust domains and the service is set up by the DMS, a crucial requirement here is that the processing at the intermediate edge infrastructure should be done over images in the ciphertext domain, which prevents third-parties from accessing/learning about the images.

In this position paper, we propose the design of a system architecture enabling secure edge-assisted image sharing for the DMS to gain timely situation awareness in disasters. At a high level, our architecture takes advantage of the emerging edge computing architecture to perform image redundancy detection and prioritize the transmission of the unique images to the DMS, so as to optimize the amount of useful information from image sharing within a certain period of time. For redundancy detection, we mainly rely on global features extracted from images and leverage them to perform similarity measurement between different images. To support secure redundancy detection efficiently, we resort to an effective similarity search technique called locality-sensitive hashing (LSH) so that the problem of similarity measurement of encrypted images can be transformed into equality testing over the protected image features.

With this as a basis, we then propose to perform secure in-batch redundancy detection and cross-batch redundancy detection, with the goal of minimizing the redundancy per image delivery from the edge server to the DMS. Here, protected in-batch redundancy detection allows the edge server to detect image redundancy across a batch of images uploaded from different users with a certain period of time and output some candidates of representative images. And secure cross-batch redundancy detection allows the edge server to further detect image redundancy between the representative images and the images previously uploaded to the DMS via the edge server. With the synergy of these protected redundancy detection strategies, it could provide as much useful and unique information as possible per image delivery from the edge server to the DMS.

Figure 1: Overview of the system architecture.

The rest of this position paper is organized as follows. Section 2 presents our problem statement. Section 3 introduces some preliminaries. Section 4 gives the details of our design. Section 5 describes the related work. Section 6 makes conclusions and indicates the future work for this position paper.

2 PROBLEM STATEMENT

2.1 System Architecture

Fig. 1 shows the envisioned system architecture of secure edge-assisted image sharing for timely situational awareness in disasters. There are three types of actors: the mobile client, the cloud-based disaster management service (DMS), and the edge server. The mobile client is a software application which enables one to share photos captured about disaster situations to the DMS. The DMS could be setup by related organization such as the emergency organization, the local government, or a social media provider (Nishiyama et al., 2017), which wants to gather rich information rapidly so as to facilitate the arrangement of the rescue activities and decision making for disaster response. The edge server, which could be hosted by an Internet Service Provider (ISP), is located at the network edge and closer to the mobile clients. It facilitates the mobile clients to quickly share images during disasters (NetworkWorld, 2017) without the need to directly contact the remote DMS. The edge server collects the images from the mobile clients in the disaster area, and directly performs image redundancy detection so as to optimize the amount of useful information delivered from image sharing for timely situational awareness in disasters.

From practical considerations, it would be the DMS's desire to prevent the intermediate edge server from directly accessing and learning about the im-

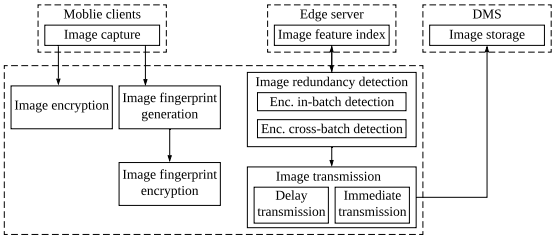

Figure 2: Service workflow.

ages, while still being able to perform the image redundancy detection. This is because the whole service is setup by the DMS, and the DMS and the edge server could belong to different trust domains. Therefore, we craft our design to support processing without disclosing the information of images. At a high level, as shown in Fig. 2, the service flow in our system is as follows. In an initialization stage, each mobile client signs up at DMS and obtains secret keys. In disaster environments, to share images, each mobile client first generates fingerprints for images , and then leverages hash function to protect the fingerprints. At last, each mobile client encrypts images. The protected fingerprints and encrypted images are then sent to the nearby edge server. Upon receiving a batch of protected fingerprints and encrypted images from possibly multiple clients within a certain period of time, the edge server will prioritize the image transmission so as to decide which images should be uploaded with high priority. Specifically, for this batch of images, the edge server leverages the protected fingerprints to do redundancy (similarity) detection, so as to select a subset of representative encrypted images. We call this procedure secure in-batch redundancy detection.

