IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 18, 2018, accepted December 14, 2018, date of publication December 19, 2018,
date of current version January 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2888685

Verifiable Keyword-Based Semantic Similarity Search on Social Data Outsourcing

YIZHU ZOU[1], (Student Member, IEEE), XIN YAO[1,2], (Member, IEEE), ZHIGANG CHEN[1,2], (Member, IEEE), AND MING ZHAO[1,2], (Member, IEEE)

[1]School of Software, Central South University, Changsha 410075, China
[2]Mobile Health Ministry of Education-China Mobile Joint Laboratory, Changsha 410075, China

Corresponding author: Xin Yao (xinyao@csu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672540, Grant 61572526, and Grant 61872131, and in part by the Major Program of National Natural Science Foundation of China under Grant 71633006.

ABSTRACT In the data-driven economy era, social data have tremendous business and potential values. Obtaining authentic social data is the first step in mining the business value of social data. In this paper, we consider an emerging social data outsourcing paradigm. Therein, different online social network (OSN) operators outsource their social data to a third-party social data provider (SDP), who resells them to data consumers who can be any individual or entities. However, a dishonest SDP may return untrusted query results to data consumers through various activities, such as adding fake data and deleting/modifying correct data. To deal with these dishonesties, we propose a basic scheme and an enhanced scheme to allow data consumers to verify the correctness and completeness of their received social data from the SDP. Data consumers in the basic scheme utilize the public APIs to collect the sampled social data and compare them with their received social data. This scheme is a probabilistic verification method as the data consumers only having a tiny proportion of the social data. To permit data consumers to verify the query results trustworthiness deterministically, we proposed an enhanced scheme, in which the OSN operator generates some cryptographic auxiliary information. The SDP can construct a verification object for the data consumer based on these information. Extensive experiments ran on a real Twitter dataset confirm that our schemes are effective and efficient.

INDEX TERMS Social data outsourcing, verifiable keyword-based search, semantic similarity measurements.

I. INTRODUCTION

As the popularity of Internet, Online Social Networks (OSNs) play an important role in people's life due to its various advantages. As reported by the Statistics Portal, almost 2,196 million, 336 million, and 411 million users are active as of July 2018 on Facebook, Twitter, and Sina Weibo, respectively [1]. So many OSN users produce large amounts of user-generated content. For example, 31,250,000 Facebook's messages, 347,222 Twitter's tweets, and 48,611 Instagram's photos are posted in a minute [2]. These social data bring new insights to the current business models. More than 92% of marketers stated that their business was deeply attached to social media marketing, and 80% of marketers' efforts increased pageviews of their websites [3]. Meanwhile, almost 97% of marketers are currently using social data [4].

How to access these social data with vast economic value? Data consumers can commonly call the public APIs provided by each OSN operator itself. However, such APIs only offer very limited functionalities, leading to data consumers receiving incomplete, biased, and even incorrect social data. For instance, Twitter provides the Sample API and the Filter API to data consumers to obtain tweets, but only randomly samples at most 1% data in all social data meeting the query condition [5]. Besides, Twitter also provides the search functionality by the Search API to search the tweets in the past seven days. However, calling these APIs suffers from limited rate. For example, only 180 calls per user and 450 calls per application are permitted in the Search API every fifteen minutes. As another way to access social data, the Firehose API can provide 100% of public tweets to data consumers. But it requires monetary cost and high-performance servers to host and process the real-time tweets. The public APIs of other OSNs such as Sina Weibo have similar constraints as well. To this end, we consider

an new paradigm social data outsourcing, which is more effective and efficient than the prior accessing methods. This system includes three entities: the OSN operator, social data provider (SDP), and data consumers. Therein, the SDP collects complete data from the OSN operator and offers paid data services to data consumers who can be any individuals or entities requiring the complete social data satisfying some criteria. There are some favorite SDP examples, like Gnip, DataSift, NTT Data, CrowdEye, etc.

However, these SDPs could not be fully trusted. In 2017, Google was pointed out that they manipulated its search suggestions to support Democratic presidential hopeful Hillary Clinton when she was in trouble due to ''Email Controversy'' [6]. When someone typed ''Hillary Clinton cri'', Google's search suggestions were like ''Hillary Clinton crime reform'' and ''Hillary Clinton crisis'', while Yahoo or Bing search suggestions were the fact ''Hillary Clinton crimes''. As other famous review websites, like Yelp or Dianping, they had been reported that they might modify the reviews from customers to improve the effects of some specific businesses [7]. In our scenario, an untrusted SDP may return fake query results to data consumers by adding fake data and deleting/modifying true data.

In this paper, we are the first to consider the problem of *verifiable keyword-based semantic similarity search* in social data outsourcing scenario, whereby data consumers can guarantee the social data receiving from the SDP are correct and complete. The reliable social data should meet the following two conditions. First, the social data should satisfy the query condition and be indeed generated by the targeted OSN users. Second, all social data satisfying the query condition should be returned.

Specifically, we tackle the following specific problem. Here, we define each user in the OSN operator as one data generator, who can publish his/her emotional states or all that he/she sees and hears at any time at any place. All social content of the data generator i published from the timestamp t_s he created the account to the current timestamp t_c can be denoted by $\mathcal{M}_i = \{m_{i,t_s}, \cdots, m_{i,t_c}\}$, where m_{i,t_s} is a concatenation of a message m and a time stamp t_s. Therefore, the social content from one OSN operator can be represented by $\mathcal{M} = \{\mathcal{M}_1, \cdots, \mathcal{M}_i, \cdots, \mathcal{M}_n\}$ $(i \in [1, n])$. In this paper, we focus on single keyword-based semantic similarity query. For example, the query condition $\mathcal{Q} = \{2,$ ''Galaxy S9'', [7/10/2018, 8/25/2018]$\}$ can be interpreted that (1) the search keyword is ''Galaxy S9''; (2) the semantic similarity distance should be less than 2; (3) the posting times of social contents should be located in the time domain [7/10/2018, 8/25/2018]. The query result should be a subset of $\mathcal{M}$. The query result is complete if all social contents satisfying the query condition, and it is correct if none of social contains therein do not satisfy the query.

We propose two schemes to address verifiable social content queries. The basic solution allows data consumers to utilize the public APIs provided by each OSN operator to collect the sampled social contents and compare these data with the query results. Since the public APIs only sample 1% data from the original data, the basic solution is a probabilistic verification scheme. However, the basic scheme suffers from three drawbacks, like the detection probability dramatically decreasing with the number of query results, needing to know the organization structure of all keywords, and running APIs all the time of the query time domain. To thwart these limitations, we propose an enhanced solution, whereby each OSN operator generates some cryptographic auxiliary information for its dataset. After receiving the query request from the data consumer, the SDP constructs a verification object based on the query condition and auxiliary information. Next, data consumer utilizes the query results and the verification object to verify the query-result trustworthiness. Note that the enhanced solution can determinately guarantee the correctness and completeness of the query results.

