Edge Computing for User-Centric Secure Search on Cloud-Based Encrypted Big Data

Sahan Ahmad*, SM Zobaed*, Raju Gottumukkala†, and Mohsen Amini Salehi*
*School of Computing & Informatics
†Informatics Research Institute
University of Louisiana at Lafayette
Lafayette,Louisiana-70504, USA
Email: {sahan.ahmad1,sm.zobaed1, raju, amini}@louisiana.edu

***Abstract*—Cloud service providers offer a low-cost and convenient solution to host unstructured data. However, cloud services act as third-party solutions and do not provide control of the data to users. This has raised security and privacy concerns for many organizations (users) with sensitive data to utilize cloud-based solutions. User-side encryption can potentially address these concerns by establishing user-centric cloud services and granting data control to the user. Nonetheless, user-side encryption limits the ability to process (*e.g.,* search) encrypted data on the cloud. Accordingly, in this research, we provide a framework that enables processing (in particular, searching) of encrypted multi-organizational (*i.e.,* multi-source) big data without revealing the data to cloud provider. Our framework leverages locality feature of edge computing to offer a user-centric search ability in a real-time manner. In particular, the edge system intelligently predicts the user's search pattern and prunes the multi-source big data search space to reduce the search time. The pruning system is based on efficient sampling from the clustered big dataset on the cloud. For each cluster, the pruning system dynamically samples appropriate number of terms based on the user's search tendency, so that the cluster is optimally represented. We developed a prototype of a user-centric search system and evaluated it against multiple datasets. Experimental results demonstrate 27% improvement in the pruning quality and search accuracy.**

***Index Terms*—Edge Computing, User-based Sampling, Markov Chain, Privacy-Preserving Big Data, Encrypted Clustering.**

I. Introduction

Cloud services provide flexible and scalable solutions to host and process big data. The number of businesses and individuals using cloud solutions for big data is skyrocketing and the volume of data stored on cloud is expected to surpass 44 ZB in near future [1]. However, data privacy concerns still preclude many businesses, specially those with sensitive and confidential data (*e.g.,* criminal reports or financial documents), to utilize cloud services. One noticeable recent data privacy incident was *Panama paper leaking* in which around 11.5 million documents of the Panama paper were leaked. The documents contained detail financial and attorney-client information for more than 214,488 offshore bank entities including actors, politicians, athletes, and businessmen [2]. Thus, data privacy has remained a major concern in using cloud services to host data [3].

Organizations are in need of solutions that securely store their large document sets in the cloud and enable operations (*e.g.,* real-time search) on them without revealing the documents to unauthorized and malicious users. With the prevalence of thin-clients (*e.g.,* smart-phones), users expect lightweight solutions that impose a minimal load on the user device. Particularly for big data, sophisticated search methods, such as semantic search [4], are desired. The reason is that in big data a concept can be potentially expressed in various ways. For a given search query, *semantic search* can find all documents that contain phrase(s) conceptually or ontologically related to the search query [5].

The motivation of this study comes from multiple law-enforcement departments (*e.g.,* detectives, criminologists, and sheriffs) desire to perform semantic search on a large document set of criminal reports [6], stored in a cloud environment. These documents are confidential and access to them must be limited to the law-enforcement officers who need to search the data using their hand-held devices in a real-time manner. As such, the law-enforcement governing body desires a solution to protect confidentiality of the big document set against external or internal attackers in the could, while maintaining the semantic search ability.

User-side encryption [7], [8] is an approach to achieve data privacy against internal and external attacks [9], [10]. In this approach, documents are encrypted with the user key, before outsourcing them to the cloud and only users possessing the encryption key can decrypt the documents. However, the problem in user-side encryption is losing the ability to perform operations (*e.g.,* semantic search) on the encrypted documents. Another problem, which is particularly prominent for big datasets is to maintain the real-time response in processing the encrypted documents.

Searchable encryption systems (*e.g.,* [11]) enable search over encrypted documents. These systems generally maintain an index to map the keywords with their associated document [12]. However, with big datasets, the index size can become extensively large [13] and the real-timeness of the system is affected. To make searchable encryption systems scalable, solutions are provided to partition the encrypted keywords of the index structure into several clusters based on the topical relatedness of keywords [7] [14]. Then, for a given search query, a *pruning* method is used to limit the search space only to clusters relevant to the query [5]. However, the current searchable encryption systems for big data introduce two challenges:

1) Although pruning method reduces the search time, it impacts the search precision because of covering only a subset of the indexed keywords. In fact, pruning is achieved based on *abstracts* that include sample keywords from each one of the encrypted clusters. The abstracts are utilized on the user side in an unencrypted

form to navigate the search only to relevant clusters. To improve search precision, the sampled keywords of an abstract should precisely capture the topics of a corresponding cluster. In addition, sampling quantity for an abstract (*i.e.,* number of keywords in an abstract) must reflect user tendency in searching within a particular cluster.