For these representative encrypted images, the edge server then performs similarity detection based on their protected fingerprints, against the set A of protected fingerprints of images previously uploaded via the edge server. We call this secure cross-batch similarity detection. For those non-similar encrypted images, the edge server places them ino a set B, and also adds their protected fingerprints to the set A. After secure cross-batch similarity detection, the edge server puts the images in B into the transmission queue, in the order of their arrival, and initiates the image transmission to the DMS. For the remaining images, the edge server puts them into another queue pending transmission, in the order of their arrival.

2.2 Threat Model

We consider that the threats in our system architecture mainly come from the engagement of the edge server. Our security goal is to protect the confidentiality of the images as well as the fingerprints, against the intermediate edge server. It is assumed to be honest-but-curious, which means that it will follow our protocol faithfully yet attempt to obtain the images. We note that this assumption is widely adopted in the literature so our adoption is consistent with prior work (Ma et al., 2019). Note that the fingerprint of an image could be used to infer some information of that image (Ferreira et al., 2017), so all the fingerprints also demand protection. Additionally, we assume that the mobile client and the DMS are fully trusted. We deem attacks like DDoS attacks and data integrity attacks out of the scope.

3 PRELIMINARIES

3.1 Image Features

Image features could provide a short summary of the content of an image, i.e., acting as a fingerprint. In general, image features can be divided into two categories: (1) local features and (2) global features. A local feature only describes an interesting point of an image. So, for an image, multiple local features need to be computed. Different from local features, a global feature can represent the whole content of an image. It is usually computed by the color histogram, texture values of an image, etc. In comparison with local features, global features are advantageous due to their high efficiency in feature extraction as well as in use for similarity detection (Chamoso et al., 2018). Therefore, we use global features to detect image similarity.

3.2 Locality-sensitive Hashing

A locality-sensitive hashing distributes a hash function family over a dataset, enabling similar data points to have hash collisions with high probability and dissimilar data to have hash collisions with little probability. It is generally used for similarity search(Zheng et al., 2017; Cui et al., 2016), with the formal definition stated as follows.

***Definition* 1.** *(Locality-sensitive Hashing) Let* F *be a hash function family and R be the range of hash results of the dataset D. The distance function is denoted as* $dis(x,y)$*, where* $x,y \in D$*. Given the two distance values* $\{(r_1,r_2)|r_1 < r_2\}$ *and the two probability values* $\{(p_1,p_2)|p_1 > p_2\}$*, the family* $\mathrm{F} = \{h : D \to R\}$ *is a* (r_1,r_2,p_1,p_2)*-locality sensitive hash function family if for any* $\{(d_1,d_2)|dis(d_1,d_2) \leq r_1 \wedge d_1,d_2 \in D\}$*, then* $Pr[h(d_1) = h(d_2)] \geq p_1$*; if*

for any $\{(d_1,d_2)|dis(d_1,d_2) > r_2 \wedge d_1,d_2 \in D\}$*, then* $Pr[h(d_1) = h(d_2)] \leq p_2$.

4 DESIGN OF SECURE EDGE-ASSISTED IMAGE SHARING IN DISASTERS

In this section, we start with the description of our design intuition on secure edge-assisted image sharing in disasters. Then, we move on to the introduction of the detailed scheme.

4.1 Design Intuition

Our main design insight is to prioritize the transmission of the unique images via fingerprint-based similarity detection at the edge server, so as to optimize the amount of useful information per image delivery for timely situational awareness in disasters. As we aim to protect the confidentiality of images and fingerprints, all the processing should be done with the prerequisite of protecting the confidentiality of images and fingerprints. To instantiate our main idea, some design considerations need to be addressed.

