Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 2076542, 11 pages
https://doi.org/10.1155/2019/2076542

Research Article

D2D Big Data Privacy-Preserving Framework Based on (a, k)-Anonymity Model

Jie Wang,[1] Hongtao Li,[1,2] Feng Guo,[2] Wenyin Zhang,[2] and Yifeng Cui[2]

[1]*College of Mathematics & Computer Science, Shanxi Normal University, Linfen 041000, China*
[2]*School of Information Science and Engineering, Linyi University, Linyi 276000, China*

Correspondence should be addressed to Hongtao Li; lihongtao7758@163.com

Received 23 November 2018; Revised 29 May 2019; Accepted 26 June 2019; Published 7 August 2019

Academic Editor: Mariko Nakano-Miyatake

As a novel and promising technology for 5G networks, device-to-device (D2D) communication has garnered a significant amount of research interest because of the advantages of rapid sharing and high accuracy on deliveries as well as its variety of applications and services. Big data technology offers unprecedented opportunities and poses a daunting challenge to D2D communication and sharing, where the data often contain private information concerning users or organizations and thus are at risk of being leaked. Privacy preservation is necessary for D2D services but has not been extensively studied. In this paper, we propose an (a, k)-anonymity privacy-preserving framework for D2D big data deployed on MapReduce. Firstly, we provide a framework for the D2D big data sharing and analyze the threat model. Then, we propose an (a, k)-anonymity privacy-preserving framework for D2D big data deployed on MapReduce. In our privacy-preserving framework, we adopt (a, k)-anonymity as privacy-preserving model for D2D big data and use the distributed MapReduce to classify and group data for massive datasets. The results of experiments and theoretical analysis show that our privacy-preserving algorithm deployed on MapReduce is effective for D2D big data privacy protection with less information loss and computing time.

1. Introduction

Device-to-device (D2D) communications have been proposed as a promising technology for fifth generation (5G) cellular networks. It has been shown that D2D communications can improve the network performance in terms of communication capacity and delay, spectral efficiency, power dissipation, and cellular coverage [1]. In recent years, the volume of data and traffic generated over mobile networks have increased significantly with the increasing quality and quantity of available multimedia services. Users prefer to share interesting files locally using wireless short-range D2D communication.

Recent studies have shown by mining the social and mobile behaviors of users that they prefer to share content offline via D2D communication [2–4]. However, past studies on the subject have been based on the small-scale data analysis and the algorithm design that applied to specific sets of users. With the rapid growth of mobile users and devices, D2D technology should be able to adapt to the delivery of massive amount of data across a large number of users. Therefore, this paper proposes an (a, k)-anonymous D2D big data privacy-preserving framework deployed on MapReduce to provide rapid sharing, high accuracy on deliveries, efficient and intelligent delivery, and accurate content promotion to a large number of users.

From the perspective of the sharing capacity of D2D communication, big data technology offers unprecedented opportunities but also poses challenges to traditional data analysis of groups of mobile users. The dimensionality, heterogeneity, and complexity of data exacerbate the security- and privacy-related problems of D2D communication [5, 6]. D2D big data usually contain private information of a user or event. The mining, analysis, and processing of D2D big data can thus lead to leaking of private user information. There are large numbers of sensing nodes in D2D communication systems that continuously transmit a large amount of data concerning users, government departments, and national infrastructures, which often contain sensitive information. If these important data are not effectively

protected during the data mining, analysis, and processing, they can be leaked, and this can significantly harm people, organizations, and national interests. Therefore, a practical framework for privacy-preserving data analytics for D2D communication systems is needed.

Data encryption is a common method of privacy protection. Such technologies as symmetric encryption, elliptic curve encryption, and data segmentation have been developed for privacy protection during data acquisition in wireless sensor networks [7]. Based on blind signature technology, Xu et al. [7] proposed a data collection framework with privacy protection capability in smart grids based on a key distribution framework that effectively protects the user's private data. However, the above privacy protection frameworks have limitations and security defects, such as key propagation and a large overhead for encryption and decryption calculations.

In recent years, the anonymity method has become the dominant data privacy protection mechanism. Anonymity requires that an attacker is not able to match sensitive information to the body of the given data with high confidence. In D2D communication networks, anonymity operations featuring multiple participants are needed to ensure that the failure of a single principal does not affect the system. In recent years, anonymity solutions for various networks have been proposed [8], but few have been developed for complex D2D communication networks. The amount of data in D2D communication systems is massive, and it grows quickly as well. Traditional privacy-preserving methods cannot meet the security requirements of D2D big data in dynamic and large-scale data environments.

This paper proposes a framework based on the (a, k)-anonymity model for privacy preservation in D2D big data networks. To solve the problem of privacy protection, we use the MapReduce framework [9, 10] to process dynamic and large-scale urban data to streamline the data collection process and avoid overloading the data transmission. At the same time, the widely used (a, k)-anonymity model [11] is used as privacy-preserving framework in D2D communication.

The contributions of this paper are as follows. Firstly, a D2D big data framework and its threat model are proposed, and various security issues therein are illustrated in this paper. Secondly, we use the distributed MapReduce to classify and group data for massive datasets to improve the efficiency of computing and reduce computing time. Thirdly, we propose (a, k)-anonymity privacy-preserving framework and algorithm for D2D big data communication deployed MapReduce. Finally, we conduct detailed theory analysis and a comprehensive set of experiments to show that our method is effective for D2D big data privacy protection with low information loss and computing time.

The remainder of this paper is organized follows. Section 2 describes the D2D big data framework and its threat model. Section 3 introduces the related definitions and background knowledge used in this paper. The proposed (a, k)-anonymity privacy-preserving framework for D2D big data deployed on MapReduce is detailed in Sections 4 and 5. Section 6 describes the experimental results as well as a detailed theoretical analysis. We introduce related work and provide the conclusions of this paper in Sections 7 and 8, respectively.

2. D2D Big Data Framework and Its Threat Model

D2D communication is a key technology of fifth generation (5G) cellular networks. Once the communication link has been established, the data can be transmitted directly without intermediate equipment. It can reduce pressure due to data on the core network of the communication system, improve the rate of spectral utilization, and significantly expand the network capacity [12]. D2D communication was originally designed to query peers adjacent to a given one for the desired content and to broadcast urgent or interesting information to other mobile users. A large amount of heterogeneous data is generated during this operation. Investigating and utilizing these large and complex datasets have significant research and practical value. Security issues are considered an important factor in D2D communication.