In this paper, we use a real Twitter dataset with 41.6 million social contents to thoroughly evaluate our two schemes. Our experimental results show that the basic scheme can guarantee the correctness and completeness of the query results with the probability of 100% when the number of query results is larger than 10^4. To evaluate the impact of data size on performance, we randomly sample three subsets of social contents, i.e., $\mathcal{D}_1$ (with 2 million contents), $\mathcal{D}_2$ (with 24 million contents) and $\mathcal{D}_3$ (with 41.6 million contents). For the dataset $\mathcal{D}_3$, the OSN operator only needs to take $5.495s$ to generate $76.9873MB$ auxiliary information (just 1.04% of the storage size of the original dataset). Our experimental results demonstrate the efficacy and efficiency of our schemes.

The roadmap of this paper is described as follows. We formally formulate our problem in Section II. Sections III and IV separately detail our two schemes. In Section V, we analyze the security and performance of our two schemes. Section VI evaluates the performance of our two schemes via a real Twitter dataset. The related work and the conclusion are described in Sections VII and VIII, respectively.

II. PROBLEM FORMULATION

A. SYSTEM AND ADVERSARY MODELS

In our social data outsourcing scenario, our system consists of three entities: the OSN operator, SDP, and data consumers. The OSN operator as a social platform provides various social services (e.g., making friends, showing daily life, or hunting job) to registered users. Some popular OSN operators include Facebook, Sina Weibo, Twitter, Youtube, LinkIn, Snapchat, and so on. The SDP integrates social data from different OSN operators and answers the data queries from data consumers. To describe clearly, we here only consider a single OSN operator, but similar operations can independently performed on social contents of each OSN operator. Data consumers can be any individuals or businesses, who request particular social contents from the SDP to improve their business interests.

In this paper, we assume each OSN operator is trusted entirely and honestly outsources its dataset to the SDP. Conversely, the SDP is considered as an untrusted entity, and may return fake query results to data consumers by adding,

deleting, or modifying data. Besides, the communications between any two entities are assumed as secured by adopting traditional mechanism like Transport Layer Security (TLS). Thus, a dishonest query result can only result from the misbehaviour of the SDP. Note that the OSN operator is highly willing to help identify malicious SDPs, as data consumers who make critical decisions based on manipulated query results may eventually blame the OSN operator.

B. PROBLEM FORMULATION

Social data can be categorized into two aspects: social graph and social contents. In a social graph, each node denotes a social user (like twitters in Twitter or bloggers in microblogging), and edges mean the relationships among connected users. We have considered the social graph in previous work [8], so we here mainly focus on social contents, which are published by social users at a specific time stamp. For instance, tweets on Twitter, comments on Youtube, posting on Facebook or Weibo. To describe simplicity, we take tweets as an example, which can easily be extended to other social contents.

We represent all tweets of the twitter i as a set with timeline denoted by $\mathcal{M}_i = \{m_{i,t_f}, \cdots, m_{i,t_l}\}$, where t_f and t_l denotes the first and last time stamp of posting tweets. To describe simply, we assume that the points t_f and t_l are identical for each twitter, and replaced with t_s and t_c. Intuitively, normal twitters always post one or multiple topics within a tweet, while not arbitrary combine several unrelated words. To represent these topics briefly, we extract keywords from each tweet as described in Section IV-A. Thus, the keywords of a tweet $m_{i,t}$ ($t_s \leq t \leq t_c$) can be denoted by $\{t, K_1, \cdots, K_{\tau_i}\}$. Note that the number of keywords in $m_{i,t}$ is $|\tau_i|$.

We consider single keyword-based semantic similarity queries for any tweets, each of which can be the following two cases: (i) **Exact Keyword Query**: Data consumer only focuses on listening or monitoring what is said about its company, product or brand on social network in a period. (ii) **Semantic Similarity Query**: Data consumer also focuses on analyzing tweets of other related companies, products or brands in the same period. Data consumers submit these queries to the SDP, which specifics the query condition and the interested OSN as well. For instance, the famous electrical company Samsung launches a query $\mathcal{Q} = \{\delta$, "Samsung", "1/15/2016$\leq t \leq$10/25/2017"$\}$). Here, the parameter δ can be utilized to distinguish two cases. Specifically, if δ is equivalent to zero, the query $\mathcal{Q}$ is an exact keyword query; Otherwise, it is a semantic similarity query, and δ denotes the semantic similarity distance.

The SDP processes the query on the specified OSN dataset, like Tweets in this paper. The query result includes a subset $\mathcal{M}'$ of social content $\mathcal{M}$, including all the social contents whose keywords satisfy the query condition. Continuing the previous example, the SDP returns the tweets with the exact keyword "Samsung" from 1/15/2016 to 10/25/2017 for the single keyword query if $\delta = 0$; or the tweets with keywords, which have similar semantic with "Samsung" (e.g., "Apple" and "Huawei") for the single keyword semantic similarity query if $\delta > 0$. If needed, the social graph for social users who post contents in $\mathcal{M}'$ needs to be returned as well.

Based on our system and adversary models, a query result is said to be trustworthy if the following requirements are satisfied.

- *Content correctness*: All social contents in $\mathcal{M}'$ are indeed in $\mathcal{M}$ and satisfy the query condition.
- *Content completeness*: $\mathcal{M}'$ contains all social contents in $\mathcal{M}$ that satisfy the query condition.
- *Social-graph authenticity*: The returned social graph for social users in $\mathcal{M}'$ is the same as that in $\mathcal{M}$.

Social-graph authenticity have been studied in our previous work [8]. The OSN operator generates auxiliary information for social graph, which can then be verified by the data consumer. Thus, we subsequently focus on achieving social content correctness and completeness.

III. BASIC SCHEME

In this section, we first take a overview of our basic scheme to understand its key idea, and then, introduce the details.

A. OVERVIEW OF BASIC SCHEME

Recall that the OSN operator also provides the public APIs for data consumers to obtain sampled social contents. In our basic scheme, we encourage data consumers to utilize these public APIs to identify the untrusted behaviors. The core idea is that, for specific queries, data consumers compare the sampled social contents collected by the public APIs with the query results returned from the SDP. If the sampled social contents are the subset of the query results, data consumers consider that the query results are trusted with a probability; Otherwise, the SDP returns fake results to him/her.

B. DETAILS OF BASIC SCHEME

The single keyword-based semantic similarity search includes two categories: exact keyword search and semantic similarity search. In our basic scheme, data consumers are assumed to know all semantic similarity keywords for any specific keyword. For example, data consumers should know that the set {"Apple", "Google", $\cdots$, "Nokia"} is the semantic similarity set of the keyword "Samsung". Hence, semantic similarity search can be replaced by multiple exact keyword searches. For brevity, we here focus on an exact keyword search.