The first consideration is how to efficiently support similarity detection for two encrypted images. Recall that we resort to global features as fingerprints for similarity detection. The straightforward way of doing this is to compute the distance (under a common metric like Hamming distance or Euclidean distance) between the fingerprints of two images. A smaller distance means a higher similarity between the two images. If the distance is less than a predefined threshold, the two images are considered as similar images. However, such a plausible approach would demand the use of advanced and expensive cryptographic primitives like homomorphic encryption in our scenario. To avoid pairwise distance computation between the fingerprints as well as achieve high efficiency in similarity detection, we leverage a set of LSH functions to hash global features of images. According to the definition of LSH, an LSH function can map the features of similar images to identical LSH values with high probability. In this case, the similarity of any two images can be efficiently measured by counting the number of their matched LSH values.

The second consideration is on the confidentiality of the image fingerprints. As LSH does not have the one-way property from a cryptographic perspective (Partridge et al., 2012), the adversary might still be able to infer information about the image fingerprints. To preserve the confidentiality of the fingerprints, our design hashes LSH values before the mobile client's uploading them to the edge server. Considering the resource constraints of mobile devices, we take advantage of a lightweight mechanism, i.e., one-way hash, for mobile clients to protect the fingerprints. In order to preserve the characteristic of LSH that maps similar data to the identical hash result with high probability, each LSH value of the fingerprint is hashed separately. Note that the similarity detection is only based on the fingerprints so the images could be independently encrypted under standard encryption mechanisms.

The third consideration is how to properly perform secure in-batch redundancy detection and cross-batch redundancy detection. As mentioned before, secure in-batch redundancy detection refers to select a subset of representative encrypted images from a batch of encrypted images collected by the edge server over a certain period of time. In contrast, secure cross-batch redundancy detection aims to select images from the representative image set so that the selected images ware not similar to the images previously uploaded by the edge server. It is noted that most of the existing designs only consider the detection of cross-batch redundancy, such as SmartEye(Hua et al., 2015) and MRC(Dao et al., 2017). The only design that takes into account both cross-batch redundancy and in-batch redundancy is the design due to Zuo et al. (Zuo et al., 2019). Their design works under a different scenario where only the cloud is engaged. Besides, they first detect the cross-batch similar images, followed by the detection of the in-batch similar images. Given that the similar images will be transmitted in the same time period with high probability under the disaster scenario, our belief is that it would be much more efficient to first perform in-batch similarity detection and then cross-batch similarity detection. Through this delicate consideration, the number of images involved in cross-batch similarity detection would be largely reduced, leading to efficiency improvement on the cost-dominant cross-batch redundancy detection.

4.2 Our Proposed Design

We now present the detailed design for secure edge-assisted image sharing in disasters. Overall, there are three key modules, i.e., data preparation, secure in-batch redundancy detection, and secure cross-batch redundancy detection. We elaborate on each of them as follows.

Algorithm 1: Data Preparation at the Mobile Client.

Input: Image I and secret key k.
Output: Encrypted image c and protected feature $\mathbf{v}$.

1: $f \leftarrow fExtract(I)$.
2: $\mathbf{w} \leftarrow LSH(f)$, where $\mathbf{w} = \{w_1, \cdots, w_l\}$;
3: **for** $i \in \{1, \cdots, l\}$ **do**
4: $\quad v_i \leftarrow H(w_i)$.
5: **end for**
6: $c \leftarrow E_k(Im)$.

4.2.1 Data Preparation

Fig. 1 shows the data preparation module. The mobile client first calls the feature extraction function $fExtract(\cdot)$ to extract the fingerprint f from the image Im to be uploaded. Examples of global features as fingerprints include dhash and ahash(Neal Krawetz, 2013). The mobile client then applies l LSH functions to the fingerprint and generates a set $\mathbf{w}$ of LSH values. At last, the mobile client hashes $\mathbf{w}$ with a one-way hash function $H(\cdot)$ and outputs the protected fingerprint $\mathbf{v} = \{v_1, \cdots, v_l\}$. For the image I, the mobile client calls an encryption scheme $E(\cdot)$ which could be the standard AES encryption scheme, and outputs the encrypted image c. Finally, the mobile client uploads all encrypted images and protected fingerprints to the nearby edge server.