The mining, analyzing, and processing of urban big data can lead to the leakage of private user data as there are a large number of sensing nodes in D2D communication systems that continuously collect information concerning users, government departments, and national infrastructures. If these data are not effectively protected during the acquisition process, their interception as a result of big data mining and analysis can lead to violations of privacy, which can seriously harm users, organizations, and national interests. Therefore, a practical privacy-preserving framework for D2D communication systems featuring big data is imperative. Figure 1 shows the D2D big data communication framework and threat model of our system. In this model, the adversary can launch active attacks such as spoofing as well as passive attacks like eavesdropping. Our aim is to protect the privacy of users against malicious attacks during D2D communication. To do so, we use the generalization method proposed in [13, 14] to anonymize the original data of users. Figure 1 shows the threat model. We assume unsecured links between user and server as well as any pair of users. Therefore, the proposed privacy-preserving framework focuses on protecting the privacy of data transmitted between user and server and between pairs of users.

3. Preliminary Considerations

3.1. (a, k)-Anonymity Model. In general, the transmitted data can be described in the following form: D (Explicit-Identifier, Quasi-Identifier, Sensitive Attributes).

"Explicit-Identifier" is a set of attributes that describe the unique identifier explicitly (e.g. ID number). The "Quasi-Identifier" (QI) is a set of attributes describing empirically unique attributes of each individual (e.g., zip code). "Sensitive Attributes" (SAs) are a set of attributes containing sensitive values that need to be protected (e.g., disease). Let T^* be the perturbed table from table T, Q^* be the perturbed QI attributes T^*, and S^* be the perturbed SA attributes.

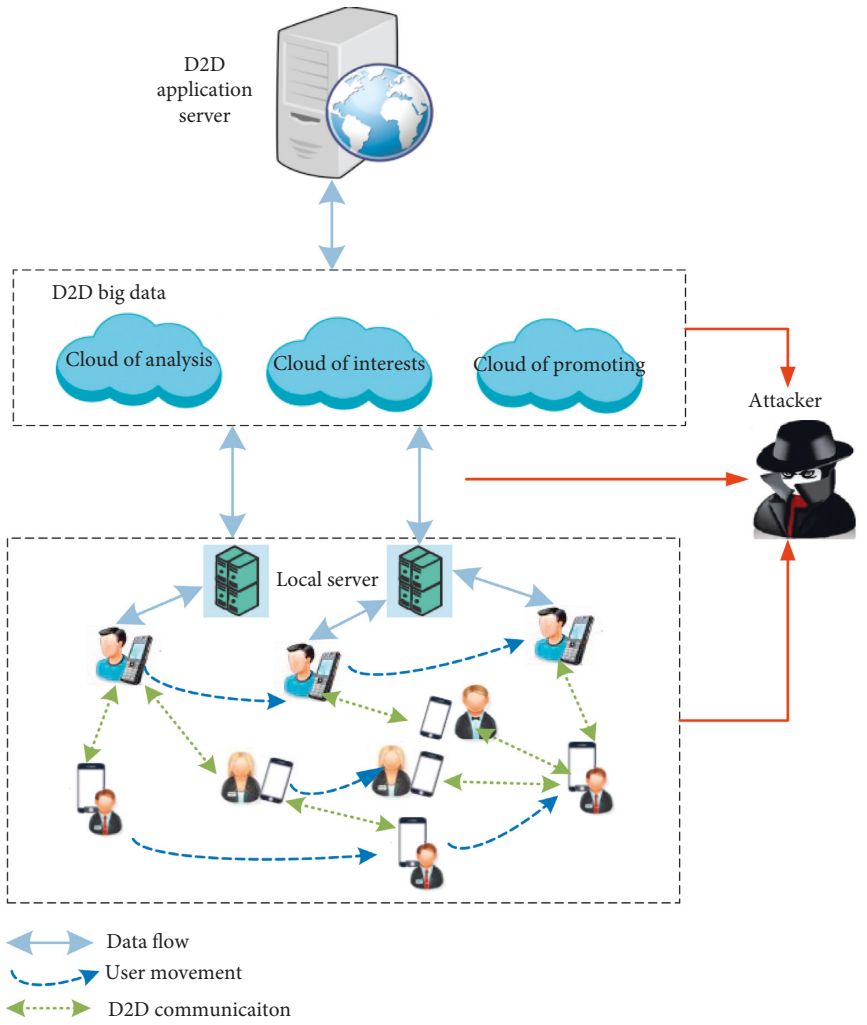

Figure 1: D2D big data framework and threat model.

Definition 1 (equivalence class). The equivalence class (EC) is a set of partial tuples in dataset DT with the same attribute value on the QI.

Definition 2 (single sensitive value (a, k)-anonymity). Given data table $DT = \{A_1, A_2, \ldots, A_m, S\}$, where the QI is $\{A_1, A_2, \ldots, A_m\}$, the SA is S. Suppose a mapping $F(DT \longrightarrow DT^*)$ that DT^* satisfies k-anonymity. For the specified $s \in S$, (EC, s) is the set of tuples for EC containing the s, and the user-specified threshold satisfies $\alpha(0 < \alpha < 1)$. If the frequency of s in each EC is not greater than α, $\forall s \in S$, $\forall EC$, s satisfies $|(EC, s)|/|(EC)| \leq \alpha$.

It is assumed that anonymous data tables DT^* satisfy the (a, k)-anonymity model for a single sensitive value with respect to QI and SA. However, (a, k)-anonymity constraints for single sensitivity values have a specified sensible value because of which the model is unsecured.

Definition 3 (multisensitive value (a, k)-anonymity). Given a data table $DT = \{A_1, A_2, \ldots, A_m, S\}$, where $QI = \{A_1, A_2, \ldots, A_m\}$, the sensitive attribute is s. Suppose a mapping $F(DT \longrightarrow DT^*)$ that DT^* satisfies k-anonymity. For $\forall s \longrightarrow S$, (EC, s) is the set of tuples for EC containing sensitive value s, and the user-specified threshold satisfies $\alpha(0 < \alpha < 1)$. If the frequency of s in each EC is not greater than $\forall s \in S$, $\forall ECs$ satisfies $|(EC, s)|/|(EC)| \leq \alpha$. Anonymous data tables DT^* are assumed to satisfy the multisensitive value (a, k)-anonymity model with respect to QI and SA.