2) The search systems introduce a significant overhead to the user's thin-client. Several resource-intensive components of the search system (*e.g.,* creating and updating abstracts, document encryption, keyword extraction, and pruning) reside on the user side that overwhelmes the user's thin-client.

To address these challenges, in this study, we propose a method to dynamically sample from each encrypted cluster and form a summary (*i.e.,* abstract) structure that qualitatively represent topics of each cluster and quantitatively reflect the user search interest to the cluster. As the keywords in the clusters are encrypted and we cannot infer anything about their semantics, we base the sampling method on the keywords and phrases already searched by the users. Our proposed method functions based on Markov Chain model [15] to analyze users' search tendency and nominate keywords whose inclusion in the abstracts can improve the search quality. Quantity of keywords in each abstract is determined based on the variety of search queries navigated to a given cluster.

Edge/fog computing has emerged to fill the gap between client machines and remote cloud datacenters. As we move from the cloud to user premises, communication latency decreases but trustworthiness increases [16]. There are edge computing solutions, such as Fortivault produced by Fortinet ltd [17], that act as a gateway between the user premises and cloud and its goal is to secure accessing to cloud datacenters. In such solutions, because edge machines are on the user's premises, both the client and edge machines are deemed trustworthy. We leverage the edge computing machines to eliminate the overheads of creating abstracts and other search components (*e.g.,* keyword extraction) from the user side device (thin-client). In addition, edge computing helps in reducing redundant processing needed to generate abstracts for users within the same department and with similar search interests.

In summary, contributions of this study are as follows:

- Developing an edge computing framework that enables lightweight semantic search on encrypted unstructured big data, hosted in cloud, in a real-time manner.
- Proposing a theory to efficiently sample from encrypted clusters. The theory enables sampling each cluster to an abstract whose quality of its elements represents topics of the cluster and quantity of it elements reflects the user search tendency.
- Evaluating and analyzing the search quality of the proposed framework against existing and established methods.

The rest of the paper is organized as follows: In Section II, we review recent studies undertaken in this area. Then, Section III provides an overview of the system architecture and explains functionality of each component. Section IV describes the proposed theory of our solution. Section V provides security analysis for edge tier. Section VI evaluates the performance of the proposed theory based on real-world datasets. Finally, Section VII concludes the paper and explains avenues of future work.

II. Related Work

In this section, we review research studies recently undertaken in cloud-based search solutions and position our contribution with respect to them. More specifically, we consider four related areas, namely user behavior model in information retrieval, user search pattern analysis, sampling from encrypted data, Secure Semantic Search.

User Behavior Model in Information Retrieval. To guess user intention while searching a query in the system, there has been extensive research work to analyze user behavior. Different models have been applied to analyze user behavior as an example reinforcement learning [18], Neural Model [19]. The final goal is to retrieve highly related documents on top ranks based on user search. The user behavior can vary in a different domain. As an example, Palotti *et al.,* [20] analysis logs from different medical sources and shows that the search behavior varies from medical professionals to laypeople. Though our research objective does not limit to a specific domain, it contains the encrypted unstructured document sets on the cloud server and the data set are divided on different cluster based on relevancy. The precision of the retrieved document highly dependent on finding out the relevant cluster after a search is performed by the user. As, we are able to extract only data volume information from the cloud server, we consider analyze user behavior is the appropriate option to project the data characteristics of a cluster.

User Search Pattern Analysis. In the information storage and retrieval area, several studies have been undertaken on analyzing the users' search behavior (*e.g.,* [21]). Dias and Vermunt [15] proposed a model-based clustering approach based on user search pattern to understand what type of information online market users demand in their interaction with the websites. They used Markov Chain to categorize a series of web pages visited by users and predict the probability that a user would visit a web page at a given time *t*. Ai *et al.,* [22] analysis email search behaviors and observed that user tend to find out specific keyword in the mail search rather than a generalized term. Benevenuto *et al.,* [23] analyzed user surfing behaviors in online social networks to understand the frequency and the time duration people spend on social media. They analyzed the streamed data to identify patterns of use in social networks. Rose *et al.,* [24] presented a framework to understand the goals of a user searching on the web. This research works focus on predicting users' behavior on the web. Alternatively, we are inspired to incorporate the idea of user behavior analysis on encrypted unstructured documents and clusters built on them. Another difference is in our objective, which is to utilize user search behavior knowledge to improve the accuracy of our search system.