For a query $\mathcal{Q} = \{\delta, \mathsf{keyword}, \mathsf{time\ domain}\}$, we assume the SDP should return η social contents (i.e., $\{m_1, \cdots, m_\eta\}$). Since the public APIs only return 1% of random sampled social contents [5] for the query $\mathcal{Q}$, we let the sampled social contents by $\{m'_1, \cdots, m'_{\lceil 0.01 \cdot \eta \rceil}\}$. To verify the correctness and completeness of social contents, data consumers calculate hash values for each social content in the query results and sampled social contents, and compare these hash values. If all hash values for sampled social contents can be matched

with that for social contents, social contents can be viewed as satisfying correctness and completeness with a probability p (namely, $\forall i \in [1, \lceil 0.01\eta \rceil], \exists j \in [1, \eta], s.t.\ h(s_i) = h(r_j)$); Otherwise, it fails. Note that the overhead of matching processing in the basic scheme is $\mathcal{O}(\eta^2)$. Intuitively, the number of η is higher, the detection probability p is higher. Therefore, we analyze the relations between the number η and the detection probability p.

In our adversary model, the attacks launched by SDPs consist of three types: deleting, adding and modifying social contents. We assume that the attackers delete, add or modify γ social contents. Based on these premises, we conclude detection probability p for these attacks as the following equations:

$$\begin{aligned} p_{\text{del,mod}} &= 1 - \frac{\binom{\eta-\gamma}{0.01\eta}}{\binom{\eta}{0.01\eta}} \\ &= 1 - \frac{\eta-\gamma}{\eta} \cdot \frac{\eta-1-\gamma}{\eta-1} \cdot \cdots \cdot \frac{\eta-0.01\eta+1-\gamma}{\eta-0.01\eta+1} \\ p_{\text{add}} &= 1 - \frac{\binom{\eta}{0.01\eta}}{\binom{\eta+\gamma}{0.01\eta}} \\ &= 1 - \frac{\eta}{\eta+\gamma} \cdot \frac{\eta-1}{\eta-1+\gamma} \cdot \cdots \cdot \frac{\eta-0.01\eta+1}{\eta+\gamma-0.01\eta+1} \end{aligned}$$

Since $\frac{\eta-j-\gamma}{\eta-j} \leq \frac{\eta-j-1-\gamma}{\eta-j-1}$, it follows that: $1\text{-}(\frac{\eta-\gamma}{\eta})^{0.01\eta} \leq p_{\text{del,mod}} \leq 1 - (\frac{\eta-0.01\eta+1-\gamma}{\eta-0.01\eta+1})^{0.01\eta}$ and $1\text{-}(\frac{\eta}{\eta+\gamma})^{0.01\eta} \leq p_{\text{add}} \leq 1 - (\frac{\eta-0.01\eta+1}{\eta-0.01\eta+1+\gamma})^{0.01\eta}$. Here, we utilize the minimal probability to represent the probability, i.e.,

$$p_{\text{del,mod}} \approx 1 - (\frac{\eta-\gamma}{\eta})^{0.01\eta} \tag{1}$$

and

$$p_{\text{add}} \approx 1 - (\frac{\eta}{\eta+\gamma})^{0.01\eta}, \tag{2}$$

separately. As shown in Figure 1, we set γ as $\{0.01\eta, 0.05\eta, 0.1\eta\}$, and conclude that the smaller the number of query results is, the lower the detection probability p is. When γ is fixed as 0.01η and η decreases from 10^5 to 10^3, the detection probability p decreases from 100% to 10%.

IV. ENHANCED SCHEME

Notwithstanding, the basic scheme can simply guarantee the correctness and completeness of query results. However, it has three drawbacks: (1) If the number of query results η is less than 10^4, the detection probability will dramatically decrease as depicted in Fig. 1; (2) Data consumers need to know the semantic similarity keyword set; (3) Data consumers cannot collect the sampled social contents published over 7 days unless running the APIs all the time [5]. For example, data consumers may wish to gather all social contents in the previous year and do not have any plans to do that beforehand. In this case, data consumers cannot collect social contents with the public APIs. To thwart these drawbacks, three challenges should be addressed: the first is

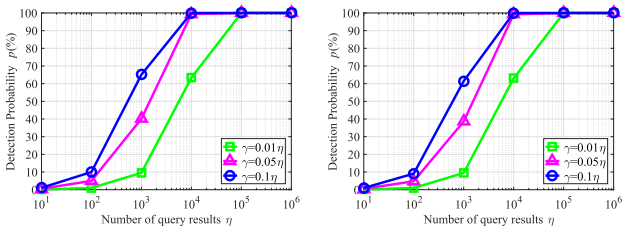


FIGURE 1. Relations between η and p for various attacks. (a) Deleting/Modifying attack. (b) Adding attack.

how to guarantee the query results $\mathcal{R}_\mathcal{Q}$ actually contains the similar semantic keywords of the query $\mathcal{Q}$; the second is how to ensure all similar semantic keywords are considered and their social contents are returned; the third is how to verify the correctness and completeness of the posting time of social content. To solve these challenges, we propose an enhanced scheme.

In what follows, we first describe text modeling to preprocess social contents in Sect. IV-A, a dataset NameNet to store the relationships among company, brand or product names in Sect. IV-B, and the semantic similarity measurements in Sect. IV-C. Subsequently, we detail the auxiliary information generation, the query processing and the results verification in our enhanced scheme, successively.

A. TEXT MODELING

Currently, the datasets outsourced by each OSN operator are noisy and unstructured. To make our discussion easily, we first deploy the text modeling to preprocess the original datasets. Let $\mathcal{M}$ be an original dataset, and n denote the number of users in the dataset. For each user $i \in [1, n]$, the corresponding social data is defined as $\mathcal{M}_i$. To prevent the identify-linkage attack, the OSN operator assigns each user an anonymous ID to hide the true ID. Intuitively, data consumers mainly focus on meaning while not meaningless words. Hence, we introduce the following text modeling for the OSN operator to extract keywords for each social content.

In the first phase, the OSN operator removes stop words for each social content in a stop-word list,[1] which are more general and meaningless. For example, "the", "those" and "it". In the second phase, the OSN operator further reduces inflected words to stem forms by conducting stemming [9]. In this case, the words with different forms can be mapped to the same word. For instance, "play", "playing", and "played" are all mitigated to "play". Therefore, we denote the keyword set for the j-th social content $m_{i,j}$ of data user i by $\{j, K_1, \cdots, K_{\tau_i}\}$.

B. NameNet

After extracting keywords for each social content, our primary task is defining the relationships among keywords. Intuitived by a lexical database of English words (WordNet) [10],

[1] http://www.lextek.com/manuals/onix/

we propose a data structure, named by NameNet, to organize these keywords. In the WordNet database, nouns, verbs, adverbs and adjectives are organized by a variety of semantic relations into synonym sets (synsets) representing one concept. However, WordNet cannot be directly adopted to our problem due to lacking special words, like company, brand and product names. To thwart this hinder, we exploit the concept of nouns in WordNet to build a database NameNet, which express the semantic relation of company, brand, and product names. Inspired by cognitive evidence that human beings organize knowledge in a hierarchical manner, we encode the NameNet with the typical hyponym/hypernym (is-a). The organization structure is a tree. The upper the names are, the more abstract the concepts are. Figure 2 shows a fragment of is-a relation between names in NameNet. The meaning of the keyword ''electronic'' is more abstract than that of ''cell phones'', while the meaning of ''Hilton'' is more specific than that of ''Chained-brand''.