The multisensitive value (a, k)-anonymity model extends single sensitive constraints to all values of SA. A uniform frequency constraint is set for all SAs so that each s SA of each attribute in the dataset is protected.

3.2. MapReduce. MapReduce uses the idea of "divide and conquer" to distribute the operation of large-scale datasets to every subnode under the management of a master node and integrates the intermediate results of each node to obtain the final result. Simply put, MapReduce breaks down tasks and the aggregate of the results.

There are two major components of MapReduce: JobTracer and TaskTracer. The JobTracer is used for scheduling work and the TaskTracer to perform work.

Each MapReduce task is initialized to a job that can be divided into two phases: the map phase and the reduce phase. These phases are represented by the map and the reduce functions, respectively. The map function takes an input of the form <*key*, *value*> and produces an intermediate output of the form <*key*, *value*>. The Hadoop function accepts an input, such as <*key*, (*list of values*)>, and processes the value set. Each reduce function produces zero or one as output, and this is also in the form <*key*, *value*>.

A combiner is a localized reducer operation that is a follow-up to the map operation. It performs a simple merge and repeats the key value operation before the map calculates the intermediate file. The file then decreases in size, which improves its transfer efficiency. The process of the mapper, combiner, and reducer is shown in Figure 2 for each iteration.

4. MapReduce-Based (a, k)-Anonymity Framework

4.1. MapReduce-Based (a, k)-Anonymity Algorithm. MapReduce automatically separates the data into a number of data block fragments and divides the equivalence class and iterates the above steps at the same time. As q increases, the expected number of iterations decreases until all data have been assigned to the EC. In this case, it should be noted that each EC has a maximum of q values. The global file here contains all ECs as well as the newly formed EC in which each MapReduce job is placed. The global file assigns a subset to the mapper, combiner, and reducer and traverses the data to merge duplicate values to streamline them. The process of (a, k)-anonymity algorithm framework based on MapReduce is shown in Figure 3. Through finding identifier of the optimal EC and adjustment of center point, fast and accurate classification analysis of big data is realized, which greatly reduces the computational complexity and avoids the problem of clustering results falling into the local optimal and effectively improves the overall clustering accuracy of the algorithm.

All data records are first assigned to a general EC, where the range of each dimension is its domain. The data records for each EC are then split into q parts until the split is violated. In the algorithm below, the global file contains all ECs, and the newly formed EC is appended by the driver to the end of each iteration. The functions of the mapper, combiner, and reducer in each iteration are as follows. In the anonymity algorithm, the mapper contains QI and SA, where the dimension refers to the number of each QI.

This paper proposes (a, k)-anonymity privacy-preserving algorithm for D2D big data deployed on MapReduce (as shown in Algorithm 1). The algorithm consists of three processes: mapper, combiner, and reducer. Having obtained the global file in a distributed file system, MapReduce calculates the input split according to the input file before performing map calculations. Each input slice is intended for a map task. The input slice stores not the data themselves but an array of slice lengths and the position of the recorded data.

Steps 1–5 are mapper processes; they help to find the optimal equivalence class of each data record and increase the key-value pairs ((dim, 1) (s, 1)).

Steps 6–10 are those of the combiner process, where Step 6 is used to superimpose the frequency of the same dimension as has been traversed and calculates the frequency of each dimension. Similarly, Step 8 is used to superimpose the frequency of SA on the dimension, Step 10 is used to count the frequencies of all SA in the dimension, and Step 11 is used to constrain SA to satisfy the (a, k)-anonymity model.

Steps 12–19 are those of the reducer stage. Each reducer accepts one (k, V), where each V is a list equal to the size of the dimension. The reducer selects the dimension into which the EC is split. This dimension is called the cut dimension. It selects the cutting boundary, called the cutting point, based on cutting size. Depending on the amount of data, various heuristic functions can be used to select the cutting dimension and the point. For example, a general heuristic function selects the largest dimension and the qth quantile as cut dimension and cut point, respectively. Having determined the cutting dimension and cutting point, the reducer checks if the EC can be further split without violating the (a, k)-anonymity model. In simultaneously executing splitting and outputting q ECs, for a newly created EC, the value of "1" of the split flag indicates that splitting has taken place. However, if the EC cannot be divided into q ECs, that is, at least one EC violates the (a, k)-anonymity model, the reducer recursively checks the feasibility of the split from $(q-1)$ to $q \geq 2$. This process guarantees that there are no more than 2,000 ECs at the conclusion of the algorithm. If this leads to a privacy violation, the process is repeated. If all dimensions are checked and there is no further split, the reducer outputs "0" as the flag for the end of the split.

The function findDimToCut (V, i) returns the ith dimension in V according to the heuristic function. The function findCutPoint$_p$ (V, d) determines the cut point based on the selection criteria such as quantile. Finally, the function Cut$_p$ (eq, c-dim, cp$_1$, cp$_2$, ..., cp$_{p\text{-}1}$) cuts the equivalence class eq according to the dimension at position cp$_1$, cp$_2$, ..., cp$_{p-1}$, and returns P ECs.

The newly constructed EC is then appended to the global file by the driver. The algorithm iterates until there is at least one split flag with a value of one. A value of "0" as split flag output indicates that there is no remaining EC that can be further split. At this point, the algorithm terminates and the global file containing all ECs. An example of the outputs of a mapper and a combiner is shown as follows (Tables 1–3).

5. Analysis

5.1. Complexity. The above algorithm calculates the complexity of the mapper, combiner, and reducer in each cycle, and the complexity of the entire algorithm is the sum of these three comparisons.

5.1.1. Complexity Analysis of the Mapper. Each mapper has N/n data records and finds the equivalence class that belongs to it for each data record. The time complexity of finding all

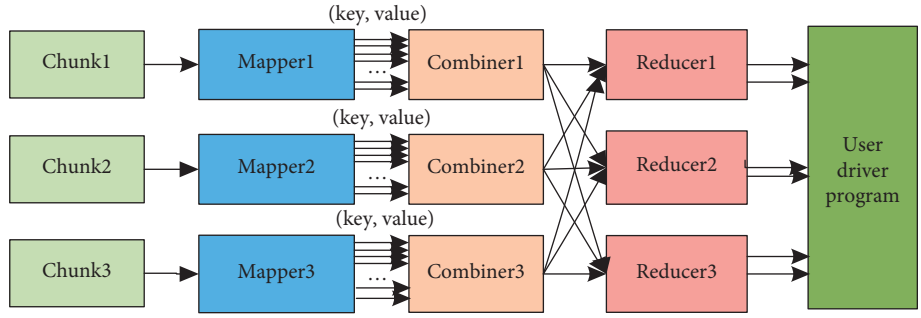

Figure 2: MapReduce work flow chart.