Sampling from Encrypted Data. Sample size determination for plain-text data is well established [25], [26]. Many research works focus on domain-specific sampling (*e.g.,* for genomes data) and determining the minimum sample size based on the purpose of analysis in that domain [27]. Eng [28] proposed five parameters that can determine the appropriate sample size from a population. These parameters are namely, estimated measurement variability, desired statistical power, significance criterion, and the shape of planned statistical analysis. Minimum sample size determination depends on the way the data is collected and the nature of the data. To the best of our knowledge, there is no research work targeting sample size determination for encrypted clustered data, which is our focus in this research.

Secure Semantic Search There have been a few research works that expand the idea of semantic search over encrypted data [5], [7]. Research works based on a fuzzy keyword search on encrypted data have been undertaken to overcome the semantic search on encrypted data. Liu *et al.,* [29] proposed a secure search schema based on fuzzy matching of keywords in which the value of matching between two keywords varies between zero and one. They proposed a method to reduce the index size, hence, reducing the search time. Fu *et al.,* [30] also presented a method based on the semantic relationship between concepts in encrypted datasets. S3BD [5] is a semantic search system based on user-side encryption and pruning technique on a cluster of encrypted data to reduce the search time.Our research uses Woodworth's user-end encryption method, however, we leverage edge computing paradigm that enables lightweight client-side and provides us more control over the searched data to improve the search precision.

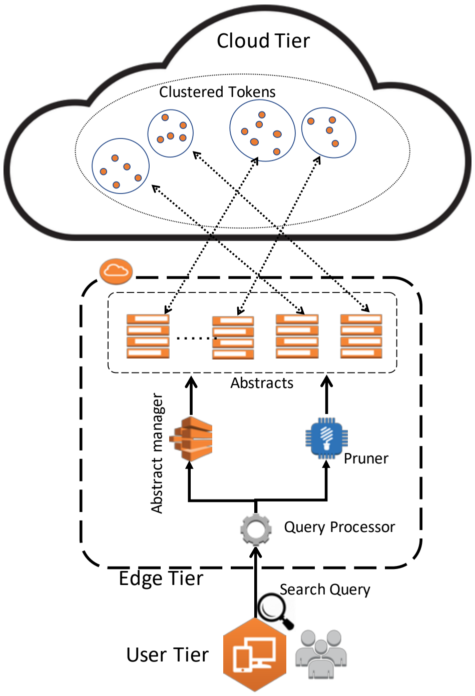

Fig. 1. Bird's eye-view of Edge-Based Architecture for Secure Search System Architecture and major processes.

III. Edge-Based Architecture for Secure Search Over Big Data

Our secure search system is composed of three tiers: *user tier*, *edge tier*, and *cloud tier*. Figure 1 provides a bird's-eye-view of primary components of each tier and major processes of the system. The details of each tier is as follows:

User Tier. User tier is composed of a lightweight interface to search over the encrypted data on the cloud. It is hosted on the users' thin-clients that assumed to be trusted. User's search query is sent to the edge tier. The search result is sent back to the user from the cloud provider in form of a list of documents ranked based on their relevance.

Edge Tier. The edge tier is a computing facility on the users' premises (organization), hence, considered to be trusted. *Query Processor* module receives the search query, pre-processes it by removing stop-words, stemming, splitting multi-phrase queries to capture partially-matching documents [7]. Then, the query is augmented to encompass the conceptual and antological semantics of the query [7]. *Abstracts* is a set of lists of unencrypted keywords that each one represents a sample of clustered keywords reside on the cloud. There is a one-to-one association between each list (*i.e.,* abstract) and a cluster. *Abstract Manager* is a module that monitors the search queries to identify users' search interest within an organization and accordingly updates the abstracts. This module operates periodically and in an offline manner. For a given search query, *Pruner* module determines the most related abstracts and limits the search operation only to the corresponding clusters. The edge tier (particularly, Abstract Manager and Pruner modules) is the main module explored in this paper (see Section IV).