4.2.2 Secure In-batch Redundancy Detection

In disaster scenarios, it is very likely that there are similar images in the batch of images collected the edge server over a certain period of time. Therefore, we perform in-batch redundancy detection so as to select a representative image set from these images. The high-level idea is to first divide the batch of images into groups according to the similarity between each of two encrypted images in the batch. In particular, the edge server first calculates the similarity between each of any two images in the batch. Let the protected fingerprints of two images I_1 and I_2 be $\mathbf{v_1}$ and $\mathbf{v_2}$ respectively. The distance between the two images is computed via $dis(I_1, I_2) = ||\mathbf{v_1} - \mathbf{v_2}||_1$, where $||\cdot||_1$ denotes the $L1$ distance. The two images I_1 and I_2 are considered to be similar, if $dis(I_1, I_2) < \varepsilon$, where ε is a pre-defined similarity threshold. Based on this, the edge server then divides the in-batch images into groups. We use k to denote the number of groups.

According to the grouping result, the edge server then proceeds to select a set of representative images in the batch. There are several intuitive strategies that could be adopted here. The first strategy is to select from each group the image that arrives at the edge server the earliest, following the general principles of packet forwarding for routers and switches, i.e., first-in, first-out (FIFO). However, this strategy may lead to an excessive content loss, given that it is very likely that the content of the first image may not well summarize the information contained in this group of images. The second possible strategy is to select the image with the highest resolution in each group. However, this also does not guarantee that the largest image is the most informative one in the group.

In short, although these intuitive strategies are simple and relatively easy to realize, they may not produce a good representative set that could well represent the whole batch of images. We note that an alternative choice is to adapt a more advanced optimization-based strategy inspired by (Zuo et al., 2019) in our scenario, which allows to select a subset T of images that could best represent the whole batch S of images. At a high level, the selection of a subset of representative images is formulated as an optimization problem, where the objective is formulated as: $T^* \in \arg\max_{T \subset S} F(T)$. Here, F is a scoring function that can quantitatively represent the quality of a summary (i.e., the subset of representative images). The realization of the function F is a weighted sum of two sub-functions: a coverage function f_c and diversity function f_d. That is, $F(T) = \lambda_1 \cdot f_c(T) + \lambda_2 \cdot f_d(T)$, where λ_1 and λ_2 are non-negative. Specifically, the coverage function is formulated as $f_c = \sum_{i \in S} max_{j \in T} w_{i,j}$, where the intuition is to use the sum of the similarity between an image i in S and its most similar image in T. For the diversity function, it is formulated as $f_d = \sum_{i=1}^{k} N(T, I_i)$, where I_i refers to the set of images in a i-th group, and $N(T, I_i)$ takes the value 0 if T and I_i have empty intersection, and the value 1 otherwise. Solving the above optimization needs a constraint on the size of the selected subset, for which we set it to the number of groups, i.e., $|T| \leq k$.

Algorithm 2 shows the procedure of in-batch redundancy detection, which allows the edge server to select a representative subset of images from a batch of images collected over a certain period of time. The edge server first calculates the similarity between any two images in the batch. Then, it uses the similarity threshold ε to partition the batch of images into several groups. Finally, the edge server solves the optimization problem via a greedy algorithm to output a subset of representative images.

4.2.3 Secure Cross-batch Redundancy Detection

In secure cross-batch redundancy detection, we aim to further select a subset from the encrypted representative images obtained from in-batch similarity detec-

Algorithm 2: Secure In-batch Redundancy Detection at the Edge Server.