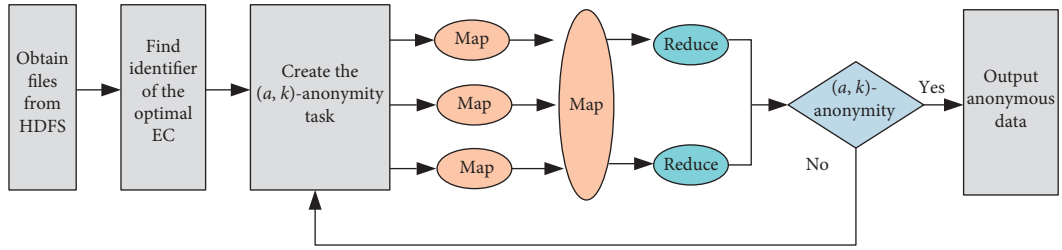

Figure 3: (a, k)-anonymity algorithm framework based on MapReduce.

Input: Data table DT, anonymous constrains (k), specified sensitive value (s), frequency constrains (a).
Output: Anonymous table DT* that satisfies multiple sensitive values (a, k)-anonymity
(1) Obtain global files from the Distributed File System (DFS)
(2) Eq-id = findFinestEQ (v) and $|EQ(v)| >= k$. // Find the identifier of the optimal EC, and at least k data record in the EC.
(3) Output-$V = \emptyset$. //V is a list of values.
(4) Add key-value pairs ((dim, 1) (s, 1)) for each dimension in v.
(5) Output (k, V ((dim, 1) (s, 1))).
(6) If dim* = dim, superimposes the frequency of dim. // dim* is the first dimension traversed to.
(7) Otherwise, iterate again until the next dim* and dim are found to have the same value.
(8) If $s^* = s$, superimpose the frequency of s. // s^* is the first SA traversed.
(9) Otherwise, traverse again until the next s^* and s are found to have the same value.
(10) Where s contains all the values if SA and superimposes the frequency of occurrence of the SA, (s_1, 1) (s_2, 1) (s_3, 1), . . ., (s_n, 1).
(11) $(|\text{dim}, s|/|\text{dim}|) < a\,(0 < a < 1)$. // Constrain the SA so that it satisfies the (a, k)-anonymity, $|(\text{dim}, s)|$ refers to the dimension frequency containing SA, and |dim| refers to the frequency of the dimension.
(12) For (i in [1, . . ., dimension]). // i indicates the number if dimensions.
(13) c-dim = findDimToCut (V, d^*).
(14) For p in [q, . . ., 2]. // p is the number of EC, and q is the maximum number that can be split.
(15) If there is no violation of the (a, k)-anonymity during the sharing process, continue cutting, $eq_1, eq_2, \ldots, eq_{p-1} = cut_p(eq, c - dim, cp_1, \ldots, cp_{p-1})$.
(16) For j in [1, . . ., p−1].
(17) Output (eq_j, "1").
(18) Exit ().
(19) Output (eq_j, "0").

Algorithm 1: Multisensitive value (a, k)-anonymity algorithm based on MapReduce.

Table 1: Source data and output of the mapper.

Source data	Output of the mapper	
[1 : 6][10 : 11]	1, {s_1, 1}, 1	10, {s_1, 1}, 1
[1 : 6][10 : 11]	1, {s_2, 1}, 1	11, {s_2, 1}, 1
[1 : 6][10 : 11]	4, {s_1, 1}, 1	10, {s_1, 1}, 1
[1 : 6][10 : 11]	6, {s_1, 1}, 1	13, {s_1, 1}, 1
[1 : 6][10 : 11]	6, {s_3, 1}, 1	11, {s_3, 1}, 1

Table 2: Grouping the output data of Table 1 as an equivalence class.

Quasi #1	Quasi #2	Sensitive
1	10	s_1
1	11	s_2
4	10	s_1
6	13	s_1
6	11	s_3

Table 3: Output of the combiner.

[1 : 5][10 : 18]	1, $\{s_1 : 1, s_2 : 1\}$, 2	10, $\{s_1, 1\}$, 1
	4, $\{s_1, 1\}$, 1	11, $\{s_2 : 1, s_3 : 1\}$, 2
	6, $\{s_1 : 1, s_3 : 1\}$, 2	13, $\{s_1, 1\}$, 1

ECs in the ith cycle is $O(i)$. To find each data record to be compared, the complexity calculation cannot exceed $O(i)$. In each comparison, the dimensions of all records are tested which were included in the EC. The process of finding the smallest equivalence class must be performed to ensure that all classes can be further separated. Therefore, the time complexity of i iterations in the mapper is $O((N/n) \cdot i \cdot d)$.

5.1.2. Complexity Analysis of the Combiner. The mapper does not output all data records. Suppose a data record has a range of variation of t_p. After ith iteration, the range is $\rho_j^i(t_q \in \mathrm{eq}_j)$, and each combiner receives $\sum_{p=1}^{(N/n)} p_{j(t_p \in \mathrm{eq}_j)}^i$, which is assumed to be the best EC. The combiner is based on the input value pair. The values in the dimension are combined to repeat the value operation so that the time complexity is $O(\sum \rho_{j|t_p \in \mathrm{eq}_j}^i \cdot d \cdot j)$. The complexity of calculating the output value is $O(\min(\sum p_{j|t_p \in \mathrm{eq}_j}^i \cdot d \cdot j, \arg\max_d dv_j^d))$.

5.1.3. Complexity Analysis of the Reducer. The time complexity of the reducer receiving values is $O(\min(\sum p_{j|t_p \in \mathrm{eq}_j}^i \cdot d \cdot j, \arg\max_d dv_j^d))$. It finds the cut dimensions and points. Thus, the total complexity is $O(\min(\sum p_{j|t_p \in \mathrm{eq}_j}^i \cdot d \cdot j, \arg\max_d dv_j^d \cdot (N/n) \cdot d \cdot p))$. The notation used in this paper can be explained as in reference [15] (Table 4).