Cloud Tier. It is a third-party cloud provider with large storage and computing services. We consider the cloud provider as *honest but curious*, hence, cannot be trusted. The cloud storage service is used to store encrypted documents and to cluster their encrypted keywords. Clustering is performed on the cloud based on the keywords extracted from documents and their co-occurrence frequency [5]. These information are extracted during the document upload process.

To avoid searching the entire dataset and perform effective pruning, clusters need to be topic-based, so that semantically related terms are co-located in the same cluster [31]. Because of encryption, extracting the topic of terms is not feasible, hence, topic-based clustering is carried out based on statistical semantics [32]. Particularly, in S3BD [5], for topic-based clustering of encrypted terms, terms' frequency of co-occurrences across documents is considered as the statistical semantics. In this case, we define *semantic relatedness* between two given terms based on the intersection of their appearances across documents.

Once the semantic relatedness between terms is defined, K-means algorithim [33] can be used to cluster the terms. For that purpose, k encrypted terms must be chosen as the centroids. Term ω is chosen as a centroid if the number of documents uniquely associated with ω is more than the number of documents it shared with other terms. Once the centroids are chosen, other terms are clustered with the centroid that they share the maximum semantic relatedness. Further details about clustering can be found in [5].

The cloud computing service is used to search over a subset of (*i.e.,* pruned) clusters, upon receiving a search request. Then, the results are ranked based on the relevance to the search query, before sending them to the user. Our objective, in this study, is to protect the data stored in the cloud from illicit access by internal and/or external attackers.

IV. Sampling from an encrypted cluster

A. Overview

Recall that there is a one-to-one association between an abstract and a cluster. In fact, each abstract is a representation of its corresponding cluster that resides on the edge tier, can be quickly compared against the search query, and navigates the search only to relevant clusters (*i.e.,* pruning).

Accurate pruning and choosing relevant clusters strongly depends on how representative the corresponding abstracts are. In fact, quantity and quality of sampled terms from a cluster determine representativeness of the abstract structure. The *quantity* refers to the sample size from the corresponding cluster, whereas the *quality* of the abstract depends on how the chosen terms conceptually express the cluster.

As for the optimal abstract (sample) size, we should note that, too small sample size impacts representativeness of the abstract, inaccuracy of pruning, and subsequently, the search precision. In contrast, too large abstract size increases the overhead and impacts the real-time quality of the search operation.

The search precision is impacted by the abstract quality (*sampling quality*). For a given search query, based on the semantic relatedness between abstracts and the query, the Pruner determines which clusters on the cloud must be excluded or included in the search operation. Therefore, each term chosen for an abstract should semantically represent a portion (*i.e.,* sub-topic) of the corresponding cluster and the union of such chosen terms in an abstract should represent topics of the entire cluster. Moreover, the abstracts should reflect the changes occur to the dataset (*e.g.,* because of document addition or removal). As such, the Abstract Manager should have the flexibility of dropping sampled terms that lose their importance over time and replacing them by more representative ones.

According to Figure 1, the Abstract Manager module in the edge tier is in charge of constructing and populating abstracts. As the terms in the clusters are encrypted, the Abstract Manager cannot use them to build abstracts. Therefore, we propose to infer the abstract items from users' search queries and their metadata. In particular, we propose to determine the quality and quantity of the abstract terms based on the following factors: *user search pattern*, *users' interest to a cluster*, and *number of cluster terms* (aka cluster size). In the rest of this section, we elaborate on the impact of these factors on the quality and/or quantity of terms that form an abstract. We also describe calculating these factors for a given cluster and determining the most representative terms for its corresponding abstract.

B. Analyzing Users' Search Pattern

Abstract Manager considers users' search pattern as an accessible metadata and leverages that for quality sampling that can semantically project the topics of the cluster. In addition, analyzing search pattern enables us to learn about search queries that potentially appear in near future. This knowledge can be leveraged to improve the precision of pruning and subsequently the search operation.

To predict users' search queries, we develop a method based on Markov Chain model, because it can describe a sequence of possible events where the probability of each event depends only on the state attained in the previous event. For that purpose, we create an adjacency graph that contains the sequence of search queries made by users to a particular cluster and leverage that to predict search queries that can potentially appear in the near future.