Input: The batch S of encrypted images $C_S = \{c_1, \cdots, c_{|S|}\}$ and the batch of protected features $V_S = \{\mathbf{v}_1, \cdots, \mathbf{v}_{|S|}\}$.

Output: T_n (n is the number of iterations).

1: Compute pairwise similarity $w_{i,j}$ among the images C_S in the batch S according to the batch of protected features V_S.
2: Divide the batch S of images into several groups according to the principle of whether $w_{i,j} < \varepsilon$.
3: Denote the number of groups as k.
4: Choose an image S_1 arbitrarily.
5: $T_1 \leftarrow S_1$.
6: While $|T_i| \leq k$
7: Choose $S_i \in \arg max_{S_i \in S \setminus T_i} F(T_i \cup \{S_i\})$.
8: $T_{i+1} \leftarrow T_i \cup \{S_i\}$.
9: $i \leftarrow i+1$.
10: end while

tion, ensuring that each of the selected images is not similar to any image uploaded to the DMS before via the edge server. The high-level idea of secure cross-batch redundancy detection is to query the protected fingerprints of the encrypted images previously uploaded to the DMS, and check if some encrypted representative images are similar to them.

A simple approach for the edge server is to send the protected fingerprints of all the representative images to the DMS, which then performs the redundancy detection and returns the detection result to the edge server. However, this could lead to high latency. To alleviate this problem, our design is to let the edge server build and maintain an index using efficient and succinct data structures over the protected fingerprints of images previously uploaded to the DMS. In this case, the edge server can perform cross-batch redundancy detection locally. When the edge server sends an image to DMS, it also inserts the protected fingerprint of that image to the index.

4.3 Security Guarantees

The security of our proposed design directly follows from that of the one-way hash and the image encryption scheme. Specifically, as we use a one-way hash function to hash the LSH values of image fingerprints, this makes it computationally infeasible to reveal the fingerprint from the protected fingerprints (Partridge et al., 2012). Therefore, the proposed design can protect the image fingerprints against the edge server. For the images, recall they are encrypted via either standard encryption schemes. So the confidentiality of the images is well protected against the edger server. Additionally, we remark that revealing the similarity of images in the ciphertext domain is necessary to support the target functionality of redundancy detection.

5 RELATED WORK

Our work is closely related to the existing works on content-aware redundancy elimination in disaster environments, where the ultimate goal is to ask mobile clients to only upload unique images after image redundancy detection. Some works ((Dao et al., 2017; Zuo et al., 2019), to just list a few) have proposed to perform redundancy detection at the remote server side (e.g., the cloud). In these designs, the mobile client is typically required to send all features and wait for the detection result. This might greatly reduce user experience when the database of images at the remote server side is very large, which could be the norm in disaster scenarios.

Instead of doing redundancy detection at the remote server side, some works ((Hua et al., 2015; Zuo et al., 2019; Weinsberg et al., 2012), to just list a few) have proposed to take advantage of advanced networking architectures like software-defined networking (SDN) or delay tolerant network (DTN) to perform in-network redundancy detection. Our work differs from prior work mainly from several aspects. Firstly, our design takes into account the security of images and fingerprints so the redundancy detection is all performed while protecting the confidentiality of images. Secondly, our design takes advantage of the emerging edge computing architecture to perform secure redundancy detection. With the assistance of the edge server, the mobile clients in our system architecture do not need to wait for the redundancy detection results. Thirdly, while most of the existing works only consider cross-batch redundancy detection, our design support both (secure) in-batch redundancy detection and cross-batch redundancy detection, which is customized for our scenario as well.

6 CONCLUSIONS AND FUTURE WORK

In this position paper, we explore and propose a system architecture for the DMS, enabling secure edge-assisted image sharing for timely disaster awareness. Aiming for redundancy detection as well as image security against the intermediate edge infrastructure, we propose to apply the strategies of in-batch redundancy

detection and cross-batch redundancy detection, and design from the ground up so as to do all the effective processing while ensuring protection of images. Our customized design comes from a synergy of a series of techniques including image processing, data encryption, efficient similarity search, and optimization.