5.2. Analysis of Information Loss. The loss of information is a good assessment for the generalization of data. If the anonymized data have n tuples and m attributes, the information loss (ILoss) is calculated as follows:

$$\mathrm{ILoss} = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{\left|\mathrm{upper}_{ij} - \mathrm{lower}_{ij}\right|}{n \cdot m \cdot \left|\max_j - \min_j\right|}, \tag{1}$$

where lower_{ij} and upper_{ij}, respectively, represent the lowest and highest boundary values of attribute j in metagroup i after generalization and $\min_j$ and $\max_j$, respectively, represent the minimum and maximum values of j in all records.

5.3. Security Analysis. In the single sensitive value (a, k)-anonymity model, the frequency of the SA in each EC is not greater than a, i.e., $|(\mathrm{EC}, s)|/|\mathrm{EC}| \le \alpha$. In this algorithm, the tuple set containing SA in the EC is $\sum \rho_{j|t_p \in \mathrm{eq}_j}^i \cdot d \cdot m$.

Therefore, in this algorithm, SA value s should satisfy

$$\frac{\sum \rho_{j|t_p \in \mathrm{eq}_j}^i \cdot d \cdot m}{k} \le \alpha. \tag{2}$$

In this paper, conditional entropy $H(\mathrm{SA} \mid \mathrm{QI})$ is adopted to reflect the degree of privacy protection, which gives the uncertainty prediction for SA when the QI is known.

Table 4: List of notations.

Notation	Explanation
N	Number of data records
n	Number of mappers/combiners
d	Number of dimensions
m	Number of sensitive attributes
t_p	Data record
eq_m	m^{th} equivalence class
ρ_m^i	Whether the decision variable is splittable in the ith iteration
dv_j^d	Record the number of different dimensions d

$$\begin{aligned} H(\mathrm{SA} \mid \mathrm{QI}) &= \sum_{q \in Q} p(q) \cdot H(S \mid Q = q) \\ &= -\sum_{q \in Q} p(q) \sum_{s \in S} p(s \mid q) \cdot \log p(s \mid q). \end{aligned} \tag{3}$$

According to equation (3), the minimum privacy level is calculated as follows:

$$H(S \mid Q)_{\min} = \log(\alpha + k). \tag{4}$$

Since the uncertainty increases when k increases or $a \cdot k$ decreases, the inequality $H(S \mid Q) \ge H(S \mid Q)_{\min}$ holds for every possible output of the proposed algorithm.

6. Experiments and Analysis of Results

In the experiment, the characteristics of the data transmission of the algorithms tested were analyzed by studying the validity of data. This experiment was implemented on Hadoop, a software framework that implements MapReduce.

6.1. Experimental Datasets. There are two datasets in this experiment which are described below.

6.1.1. Poker Hand Datasets. The poker hand dataset contained 11 numeric attributes. The first 10 predictive attributes were used as those of the QI, and variable classes were used as SA. The ranges of odd and even QI were 1–4 and 1–13, respectively, and the dataset was split into small blocks using the preprocessed MapReduce.

6.1.2. Synthetic Datasets. A synthetic dataset [10] was formed which comes from two sets of data. One consisted of 10 million data records with a size of 1.4 GB, and the other consisted of 10 million data records with a size of 14 GB. A set of data consisted of 15 dimensions and 10 clusters, each with a random mean and bias. Each dimension in the mapper was normalized to 1 M~100 M. The dataset with 10 M was divided into 70 copies and that with 100 M into 300 fragments.

6.2. Results. We set $q = 2$ and used the longest side of the bounding rectangle to select the cut dimension. The middle value was used to perform the cut.

Table 5: Number of mappers/reducers.

	Poker	Synthetic (10 M and 100 M)
#mapper	10	100
#reducer	5	30

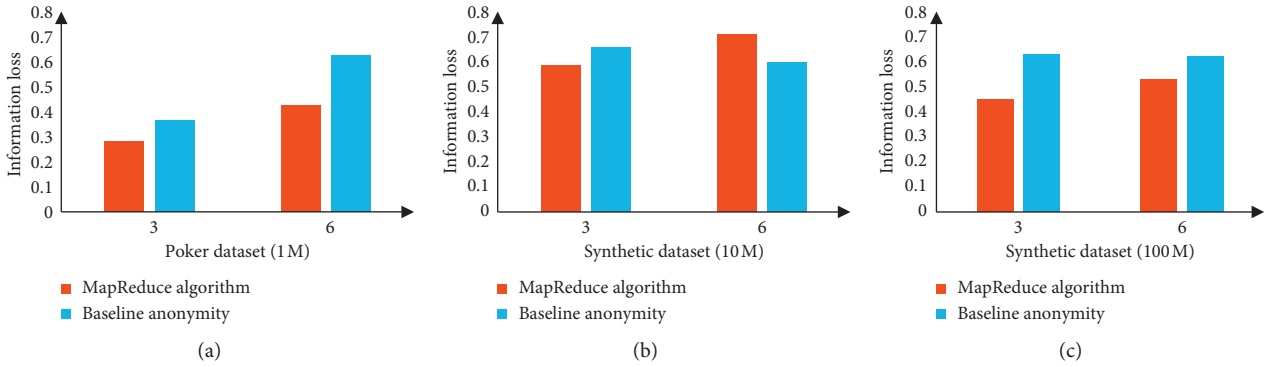

Figure 4: ILoss vs. poker dataset and synthetic dataset.