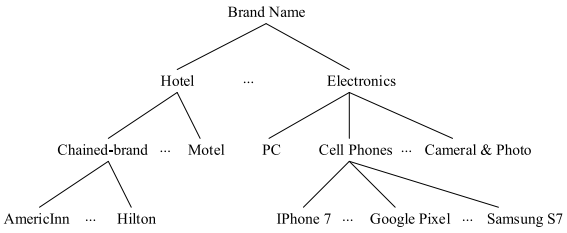


FIGURE 2. Is-a relation in NameNet.

C. SEMANTIC SIMILARITY MEASUREMENTS

In this subsection, we introduce the method to evaluate the similarity between any two keywords in NameNet. Let the path from the root to the current keyword denote the keyword in NameNet. For instance, the keyword ''Hilton'' is denoted by Path(Hilton)={''Brand Name'', ''Hotel'', ''Chained-brand'', ''Hilton''}. Hence, the depth is equivalent to the cardinality of the path, e.g., depth(Hilton)=|Path(Hilton)|=4. Since the structure of NameNet is a tree, we adopt the path-based measurements [11] to define the semantic similarity of any two keywords. They mainly consist of two aspects: the path length of two keywords, and the position of the keywords in the NameNet.

Here, we consider two typical measurements. Given two keywords K_1 and K_2 in NameNet, Varelas *et al.* [11] define their semantic similarity measurement as:

$$\mathsf{sim}_1(K_1, K_2) = 2 \cdot \mathsf{deep\_max} - \mathsf{len}(K_1, K_2) \tag{3}$$

, Leakcock and Chodorow [12] define it as:

$$\mathsf{sim}_2(K_1, K_2) = -\log \frac{\mathsf{len}(K_1, K_2)}{2 \cdot \mathsf{deep\_max}} \tag{4}$$

. Therein, the notation $\mathsf{len}(K_1, K_2)$ denotes the length of the shortest path from the keyword K_1 to the keyword K_2 in NameNet. For example, the value of len(''AmericInn'', ''Hilton'') is 2 and that of len(''AmericInn'', ''IPhone 7'') is 6 in Figure 2. Besides, the notation $\mathsf{deep\_max}$ means the max depth of the taxonomy (i.e., the height of the NameNet).

In what follows, we detail the auxiliary information generation, the query processing and the results verification in our enhanced scheme, successively.

D. GENERATING AUXILIARY INFORMATION

As stated before, we should guarantee the correctness and completeness of social contents, all similar keywords and the posting time. Next, we will introduce the auxiliary information generation in our enhanced scheme.

As described in Section IV-A, each OSN operator extracts keywords from each social content. To organize social contents efficiently, we introduce the inverted index structure, which commonly consists of two parts. The first part contains uniform keywords, like ''Samsung S7'', ''Google Pixel'' and so on in Figure 3. The second part contains social contents, which is listed with the order of posting time. The posting time of the social contents in the left should be earlier than that of the social contents in the right. As for the keyword K=''Google Pixel'' in Fig. 3, there are eight social contents with this keyword from the starting time s to the last time e, i.e., $\{m_{2,s}, m_{4,s+1}, \cdots, m_{4,e}\}$, in which each social content $m_{i,t}$ contains the keyword ''Google Pixel''.

For all social contents of each keyword, the OSN operator builds an Merkle Hash Tree (MHT).[2] As shown in Figure 3, the OSN operator first concatenates each social content with the posting time, and hashes the concatenated value by the classical SHA-1 method [14]. For example, the values from h_1 to h_8 are the hash values of social contents with the keyword ''Google Pixel''. Subsequently, the OSN operator computes hash values of parent nodes, like $h_{1-2} = h(h_1||h_2)$, and signs the root h_{root} (e.g., $\mathcal{S}(h_{1-8})$ for ''Google Pixel''). Finally, the OSN operator denotes all signatures of roots as the first auxiliary information $\mathcal{AUX}_1$.

By analyzing the Eqs. (3) and (4), we conclude that the distance between any two keywords is tied with two varieties, $\mathsf{deep\_max}$ and $\mathsf{len}(K_1, K_2)$. The $\mathsf{deep\_max}$ is an constant value and its value is equivalent with the height of the NameNet. Besides, the value $\mathsf{len}(K_1, K_2)$ can be caculated by $|P(K_1) \cup P(K_2)| - |P(K_1) \cap P(K_2)|$. Therefore, the maximum value of $\mathsf{len}(K_1, K_2)$ should be $2 \cdot \mathsf{deep\_max} - 1$. To generate the auxiliary information that authenticating the semantic similarity keywords, the OSN operator needs to traverse the whole tree and count all keywords for each different distance. As shown in Figure. 4, the OSN operator first builds an table under the aforementioned example. Next, the OSN operator also builds an merkle hash tree for each uniform distance (δ=1 in our example). Finally, the OSN operator defines all signatures for various distances as the second auxiliary information $\mathcal{AUX}_2$. All these auxiliary information $\{\mathcal{AUX}_1, \mathcal{AUX}_2\}$ and the original social contents will be outsourced to the SDP.

[2]More details about MHT can refer to [13]

Keywords	Lists of Social Contents					
Samsung S7	$m_{1,k}\|\|t_s$	$m_{3,k}\|\|t_{s+1}$	$m_{1,k}\|\|t_{s+2}$	...	$m_{3,k}\|\|t_{e-1}$	$m_{4,k}\|\|t_e$
Iphone 7	$m_{3,k}\|\|t_s$	$m_{1,k}\|\|t_{s+1}$	$m_{5,k}\|\|t_{s+2}$	...	$m_{2,k}\|\|t_{e-1}$	$m_{1,k}\|\|t_e$
Google Pixel	$m_{2,k}\|\|t_s$	$m_{4,k}\|\|t_{s+1}$	$m_{3,k}\|\|t_{s+2}$	...	$m_{5,k}\|\|t_{e-1}$	$m_{4,k}\|\|t_e$
Hilton	$m_{4,k}\|\|t_s$	$m_{4,k}\|\|t_{s+1}$	$m_{7,k}\|\|t_{s+2}$	...	$m_{3,k}\|\|t_{e-1}$	$m_{2,k}\|\|t_e$
AmericInn	$m_{6,k}\|\|t_s$	$m_{2,k}\|\|t_{s+1}$	$m_{4,k}\|\|t_{s+2}$	...	$m_{2,k}\|\|t_{e-1}$	$m_{7,k}\|\|t_e$
Cannon 760D	$m_{7,k}\|\|t_s$	$m_{5,k}\|\|t_{s+1}$	$m_{2,k}\|\|t_{s+2}$	...	$m_{4,k}\|\|t_{e-1}$	$m_{5,k}\|\|t_e$
Nikon D5500	$m_{1,k}\|\|t_s$	$m_{9,k}\|\|t_{s+1}$	$m_{8,k}\|\|t_{s+2}$	...	$m_{7,k}\|\|t_{e-1}$	$m_{3,k}\|\|t_e$

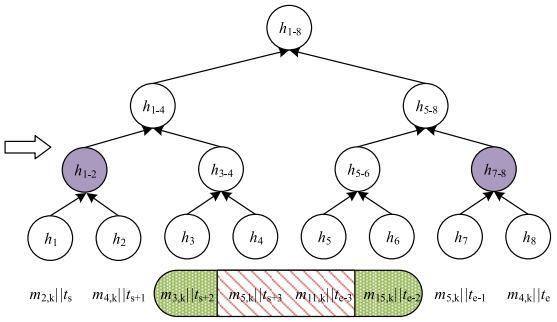
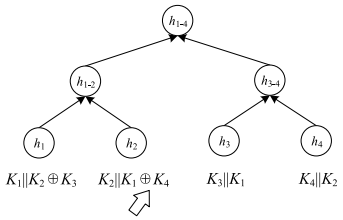


FIGURE 3. Building MHTs for social contents in invert index.