As future work, we plan to implement a proof-of-concept system prototype and conduct a comprehensive evaluation over real-world datasets. Specifically, we will measure the effectiveness of our secure redundancy detection design over some real-world disaster image datasets. We will also evaluate the cost efficiency on different ends along the service flow. We also intend to define formal security definitions and provide formal proofs. Besides, we will explore the design space of emerging security solutions like trusted execution environments, for efficiently defending against malicious adversaries that compromise the edge server and deviate arbitrarily.

ACKNOWLEDGEMENTS

This work was supported in part by the Research Grants Council of Hong Kong under Grants CityU 11202419, CityU 11212717, CityU 11217819, and CityU C1008-16G.

REFERENCES

Chamoso, P., Rivas, A., Sánchez-Torres, R., and Rodríguez, S. (2018). Social computing for image matching. *PloS one*, 13(5):e0197576.

Cui, H., Yuan, X., Zheng, Y., and Wang, C. (2016). Enabling secure and effective near-duplicate detection over encrypted in-network storage. In *Proc. of IEEE INFOCOM*.

Dao, T., Roy-Chowdhury, A. K., Madhyastha, H. V., Krishnamurthy, S. V., and Porta, T. L. (2017). Managing redundant content in bandwidth constrained wireless networks. *IEEE/ACM TON*, 25(2):988–1003.

Ferreira, B., Rodrigues, J., Leitao, J., and Domingos, H. (2017). Practical privacy-preserving content-based retrieval in cloud image repositories. *IEEE TCC*, 13(9):1–14.

Hua, Y., He, W., Liu, X., and Feng, D. (2015). Smarteye: Real-time and efficient cloud image sharing for disaster environments. In *Proc. of IEEE INFOCOM*.

Ma, L., Liu, X., Pei, Q., and Xiang, Y. (2019). Privacy-preserving reputation management for edge computing enhanced mobile crowdsensing. *IEEE Trans. Services Computing*, 12(5):786–799.

Neal Krawetz (2013). Kind of Like That. http://www.hackerfactor.com/blog/?/archives/529-Kind-of-Like-That.html/.

NetworkWorld (2017). When disasters strike, edge computing must kick in. https://www.networkworld.com/article/3228884/when-disasters-strike-edge-computing-must-kick-in.html.

Nishiyama, J., Tabata, S., and Shigeno, H. (2017). An efficient image gathering scheme with quality control in disaster. In *Proc. of IEEE AINA*.

Partridge, K., Pathak, M. A., Uzun, E., and Wang, C. (2012). Picoda: Privacy-preserving smart coupon delivery architecture. In *Proc. of HotPETs*.

StorageCraft.com (2020). Edge Computing vs. Cloud Computing. https://blog.storagecraft.com/edge-computing-cloud-computing/.

Weinsberg, U., Li, Q., Taft, N., Balachandran, A., Sekar, V., Iannaccone, G., and Seshan, S. (2012). CARE: content aware redundancy elimination for challenged networks. In *Proc. of ACM HotNets*.

Zheng, Y., Cui, H., Wang, C., and Zhou, J. (2017). Privacy-preserving image denoising from external cloud databases. *IEEE Trans. Information Forensics and Security*, 12(6):1285–1298.

Zheng, Y., Duan, H., and Wang, C. (2018). Learning the truth privately and confidently: Encrypted confidence-aware truth discovery in mobile crowdsensing. *IEEE Trans. Information Forensics and Security*, 13(10):2475–2489.

Zuo, P., Hua, Y., Sun, Y., Liu, X. S., Wu, J., Guo, Y., Xia, W., Cao, S., and Feng, D. (2019). Bandwidth and energy efficient image sharing for situation awareness in disasters. *IEEE TPDS*, 30(1):15–28.