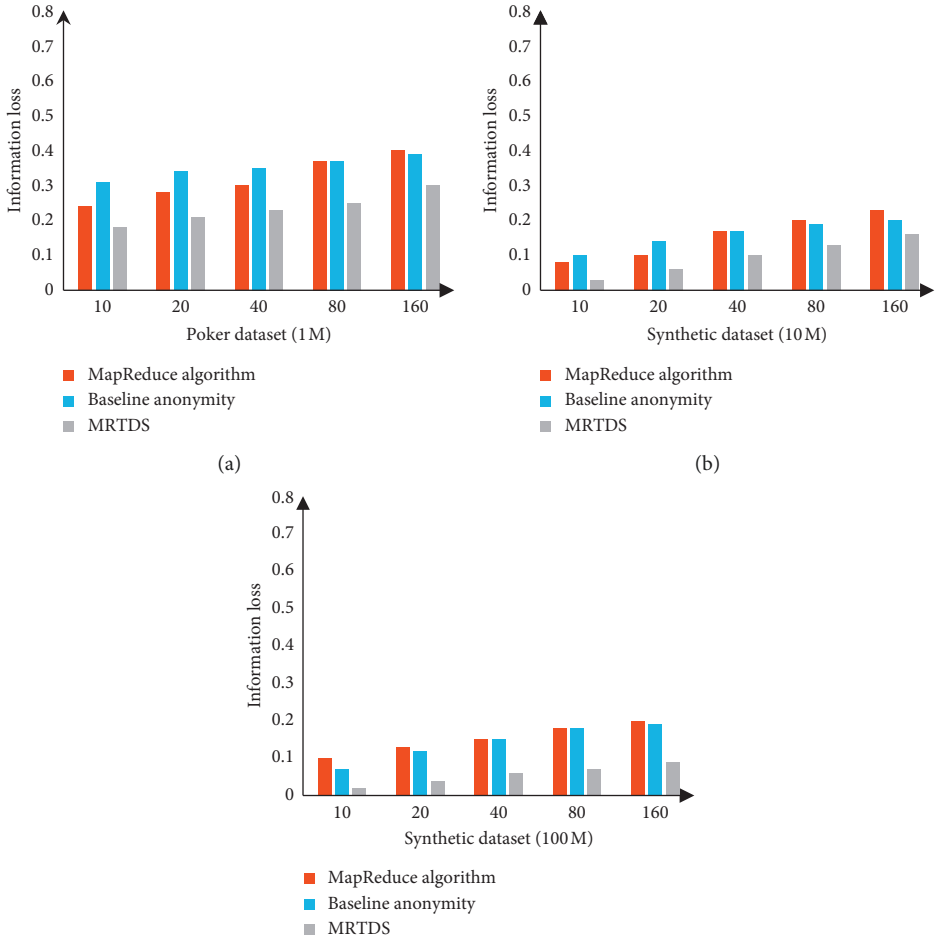

Figure 5: ILoss vs. poker dataset and synthetic dataset.

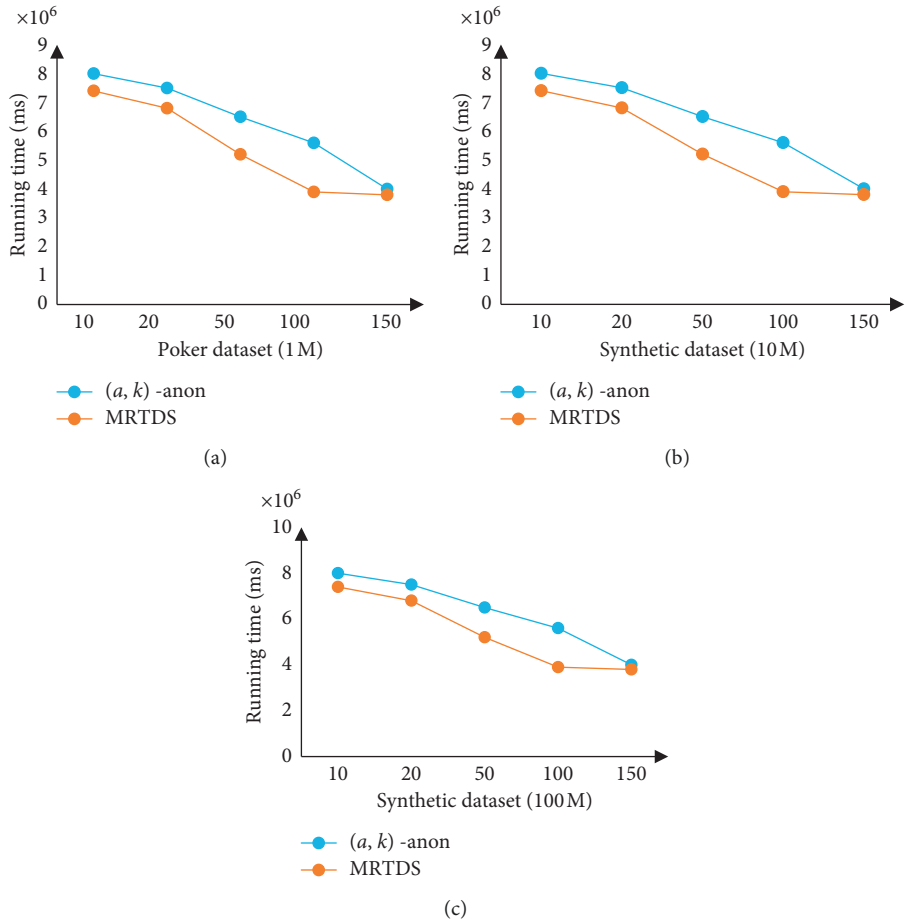

Figure 6: Running time vs. poker and synthetic datasets.

Anonymity varied from 10 to 160; the number of mappers/reducers selection method in this paper refers to the method in reference [15]. Table 5 from reference [15] shows the number of mappers/reducers for each dataset.

To analyze the impact of the algorithm on the collected data, this paper compared the ILoss in the MapReduce anonymity algorithm with that in the baseline anonymization algorithm as shown in Algorithm 1. In database anonymization, each dataset was divided into eight equal parts and each part was anonymized separately. The result was divided into eight equal parts. Figure 4 shows the comparison of the ILoss between the proposed MapReduce anonymity algorithm and baseline anonymity algorithm with respect to different record sizes. As the value of k increased, the amount of ILoss decreased. The reason is that as the record size increases, it needs fewer generalizations to achieve (a, k)-anonymization. The ILoss of the MapReduce anonymity algorithm was smaller than that of the baseline anonymization algorithm; this is because there is more completely traversed process in the MapReduce anonymity algorithm which reduces the ILoss.

Figure 5 shows the comparison of the ILoss between the proposed MapReduce anonymity algorithm, baseline anonymity algorithm, and MapReduce top-down specialization (MRTDS) (proposed by Zhang et al. [16]) with respect to different record sizes. As the record size increased, the amount of ILoss decreased. This is because as the record size increases, it needs fewer generalizations to achieve (a, k)-anonymization. By comparison, the MapReduce anonymity algorithm had the smallest amount of ILoss because the baseline anonymization and the MRTDS algorithms could not make full use of all the data, unlike the MapReduce anonymity algorithm, and could not combine values between large data blocks. If the data were split into more blocks, the difference would have been greater.