Keywords	δ=0	δ=1	δ=2	δ=3
K_1=Brand Name	K_1	$K_1\|\|K_2 \oplus K_3$	$K_1\|\|K_2 \oplus K_3 \oplus K_4$	$K_1\|\|K_2 \oplus K_3 \oplus K_4$
K_2=Hotel	K_2	$K_2\|\|K_1 \oplus K_4$	$K_2\|\|K_1 \oplus K_4 \oplus K_3$	$K_2\|\|K_1 \oplus K_4 \oplus K_3$
K_3=Electronics	K_3	$K_3\|\|K_1$	$K_3\|\|K_1 \oplus K_2$	$K_3\|\|K_1 \oplus K_2 \oplus K_4$
K_4=Chained-brand	K_4	$K_4\|\|K_2$	$K_4\|\|K_2 \oplus K_1$	$K_4\|\|K_2 \oplus K_1 \oplus K_3$

FIGURE 4. Building MHTs for different semantic distance δ.

E. QUERY PROCESSING

When receiving the query request $\mathcal{Q}$ from data consumers, the SDP first searches all social contents and finds out the corresponding query result $\mathcal{R}_\mathcal{Q}$. To guarantee the correctness and completeness of $\mathcal{R}_\mathcal{Q}$, the SDP needs to return two untamable verification objects to data consumers, i.e., $\mathcal{VO}_1$ and $\mathcal{VO}_2$.

Given a query $\mathcal{Q}$=$\{\delta, K, [t_s, t_e]\}$, the SDP first searches the NameNet database and finds out all keywords $\{K_1, \cdots, K_z\}$, in which the semantic similarity distance between any keyword K_i and K should be smaller than δ. For each keyword K_i, the SDP finds out all social contents whose posting time are located in the time domain $[t_s, t_e]$.

Next, the SDP checks $\mathcal{AUX}_1$ to obtain the verification object $\mathcal{VO}_1$ to reconstruct the MHT root for the social contents of each keyword (see Section IV-D). Specifically, the qualified posting time and two boundary elements each correspond to the posting time of a unique element, so each hash value of the concatenation of the posting time and its corresponding social content is a leaf node of the MHT. The verification object for each such leaf node includes the siblings of itself, its parent, its parent's parent, etc. Since these leaf nodes are adjacent in the MHT, their auxiliary authentication information should be combined to reduce the likely redundancy. The verification object $\mathcal{VO}_1$ includes all the auxiliary authentication information and related root signature.

Finally, the SDP checks $\mathcal{AUX}_2$ to obtain the auxiliary information needed to reconstruct the MHT root for the keywords of each semantic similarity distance value (see Section IV-D). Since we here also build MHT for each semantic similarity distance value, the construction processing of the verification object $\mathcal{VO}_2$ is also similar to that of $\mathcal{VO}_1$, including all the auxiliary authentication information and related root signature.

F. CORRECTNESS AND COMPLETENESS VERIFICATION

Data consumers utilize the verification objects $\mathcal{VO}_1$ and $\mathcal{VO}_2$ to verify the correctness and completeness of the query results with the following operations.

First, data consumers check that the posting time of each content locates in the range $[t_s, t_e]$. This step ensures that all contents are posted in the time range of the query.

Second, data consumers reconstruct the MHT root of all social contents for each keyword based on the $\mathcal{R}_\mathcal{Q}$ and the auxiliary information in the verification object $\mathcal{VO}_1$. The reconstructed MHT root should match the root signature in the verification object $\mathcal{VO}_1$. This step ensures that all contents satisfying the posting time are returned.

Third, data consumers reconstruct the MHT root of all based on the $\mathcal{R}_\mathcal{Q}$, and the auxiliary information in the verification object $\mathcal{VO}_2$. The reconstructed MHT root should match the root signature in the verifiation object $\mathcal{VO}_2$. This step ensures that the correctness and completeness of all keywords in the query result.

The query result is considered complete and correct if and only if the above verification succeed. The security of this scheme relies on the unanimously assumed security of the cryptographic hash function $h(\cdot)$ and the digital signature scheme. In particular, the SDP cannot fabricate a query result that can lead to valid MHT roots with correct signatures.

G. A WORKING EXAMPLE

To better understand the enhanced scheme, we take an example in Figs. 2, 3 and 4 to show these operations. Given a concrete query $\mathcal{Q}$=$\{1$, "Google Pixel", $[t_{s+3}, t_{e-3}]\}$, the SDP first finds out the correct social contents. As shown in Fig. 3, the social contents $m_{7,s+3}$ and $m_{1,e-3}$ satisfy the query $\mathcal{Q}$.

To guarantee the correctness and completeness of the posting time of social contents, SDPs return verification object $\mathcal{VO}_1$ as $\{m_{3,k}||t_{s+2}, m_{5,k}||t_{e-2}, h_{1-2}, h_{7-8}, \mathcal{S}(h_{1-8})\}$. When data consumers receive the query result and the verification object $\mathcal{VO}_1$, they reconstruct MHTs and verify the signature of the root. If the verification passes, data consumers can consider the posting time of the query results are correct and completeness; Otherwise, they reject the query results. To guarantee the correctness and completeness of the semantic similarity keywords, SDPs return verification object $\mathcal{VO}_2$ as $\{K_1||K_2 \oplus K_3,\ h_{3-4}, \mathcal{S}(h_{1-4})\}$. If data consumers can reconstruct MHTs and verify the signature of the root, data consumers can consider the returned semantic similarity keywords are correct and completeness.

V. SECURITY AND OVERHEAD ANALYSIS

In this section, we will separately analyze the security and performance of our schemes.

A. SECURITY ANALYSIS

The basic scheme asks data consumers to call the public APIs and obtain the sampled social contents. As long as these public APIs in the OSN operator has not been hacked, data consumers can collect the true sampled data. Therefore, data consumers can verify the correctness and completeness of the query results with these collected sampled social content. In our enhanced scheme, data consumers detect an incorrect and/or incomplete query result in a deterministic fashion. The reason is that the auxiliary information amounts to social contents with cryptographic methods. As long as the hash function and digital signature scheme used for constructing the MHTs are secure, the SDP cannot modify the authentic query result without failing the signature verification.