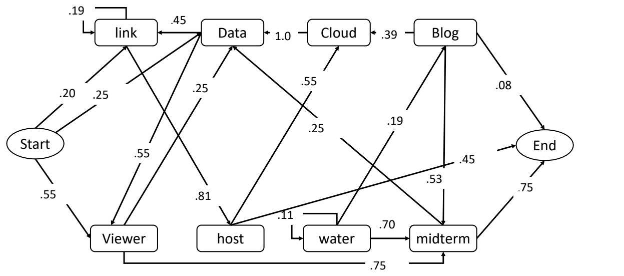

Fig. 2. An overview of user search pattern analysis using Markov Chain. The states represent user search terms and the arrows and their weights represent the transition probability from one state to another. The ovals denote the start and end of an interval.

Let W be the set of m searched terms (we have $W : \{w_1, w_2, ..., w_m\}$) appear in the system during a certain time interval and hit a particular cluster c. Elements of W form the states of a Markov Chain. Figure 2 represents Markov Chain in form of a graph where each vertex w_k shows a search term and each edge (w_k, w_j) shows the probability that term w_j is searched right after w_k. This probability between adjacent vertices is termed *transition probability* and denoted as P_{kj}. Then, the *n-step* transition probability between w_i and w_j, denoted $w_{ij}(n)$, is defined as the compound probability of searching term w_j, after n steps from searching w_i. Equation 1 formally shows how $w_{ij}(n)$ is calculated.

$$w_{ij}(n) = \sum_{k=1}^{m} w_{ik}(n-1) \cdot P_{kj} \tag{1}$$

The *state probability* of term w_i, denoted P_i^s, is defined as the likelihood of the term being searched in future. The initial value for the state probability of w_i is calculated based on the ratio of searching frequency for w_i to the total number of searched terms in cluster c.

For the set of all terms $w_i \in W$, the state probabilities form a matrix with m elements and transition probabilities form a $m \times m$ matrix. Then, the probability of searching terms in future is obtained by multiplying the state probability matrix with transition probability matrix iteratively, until the state probability matrix converges. The converged state probability matrix also indicates the appropriate terms that can be chosen for the abstract (called *qualified terms*). Specifically, the state probability of term w_i can be used to add or replace an term in the abstract structure of that cluster.

C. Users' Interest to a Cluster

To perform accurate pruning and navigate search to relevant clusters, in determining the quantity of terms in each abstract, we should consider the users' interest to the corresponding clusters. The clusters that are targeted more frequently (*i.e.,* higher users' interest) should be sampled more granularly (*i.e.,* have larger abstracts) and vice versa.

For a given encrypted cluster i, the number of search queries hit the cluster show its popularity and can be leveraged to infer the topics covered by that cluster. In particular, cluster popularity can be vary because the cluster covers several topics and/or because users are interested to the few topics of that cluster. Popularity of a cluster is measured with respect to the number of queries other clusters receive. In practice, distribution of queries across clusters is unbalanced [5]. Therefore, to measure popularity of a given cluster i, we use the ratio of deviation of number of queries hit cluster i to the mean number of queries each cluster should ideally receive. Let q_i the number of queries hit cluster i and $\bar{q}$ the average number of queries each cluster should receive. Then, *popularity* of cluster i is defined as: $\sigma_i = (q_i - \bar{q})/\bar{q}$.

Let $0 \leq \delta_i(a,b) \leq 1$ represent the semantic similarity between queries a and b that hit encrypted cluster i. Then, the average semantic similarity of all queries that hit the cluster (denoted $\bar{\delta_i}$) show variations of topics contained by cluster i. In particular, dissimilar search queries (*i.e.,* $\bar{\delta_i} \to 0$) imply that the cluster terms cover several topics. Alternatively, $\bar{\delta_i} \to 1$ indicates that the cluster terms are concentrated on a few topics.

We define users' interest to cluster i, denoted β_i, based on their interest to the topics of that cluster. As such, both the popularity of the cluster (σ_i) and the average semantic similarity of queries hit the cluster ($\bar{\delta_i}$) can be used to measure the users' interest to cluster i. Therefore, we calculate β_i as: $\beta_i = 1/(\delta_i + \sigma_i)$.

D. Cluster Size

For a given cluster i, the number of terms in that cluster (*i.e.,* cluster size) is positively correlated with the quantity of terms need to be sampled from that cluster. The relationship between population and sample size is not linear and it varies based on different application and data nature [27]. Hence, to determine the number of terms for the abstract representing cluster i, in addition to users' interest to cluster i, we consider the number of terms in the cluster (denoted as γ_i).