Figure 6 shows a comparison of run time between the proposed MapReduce anonymity algorithm and MapReduce top-down specialization (MRTDS). Each dataset was divided into eight parts. As the value of k increased, the run time decreased because fewer iterations were needed to satisfy the privacy requirements for the value of k. As the privacy parameters increased, the privacy requirements became increasingly strict, and more dimensions were needed to be checked to select the cutting size. In the MapReduce anonymity algorithm, on the one hand, the number of iterations was reduced by a combination of local repeated values in the combiner so that as little data as possible were written to the disk. On the other hand, the redundancy of the data was reduced by a combination of duplicate values for all data through the reducer stage. The

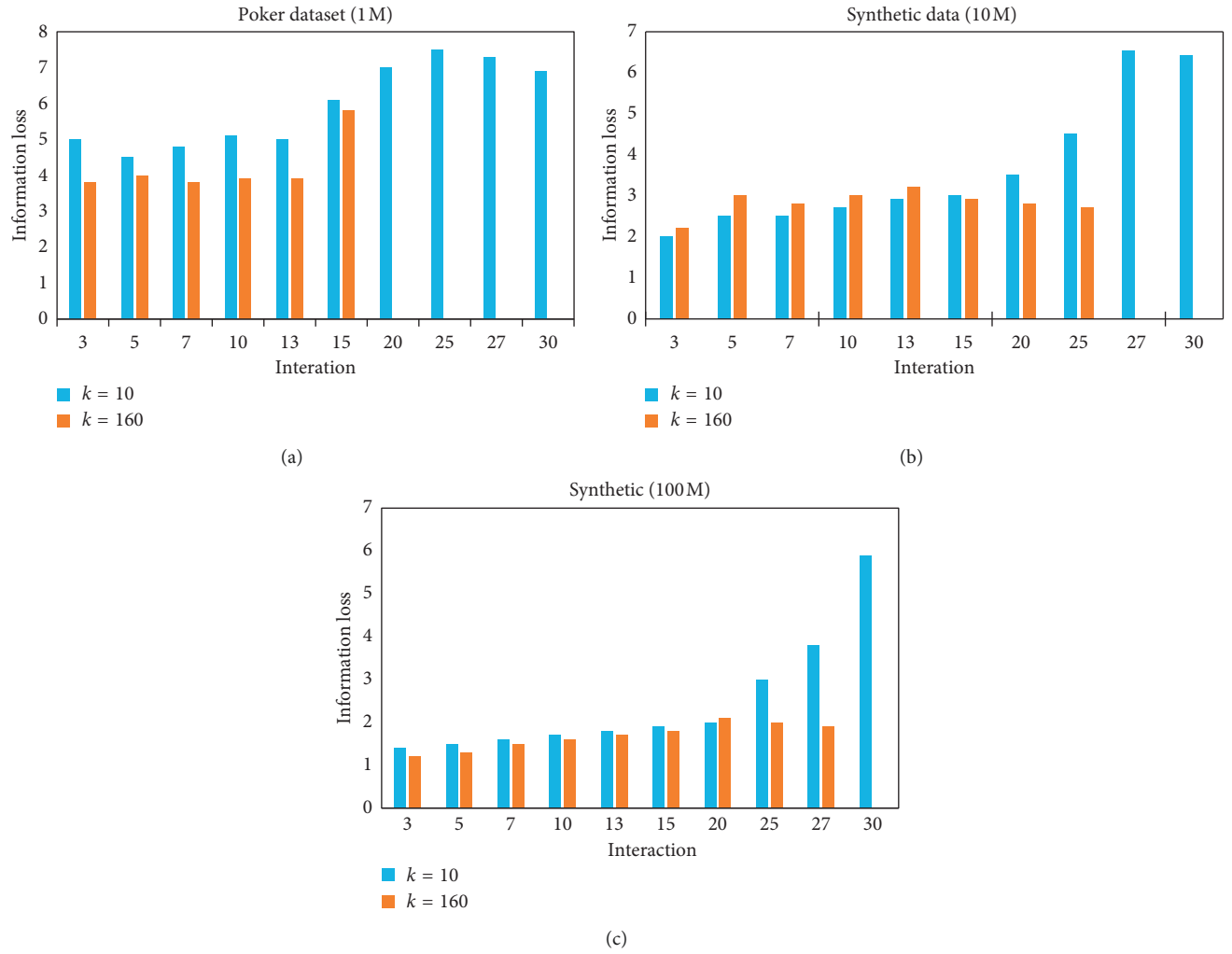

Figure 7: Running time and ILoss with the number of iterations vs. poker and synthetic datasets.

Table 6: Comparison with existing works.

Privacy model	Network	Solution	ILoss
Wu et al. [18]	Mobile network	Encryption	Less
Hakola et al. [19]	IoT	Encryption	Less
Li et al. [21]	Mix-net	Encryption generalization	Reasonable
Our model	IoT	MapReduce generalization	Reasonable

process of merging produced a large number of intermediate files, but MapReduce reduced data written to the disk to as little as possible and directly outputted to the reduce function.

Figure 7 shows the variation of ILoss with respect to the number of iterations for different record sizes ($k = 10$ and $k = 160$). As the number of iterations increases, the amount of ILoss decreased. This is because as the number of iterations increases, it needs more generalizations to achieve (a, k)-anonymization. When the number of iterations was small, the amount of information lost at $k = 10$ was not considerably different from that at $k = 160$. As the number of iterations increased, the amount of ILoss at $k = 10$ became larger and that at $k = 160$ decreased. That is to say, in the process of big data collection, in light of the security analysis needed, given the volume of data and with the privacy protection algorithm proposed in this paper, the larger the value of k, the smaller the amount of ILoss.

7. Related Work

Two methods are mainly used to solve the problem of privacy protection in D2D big data communication: encryption and anonymization methods. Encryption methods do not reduce the amount of data, and the energy consumed in data processing is not reduced. However, the amount of information loss is minimal. Anonymization methods reduce the amount of data and the energy consumed during data processing but cause a larger amount of information loss.

Encryption methods focus on implementing identity and data authentication, key generation, distribution and effective

management through symmetric encryption, and asymmetric encryption. Fu et al. [17] proposed a privacy-preserving and secure multidimensional aggregation scheme for smart grid communications by integrating privacy homomorphism encryption with aggregation signature scheme. Wu et al. [18] proposed a dynamic trust-relationship-aware data privacy protection (DTRPP) mechanism for mobile crowd-sensing for data privacy by distributing forged public keys. The DTRPP can dynamically manage nodes and estimate the degree of trust of the public key. Hakola et al. [19] proposed a method for D2D key management, where the method features the reception of a communication-mode change command and the generation of a local device security key based on a secret key and a base value. Kumari et al. [20] used encryption technology to encrypt information repeatedly and transmit it to the next hop for privacy protection. However, this method incurs computational overhead in the processes of data encryption and decryption.