B. OVERHEAD ANALYSIS

Next, we analyze the computation, communication, and storage overhead in our schemes. The time overhead to search for qualified social contents and initial datasets preprocessing are ignored here. Note that the basic scheme needs to adopt public APIs to collect social contents, and the collection time depends on the responsing time of the OSN operator and the bandwidth. Therefore, we do not discuss the basic scheme here.

1) COMPUTATION OVERHEAD

All our schemes involve digital signature generations, verifications, and hash operations. The computation overhead is dominated by signature generations and verifications (especially, the former). Hash operations take less time in the experiment, so we can safely ignore the hash operations for simplicity.

First, we estimate the computation overhead that generating the auxiliary information. The OSN operator needs to sign each keyword. Let the number of keywords be n, so the complexity of signature operations is $\mathcal{O}(n)$. As analyzed in Section IV-D, the maximum value of $\mathsf{len}(K_1, K_2)$ should be $2 \cdot \mathsf{deep\_max} - 1$. Therefore, the number of unique semantic similarity distance values is $|2 \cdot \mathsf{deep\_max} - 1|$, where $\mathsf{deep\_max}$ denotes the heighth of the NameNet. When the NameNet is a binary tree, the complexity of signature operations is $\mathcal{O}(\log m + 1)$, where m denotes the number of keywords in the NameNet.

Next, we discuss the computation overhead that verifying a query result. Suppose that the number of keywords in the query result is z, data consumers performs up to z signature verifications, leading to the complexity of $\mathcal{O}(z)$.

2) COMMUNICATION AND STORAGE OVERHEAD

We discuss the communication overhead for transmitting the auxiliary information from the OSN operator to the SDP, which is the same as the storage overhead at the SDP for storing the auxiliary information. For convenience, let t be the number of time points during posting all social contents, and let l_{hash} and l_{sig} denote the lengths of a hash value and a digital signature. The communication overhead includes $(n+2\cdot\lceil\log_2\ n\rceil+1)\cdot l_{sig}$ and $(n\cdot(2^{\lceil\log_2\ t\rceil}-2)+(2^{\lceil\log_2\ n\rceil}-2)\cdot 2\cdot(\lceil\log_2\ n\rceil+1))\cdot l_{hash}$.

VI. EXPERIMENTAL RESULTS

In this section, we thoroughly evaluate our schemes with a real Twitter datasets. We implement our schemes with Python 3.4. All the experiments are carried out on a server with Intel(R) Xeon(R) Silver 4110 CPU@2.10GHz, 128 GB memory, 3.6 TB hard disk, and Centos 6.0.

A. DATASETS

The dataset we used is a real-world Twitter dataset and collected in 2016. The dataset includes 41.6 million social contents. To evaluate the performance of our schemes for datasets with different sizes, we have randomly sampled three subsets of social contents ($\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$) from the above dataset. The number of social contents in $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$ are 2 million, 24 million and 41.6 million, respectively. To analyze English and practical meaning keywords, we deploy Natural Language Toolkit for social content preprocessing. To construct the NameNet database, we first generate a word list based on the vocabulary of content statistics. In our experimental results, our three datasets have the same keywords, but their frequencies are different. The frequency for each word in $\mathcal{D}_3$ is larger than that in $\mathcal{D}_2$ and $\mathcal{D}_1$ Thus, we select Top-1500 keywords to construct the tree-like database.

B. GENERATING AUXILIARY INFORMATION

We now evaluate the computation and storage overhead incurred by generating auxiliary information in the schemes.

1) COMPUTATION OVERHEAD

We evaluate our scheme in terms of the number of hash, signature operations, and computation time. Table 1 lists the numbers of hash and sign operations for three datasets. We can conclude from Table 1 that (1) the numbers of sign operations for three datasets are identical due to having

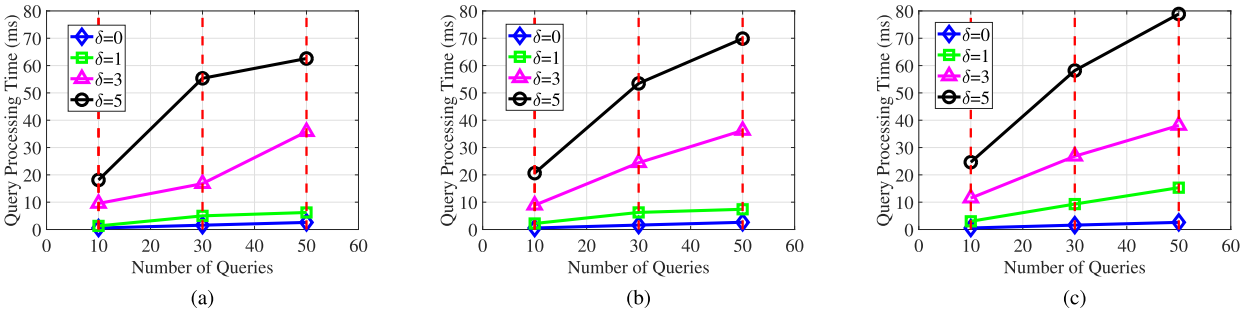


FIGURE 5. Query processing time. (a) $\mathcal{D}_1$. (b) $\mathcal{D}_2$. (c) $\mathcal{D}_3$.

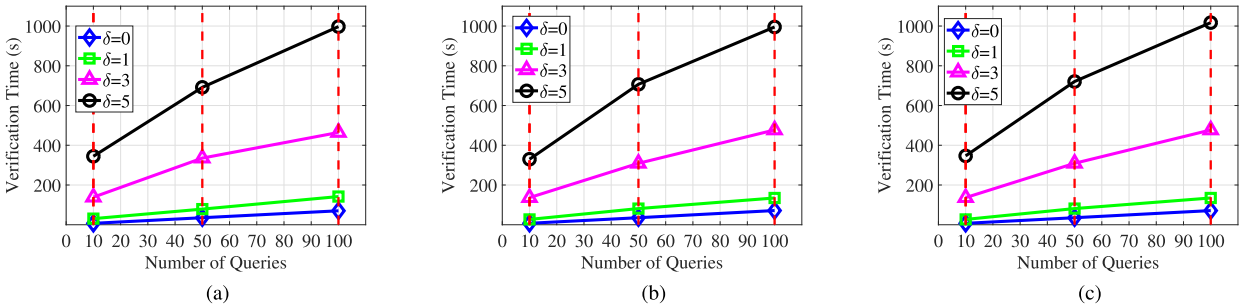


FIGURE 6. Query-result verification time. (a) $\mathcal{D}_1$. (b) $\mathcal{D}_2$. (c) $\mathcal{D}_3$.

TABLE 1. Computation overhead.