E. Procedure to Add or Remove Terms in Abstract Structures

Initially, abstracts are populated with equal number of (*e.g.,* ten) terms, chosen based on the number of document association in each cluster [5]. Then, while users issue search queries, the edge tier collects metadata and refines the abstracts' terms based on the factors mentioned in the previous part and by using the following main steps:

1) Analyzing users' search pattern and determine if a searched term is qualified to be placed in the abstracts or not.
2) If the term is qualified, then the appropriate abstract is selected.
3) Within the selected abstract, it is determined either to add the qualified term or replace it with an existing term.

In the rest of this section, we elaborate on each one of these steps.

1. Determining qualified terms to be placed in the abstracts. Successful search queries issued by users are examined against the Markov Chain model explained in Section 1 to learn their potential use in future. The terms that have a significant user interest can be considered as an abstract term and are called *qualified terms.*

2. Selecting appropriate abstract. To select an appropriate abstract for a qualified term, semantic similarity of the qualified term and existing members of each abstract is measured based on Wu Palmer method [34]. The abstract that offers the highest semantic similarity with the qualified term is selected as the target abstract. Ties are broken by selecting the abstract that is more often used, when the qualified term is searched.

3. Determining adding or replacing the qualified term in the selected abstract. Assume abstract i is selected for a qualified term w. Term w is added to abstract i, if w introduces a topic that is not covered by existing terms in the abstract. Otherwise, w replaces the most semantically similar term exist in the abstract i, only if it has a higher probability to appear in future than the existing abstract term. Accordingly, the decision to add or replace w in abstract i is made based on the cluster size(γ_i) and the user interest to the corresponding cluster (β_i). That is, the higher the cluster size and the more user interest for a cluster imply a larger abstract size.

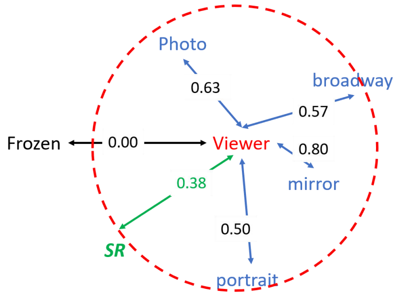

Fig. 3. Example of semantic radius (SR=0.38) for term *Viewer* in an abstract. Blue terms are within and black term (*Frozen*) is outside of the semantic radius. Numbers show semantic similarity between the *Viewer* and other terms.

We define *semantic radius* for abstract i, denoted SR_i, as the threshold semantic similarity to add or replace terms in abstract i. For a qualified term w, if it is within the semantic radius of an existing term t of abstract i (*i.e.,* $\delta_i(w,t) \geq SR_i$), then it implies that the topic w represents is already being covered by t. In this circumstance, the competition between w and t is resolved based on the future appearance probability of the two terms that is calculated based on the Markov Chain model. Alternatively, if w is not in the semantic radius of any term in the selected abstract i (*i.e.,* $\delta_i(w, \forall t \in i) < SR_i$), it implies that the topic has not been covered by the existing abstract terms. Hence, it has to be added to abstract i. The value of SR_i is calculated based on Equation 2.

$$SR_i = \frac{1}{\delta_i + \sigma_i + log(\gamma_i)} \tag{2}$$

Figure 3 visually shows and example of a semantic radius, where the value of SR is assumed to be 0.38. In this example, *Viewer* is an existing abstract term and *Frozen, Portrait, Broadway, Mirror,* and *Photo* are considered as qualified terms. Semantic similarity of all the qualified terms with *Viewer* are shown. The calculated semantic similarity of *Frozen* and *Viewer* is 0.00. Hence, *Frozen* represents a topic that is not covered by *Viewer* and its being added to the abstract. However, semantic similarity of other terms with *Viewer* fall within the semantic radius that implies topical overlapping. In this case, the term with the highest future appearance probability becomes a term in the abstract and other terms are discarded. It is noteworthy that the terms lose users' interest over time become less competitive and are replaced with qualified terms.

V. Security Analysis

Using clouds, data owner loses his/her full control over the outsourced data. Our threat model considers internal and/or external attacks to privacy of the outsourced cloud data and the goal is to preserve the data privacy. Our solution preserves data privacy by maintaining documents, their metadata (clustered terms), communication, and processing only in encrypted format. Hence, the attacker can learn nothing about documents and cluster tokens, once they are out of the user (trusted) premises.