Anonymization refers to hiding the identity of and sensitive information concerning participants of communication. Anonymity makes it impossible to match sensitive information with specific entities. Li et al. [21] proposed a privacy-preserving data collection model based on (a, k)-anonymity; they dynamically encrypt some data and adjust the portion to balance the trade-off measure in generalization. Cordeiro et al. [10] proposed a cloud-oriented access control mechanism for big data privacy protection and authentication that attains privacy protection of large-scale data in the cloud environment. Zhang et al. [22] proposed a cloud-oriented scalable big data privacy protection framework that can perform large-scale dataset anonymization and process anonymity datasets.

Table 6 shows the comparison of our method and the state of the art. Our privacy model achieves ideal privacy level and reasonable information loss.

8. Conclusion

D2D communication has been proposed as a promising technology for 5G cellular networks. The data often contain sensitive information which should be protected during data transmission. In this paper, we proposed a (a, k)-anonymous D2D big data privacy-preserving framework deployed on MapReduce. To improve the efficiency of computing and to reduce computing time, we use the distributed MapReduce to classify and group data for massive datasets. To resist the possible attacks, we adopt (a, k)-anonymity as a security framework for privacy preserving. Experimental results and theoretical analysis show that our method is effective for privacy protection in D2D big data communication.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of China (grant no. 61702316), Shanxi Provincial Natural Science Foundation (grant no. 201801D221177), Shandong Provincial Natural Science Foundation, China (grant no. ZR2015FL032), and Project of Shandong Province Higher Educational Science and Technology Program, China (grant no. J13LN84).

References

[1] X. Wang, Y. Zhang, V. C. M. Leung, N. Guizani, and T. Jiang, "D2D big data: content deliveries over wireless device-to-device sharing in large-scale mobile networks," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 32–38, 2018.

[2] X. Chen, B. Proulx, X. Gong, and J. Zhang, "Exploiting social ties for cooperative D2D communications: a mobile social networking case," *IEEE/ACM Transactions on Networking*, vol. 23, no. 5, pp. 1471–1484, 2015.

[3] X. Wang, Z. Sheng, S. Yang, and V. C. M. Leung, "Tag-assisted social-aware opportunistic device-to-device sharing for traffic offloading in mobile social networks," *IEEE Wireless Communications*, vol. 23, no. 4, pp. 60–67, 2016.

[4] A. Zhang, L. Wang, X. Ye, and X. Lin, "Light-weight and robust security-aware D2D-assist data transmission protocol for mobile-health systems," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 3, pp. 662–675, 2017.

[5] X. Zhang, Z. Yi, Z. Yan et al., "Social computing for mobile big data," *Computer*, vol. 49, no. 9, pp. 86–90, 2016.

[6] I. Anagnostopoulos, S. Zeadally, and E. Exposito, "Handling big data: research challenges and future directions," *Journal of Supercomputing*, vol. 72, no. 4, pp. 1494–1516, 2016.

[7] J. Xu, G. Yang, Z. Chen, and Q. Wang, "A survey on the privacy-preserving data aggregation in wireless sensor networks," *China Communications*, vol. 12, no. 5, pp. 162–180, 2015.

[8] S. Kumari, "Design flaws of "an anonymity two-factor authenticated key agreement framework for session initiation protocol using elliptic curve cryptography"," *Multimedia Tools & Applications*, vol. 76, no. 11, pp. 1–3, 2016.

[9] X. Zhang, L. T. Yang, C. Liu, and J. Chen, "A scalable two-phase top-down specialization approach for data anonymization using mapreduce on cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 363–373, 2014.

[10] R. L. F. Cordeiro, C. Traina, A. J. M. Traina, J. López, U. Kang, and C. Faloutsos, "Clustering very large multi-dimensional datasets with MapReduce," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD 11'*, pp. 690–698, ACM, San Diego, CA, USA, August 2011.

[11] X. Ye, Y. Zhang, and M. Liu, "A personalized (a, k)-anonymity model," in *Proceedings of the 2008 the Ninth International Conference on Web-Age Information Management*, pp. 341–348, IEEE, Zhangjiajie Hunan, China, July 2008.

[12] M. Wang and Z. Yan, "A survey on security in D2D communications," *Mobile Networks and Applications*, vol. 22, no. 2, pp. 195–208, 2017.

[13] A. K. Pal, "Achieving k-anonymity using full domain generalization," thesis, Department of Computer Science and Engineering and National Institute of Technology Rourkela, Odisha, India, 2014.

[14] L. Sweeney, "Achieving *k*-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 571–588, 2002.

[15] H. Zakerzadeh, C. C. Aggarwal, and K. Barker, "Privacy-preserving big data publishing," in *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*, ACM, New York, NY, USA, June 2015.

[16] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. S. Shen, "Security and privacy in smart city applications: challenges and solutions," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 122–129, 2017.

[17] S. Fu, J. Ma, H. Li, and Q. Jiang, "A robust and privacy-preserving aggregation scheme for secure smart grid communications in digital communities," *Security and Communication Networks*, vol. 9, no. 15, pp. 2779–2788, 2016.

[18] D. Wu, S. Si, S. Wu, and R. Wang, "Dynamic trust relationships aware data privacy protection in mobile crowd-sensing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2958–2970, 2018.

[19] S. J. Hakola, T. Koskela, and H. M. Koskinen, "Method and apparatus for device-to-device key management," US 8989389 B2, 2015.

[20] S. Kumari, M. K. Khan, and M. Atiquzzaman, "User authentication schemes for wireless sensor networks: a review," *Ad Hoc Networks*, vol. 27, pp. 159–194, 2015.

[21] H. Li, J. Ma, and S. Fu, "A privacy-preserving data collection model for digital community," *Science China Information Sciences*, vol. 58, no. 3, pp. 1–16, 2014.

[22] X. Zhang, C. Liu, S. Nepal, C. Yang, W. Dou, and J. Chen, "SaC-FRAPP: a scalable and cost-effective framework for privacy preservation over big data on cloud," *Concurrency and Computation: Practice and Experience*, vol. 25, no. 18, pp. 2561–2576, 2013.