	# of *hash*	time of *hash*	# of *sign*	time of *sign*
$\mathcal{D}_1$	670894	0.5825*s*	1521	3.1636*s*
$\mathcal{D}_2$	3326222	1.4365*s*	1521	3.2036*s*
$\mathcal{D}_3$	6183438	2.2414*s*	1521	3.2536*s*

TABLE 2. Storage overhead.

	hash	*sign*	*total cost*
$\mathcal{D}_1$	9.5100*MB*	0.2608*MB*	9.7774*MB*
$\mathcal{D}_2$	44.8242*MB*	0.2608*MB*	45.0850*MB*
$\mathcal{D}_3$	76.7265*MB*	0.2608*MB*	76.9873*MB*

the same keywords' number; (2) The compuation overhead satisfies the practical requirement.

2) STORAGE OVERHEAD

Table 2 shows the storage overhead of auxiliary information, i.e., the total size of signatures, internal nodes of MHTs. As we can see from Table 2, for the same keywords in different datasets, the storage overhead of the hash and signature are independent of the datasets size. When the number of keyword increases, the hash and signature storage overhead also increase. This is of no surprise, as the complexities of storage overhead for signatures and hash values in the enhanced schemes are fixed at $\mathcal{O}(n)$. Table 2 is also clear that the scheme only requires less overhead for auxiliary information storage.

C. QUERY PROCESSING

To measure the computation overhead of query processing in our scheme. We generate three types of queries for our three datasets: $\mathcal{Q}_{10}$, $\mathcal{Q}_{30}$, and $\mathcal{Q}_{50}$. Here, $\mathcal{Q}_i$ ($i = 10$, 30, and 50) means that the number of randomly selected keywords. Besides, we set δ as {0, 1, 3, 5} in our experiments. When δ is equivalent to zero, the query is the exact keyword search; Otherwise, it is the single keyword semantic similarity search. Fig. 5 shows the query processing time of our enhanced scheme for three types of queries with all three datasets, where each point represents the average of 100 runs, each with a random seed. We can see that the query processing time of our enhanced scheme is in line with the previous analysis. When δ is larger than 5, the query processing time also meet our analysis.

D. QUERY-RESULT VERIFICATION

Figs. 6a to 6b compare the verification overhead under our enhanced for three datasets and same keywords. We also select δ from {0, 1, 3, 5} as the previous setting. We can see that the verification time is basically the same on different data sets, and satisfies the complexity analysis in Sect. V-B ($\mathcal{O}(z)$).

VII. RELATED WORK

Our work is mostly close to data outsourcing paradigm [15]–[17]. The data owner outsources his/her dataset to a third party, who answers the queries from either

data owner or other users. One security provision in data outsourcing is to ensure the integrity of the query results [18], [19]. A trivial solution to authenticate query results is to let the data owner outsource its dataset and some auxiliary information over the data to the third party which returns both the query result and a verification object computed from the auxiliary information for the querying user to verify query integrity.

Here, we mainly discuss single- and multi-dimensional query authentication with signature chaining-based schemes and MHT-based schemes. Besides, we also overviews existing solutions for authenticating keyword-based queries, and surveys alternative methods for database outsourcing.

A. SINGLE- AND MULTI-DIMENSIONAL QUERY AUTHENTICATION

To address the query authentication problem, the previous literature can be classified into two aspects: signature chaining-based schemes and Merkle Hash Tree (MHT) based schemes [13]. Narasimha and Tsudik [20] proposed an approach *DSAC* based on signature chain to verify the integrity of dynamic databases. To reduce the overhead of *DSAC*, Pang *et al.* [21] proposed a novel signature caching scheme *SigCache*. To support multi-dimensional range aggregate query, Pang and Tan [22] proposed efficient authentication schemes based on signature chaining and MHT. Moreover, many variants based on MHT are proposed for authenticating aggregation queries [23], kNN queries [24], [25], top-k spatial keyword queries [26], [27], and location-based skyline queries [28], [29]. However, none of these schemes consider semantic-similarity query over social contents, so they cannot be applied to our problem.

B. KEYWORD-BASED QUERY AUTHENTICATION

Another line of research has been devoted to verifiable keyword-based search [30]–[32], in which each keyword is represented as a root of some polynomial. It is possible to verify whether a keyword is present by evaluating the polynomial on the keyword and testing whether the output is zero or not. However, these schemes cannot be directly deployed for addressing our problem due to the query complexity linearly increasing with the number of tweets. Besides, Pang and Mouratidis [33] proposed an authenticated similarity-based document retrieval scheme. Moreover, many schemes [34]–[36] supporting verifiable keyword-based search on encrypted data are proposed. Nevertheless, none of these schemes can be applied to verifiable keyword-based semantic similarity queries over short plaintext texts.

C. OTHER RELATED WORK

In other contexts, authenticating is also a critical problem, like two-layer sensor networks. Zhang *et al.* [27] proposed techniques to verify the top-k query results returned by untrusted service providers. Besides, Shi *et al.* [37] presented several schemes to guarantee the completeness and correctness of range query results even if data is encrypted. However, all these schemes focus on different contexts and cannot be directly deployed for addressing our problem.

VIII. CONCLUSION

In this paper, we consider verifiable keyword-based semantic similarity search problem in social data outsourcing scenario. To address this problem, we propose two schemes. The basic scheme depends on the public APIs provided by the OSN operator. Data consumer verifies the correctness and completeness of the query results by comparing the sampled social contents and the query results. The enhanced scheme first introduce the NameNet to organize the keywords and deploy the classical Merkle hash tree to guarantee the authentic query results. Our experimental results shows that our schemes are efficacy and efficiency.