Edge computing paradigm is predominantly used for low-latency communication [16]. From data control perspective, however, this paradigm brings multi-layer data control architecture to cloud-based storage solutions. From security perspective, edge computing resides at the user's premises, hence, can be considered trusted. For instance, Fortinet ltd. [17] has developed a secure edge platform, called Fortivault, that is deployed on a trusted hardware at the user's premises to facilitate secure access to the cloud [8]. Using edge-based platforms, the further data get from the users' machine, the less privacy it is provided with. However, as edge resources still reside in user's premises, they offer more control and privacy in compare to third-party clouds. In this study, our aim is to leverage the secure property of edge computing to enable accurate, real-time, and yet lightweight search solution while maintaining big data privacy.

We note that, even though edge is considered trusted, it does not have access to user's private key. It only contains the abstract structures to facilitates the search. In case, the edge is attacked, the attacker can only learn about the topics of each cluster and user's queries. However, documents and clusters terms remain private and secure.

VI. Performance Evaluation

A. Experimental Setup and DataSet

For performance evaluation, we implemented a prototype of the proposed system in the context of S3BD. We used two datasets, namely Amazon Common Crawl Corpus (ACCC) [35] and Request For Comments (RFC) [36], that have distinct characteristics and volumes. ACCC is $\approx$ 150 terabytes, contains web contents, and is not domain-specific. RFC is domain-specific and contains topics about Internet and network protocols. It includes 6298 documents and the size is $\approx$ 247 MB. All the experiments were carried out on 10-core 2.8 GHz E5-2680v2 Xeon processor with 64 GB memory. On this machine, we created three Virtual Machines (VMs) to represent client, edge, and cloud tiers.

To simulate users' search behavior and evaluate our system, we synthesized 1,000 benchmark queries for the RFC and 2,000 for the ACCC dataset. To synthesize queries from these datasets, we used Maui [37], a term extractor tool, to obtain 15 keywords from a subset of documents in each datasets. Then, we concatenate three keywords extracted from each document to generate one search query. We used 70% of the benchmark for learning users' search pattern and the rest of 30% queries for evaluating our proposed system in identifying the right cluster.

1) Quality of Clusters' Sampling in Abstracts: The quality of sampled terms (*i.e.,* abstracts) depends on how well those terms cover the topics within the whole cluster population. To evaluate the quality of sampling from the clustered terms, in Figure 4, we visually show the dispersion of sampled terms (red-cross symbols in the figure) throughout a cluster. In addition, we show the terms whose topics are captured with the abstract terms (shown in blue dots) and those topics that are not captured (shown in black dots).

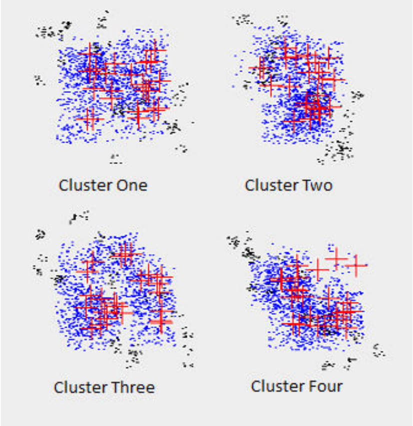

Fig. 4. Visualizing samples (abstracts) of four clusters of RFC dataset. Red-cross shows abstract terms of a cluster, blue and black dots, respectively, show terms whose topics are covered and not covered by the abstract terms.

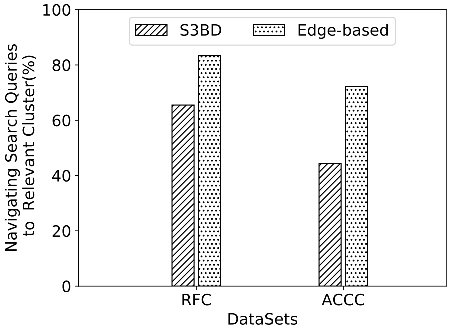

Fig. 5. Pruning accuracy of edge-based approach versus S3BD. The vertical axis shows the percentage of benchmark queries navigated to relevant clusters and the horizontal axis represents the evaluated dataset.

As we can see in this figure, most of the terms in the cluster are represented by the abstract terms and the abstract terms are well distributed throughout the clusters.