REFERENCES

[1] The Statistics Portal. (2018). *Global Social Networks Ranked by Number of Users 2018*. [Online]. Available: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/
[2] M. Kapko. (2015). *7 Staggering Social Media Use by-the-Minute Stats*. [Online]. Available: https://www.cio.com/article/2915592/social-media/7-staggering-social-media-use-by-the-minute-stats.html#slide5
[3] S. Kusinitz. (2017). *16 Stats That Prove Social Media Isn't Just a Fad [New Data]*. [Online]. Available: https://blog.hubspot.com/marketing/social-media-roi-stats
[4] M. Stelzner. (2014). *Social Media Marketing Industry Report*. [Online]. Available: https://www.socialmediaexaminer.com/social-media-marketing-industry-report-2014/
[5] F. Morstatter, J. Pfeffer, H. Liu, and K. Carley, "Is the sample good enough? Comparing data from Twitter's streaming API with Twitter's Firehose," in *Proc. ICWSM*, Boston, MA, USA, Jul. 2013, pp. 400–408.
[6] V. Richardson. (2017). *Google Accused of Manipulating Searches, Burying Negative Stories About Hillary Clinton*. [Online]. Available: https://www.washingtontimes.com/news/2016/jun/9/google-accused-burying-negative-hillary-clinton-st/
[7] C. Patrick. (Jul. 2014). *Yelp's Newest Weapon Against Fake Reviews: Lawsuits*. [Online]. Available: http://www.bloomberg.com/news/articles/2013-09-09/yelps-newest-weapon-against-fake-reviews-lawsuits
[8] X. Yao, R. Zhang, Y. Zhang, and Y. Lin, "Verifiable social data outsourcing," in *Proc. INFOCOM*, Atlanta, GA, USA, May 2017, pp. 1–9.
[9] M. Porter, *Readings in Information Retrieval*. San Mateo, CA, USA: Morgan Kaufmann, 1997.
[10] A. Kilgarriff and C. Fellbaum, *WordNet: An Electronic Lexical Database*. Cambridge, MA, USA: MIT Press, 1998.
[11] G. Varelas, E. Voutsakis, P. Raftopoulou, G. Petrakis, and E. Milios, "Semantic similarity methods in wordnet and their application to information retrieval on the Web," in *Proc. CIKM*, Bremen, Germany, Oct. 2005, pp. 10–16.
[12] C. Leacock and M. Chodorow, "Combining local context and WordNet similarity for word sense identification," *WordNet, Electron. Lexical Database*, vol. 49, no. 2, pp. 265–283, 1998.
[13] C. Merkle, "A certified digital signature," in *Proc. CRYPTO*, Santa Barbara, CA, USA, Aug. 1989, pp. 218–238.
[14] D. Eastlake and P. Jones, *US Secure Hash Algorithm 1 (SHA1)*, document RFC 3174, Sep. 2001.
[15] H. Hacigümüş, B. Iyer, and S. Mehrotra, "Providing database as a service," in *Proc. ICDE*, San Jose, CA, USA, Feb./Mar. 2002, pp. 29–38.
[16] G. Xu, H. Li, Y. Dai, K. Yang, and X. Lin, "Enabling efficient and geometric range query with access control over encrypted spatial data," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 4, pp. 870–885, Apr. 2019.
[17] J. Song, M. Zhao, and S. Long, "The minimum selection of crowdsourcing images under the resource budget," *Symmetry*, vol. 10, no. 7, pp. 256–277, Jul. 2018.
[18] S. Ku, L. Hu, C. Shahabi, and H. Wang, "Query integrity assurance of location-based services accessing outsourced spatial databases," in *Proc. SSTD*, Jul. 2009, pp. 80–97.

[19] H. Ren, H. Li, Y. Dai, K. Yang, and X. Lin, "Querying in Internet of Things with privacy preserving: Challenges, solutions and opportunities," *IEEE Netw.*, vol. 32, no. 6, pp. 144–151, Nov./Dec. 2018.
[20] M. Narasimha and G. Tsudik, "Authentication of outsourced databases using signature aggregation and chaining," in *Proc. DASFAA*, Singapore, Apr. 2006, pp. 420–436.
[21] H. Pang, L. Zhang, and K. Mouratidis, "Scalable verification for outsourced dynamic databases," *Proc. VLDB Endowment*, vol. 2, no. 1, pp. 802–813, Aug. 2009.
[22] H. Pang and L. Tan, "Verifying completeness of relational query answers from online servers," *ACM Trans. Inf. Syst. Secur.*, vol. 11, no. 2, pp. 5:1–5:50, May 2008.
[23] Y. Liu, P. Ning, and H. Dai, "Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures," in *Proc. S&P*, Washington, DC, USA, May 2010, pp. 286–301.
[24] L. Hu, W.-S. Ku, S. Bakiras, and C. Shahabi, "Spatial query integrity with Voronoi neighbors," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 863–876, Apr. 2013.
[25] A. Liu and S. Zhao, "High-performance target tracking scheme with low prediction precision requirement in WSNs," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 29, no. 4, pp. 270–289, Aug. 2018.
[26] D. Wu, B. Choi, J. Xu, and S. Jensen, "Authentication of moving top-k spatial keyword queries," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 4, pp. 922–935, Apr. 2015.
[27] R. Zhang, Y. Zhang, and C. Zhang, "Secure top-k query processing via untrusted location-based service providers," in *Proc. INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 1170–1178.
[28] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava, "Truth finding on the deep Web: Is the problem solved?" *Proc. VLDB Endowment*, vol. 6, no. 2, pp. 97–108, Aug. 2013.
[29] Y. Deng, Z. Chen, D. Zhang, and M. Zhao, "Workload scheduling toward worst-case delay and optimal utility for single-hop fog-IoT architecture," *IET Commun.*, vol. 12, no. 17, pp. 2164–2173, Oct. 2018.
[30] S. Benabbas, R. Gennaro, and Y. Vahlis, "Verifiable delegation of computation over large datasets," in *Proc. CRYPTO*, Santa Barbara, CA, USA, Aug. 2011, pp. 111–131.
[31] D. Fiore and R. Gennaro, "Publicly verifiable delegation of large polynomials and matrix computations, with applications," in *Proc. CCS*, Raleigh, NC, USA, Oct. 2012, pp. 501–512.
[32] H. Li, D. Liu, Y. Dai, T. H. Luan, and S. Yu, "Personalized search over encrypted data with efficient and secure updates in mobile clouds," *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 1, pp. 97–109, Jan./Mar. 2018.
[33] H. Pang and K. Mouratidis, "Authenticating the query results of text search engines," *Proc. VLDB Endowment*, vol. 1, no. 1, pp. 126–137, 2008.
[34] H. Li, D. Liu, Y. Dai, T. H. Luan, and X. S. Shen, "Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage," *IEEE Trans. Emerg. Topics Comput.*, vol. 3, no. 1, pp. 127–138, Mar. 2015.
[35] W. Sun *et al.*, "Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 3025–3035, Nov. 2014.
[36] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 3, pp. 312–325, May/Jun. 2016.
[37] J. Shi, R. Zhang, and Y. Zhang, "Secure range queries in tiered sensor networks," in *Proc. INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 945–953.

YIZHU ZOU received the B.S. degree in computer science and technology from Hunan Agricultural University, in 2015. He is currently pursuing the master's degree in software engineering with Central South University. His research interests include security and privacy issues in social network, cloud computing, and big data. He is a Student Member of the IEEE.

XIN YAO received the B.S. degree in computer science from Xidian University, in 2011, and the M.S. degree in software engineering and the Ph.D. degree in computer science and technology from Hunan University, in 2013 and 2018, respectively. From 2015 to 2017, he was a Visiting Scholar with Arizona State University. He is currently an Assistant Professor with Central South University. His research interests include security and privacy issues in social network, the Internet of Things, cloud computing, and big data. He is a member of the IEEE and CCF.

ZHIGANG CHEN received the B.E., M.S., and Ph.D. degrees from Central South University, China, in 1984, 1987, and 1998, respectively. He is currently a Professor, a Ph.D. Supervisor, and the Dean of School of Software, Central South University. He is also the Director and an Advanced Member of the China Computer Federation (CCF), where he is also a member of the Pervasive Computing Committee. His research interests include the general area of cluster computing, parallel and distributed systems, computer security, and wireless networks.

MING ZHAO received the B.S. degree in mathematics from Hunan Normal University, in 1996, the M.S. degree in mathematics from the Huazhong University of Science and Technology, in 1999, and the Ph.D. degree in computer science and technology from Central South University, in 2007. From 2013 to 2014, he was a Visiting Scholar with Missouri State University. He is currently a Professor with Central South University. His research interests include network computing and machine learning. He is a member of the IEEE and CCF.