2) Evaluating Pruning Accuracy: The sampling (*i.e.,* abstract) quality from clusters is crucial for accurate pruning and targeting relevant clusters to search. To evaluate pruning accuracy, we run the benchmark queries and measure the percentage queries that hit the relevant clusters and compare it with the original approach in S3BD. The result of this experiment is shown in Figure 5 for different datasets. We observe that, for both RFC and ACCC datasets, our approach remarkably outperforms the original S3BD approach in pruning and navigating the search to relevant clusters. The difference between our edge-based approach and S3BD is more remarkable for ACCC dataset, because this dataset is not domain-specific and its topics are more diverse, so that they cannot be captured by only sampling from highly associated documents.

We also observe that pruning accuracy of the edge-based approach for the ACCC dataset is lower than the RFC dataset. The reason is that, in ACCC, there are numerous noun phrases (*e.g.,* for location and institution names) that has no general meaning. However, to keep generality in the abstracts, we opted out those specific terms. Hence, when such terms are searched, the pruner cannot navigate the search to relevant

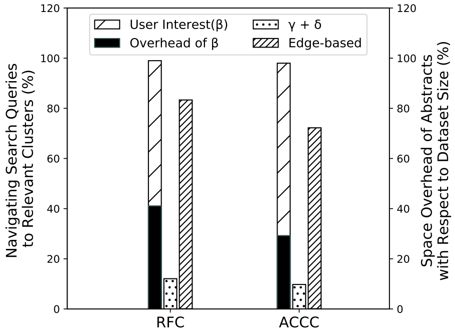

Fig. 6. The impact of abstract size in RFC and ACCC datasets on pruning accuracy and abstracts' overhead. The vertical axis shows both the percentage of queries navigated to relevant clusters and the abstracts' overhead.

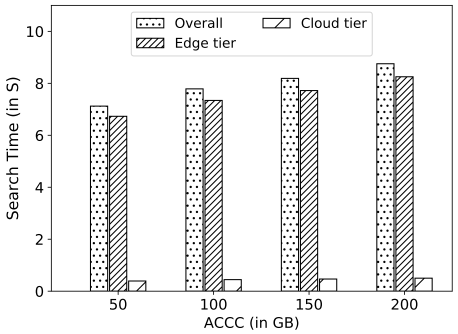

Fig. 7. Analyzing search time overhead of edge-based system. Overall search time, the time overhead for edge tier and cloud tier are reported. Vertical axis shows time (in S) and horizontal axis shows subsets of the ACCC dataset with different sizes (in GB).

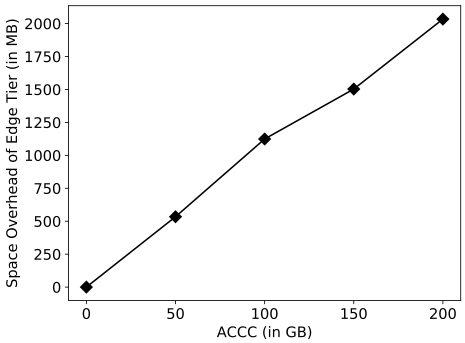

Fig. 8. Analyzing memory overhead of edge tier for datasets with different volumes. Vertical axis shows memory overhead of the edge tier (in MB) and horizontal axis shows subsets of the ACCC dataset with different sizes (in GB).

clusters. We can conclude that our edge-based approach particularly performs well in sampling general terms as opposed to specific ones.

3) The Impact of Abstract Size on Pruning Accuracy: The proposed abstract manager determines both the quality and quantity of terms to be sampled in abstracts. In this experiment, we concentrate on the quantity aspect of sampling, controlled by the semantic radius. We note that larger samples generally increase the pruning accuracy, however, they increase the overhead too. As such, in this experiment, we study the impact of different ways to determine semantic radius and their respective impacts on both the pruning accuracy and overhead. We name the semantic radius mentioned in equation 2 as the *edge-based* approach and compare it against two other approaches for the semantic radius: (A) user interest (β); and (B) cluster size plus variation of topics ($\gamma + \delta$).

As shown in Figure 6, for different datasets, we measure the percentage of benchmark queries navigated to relevant clusters. We also measure the abstracts' overhead based on the sample size to the total number of dataset terms.

In RFC and ACCC datasets, using user interest (β) semantic radius leads to $\approx$99% pruning accuracy, however, it creates prohibitively large abstracts ($\approx$41.07% and $\approx$29.23% of the dataset terms) that cannot be called samples anymore. In case of $\gamma + \delta$, we observe that the pruning accuracy is unacceptably low, because the abstract sizes are small and are not representative of the clusters' topics. Alternatively, in the edge-based approach, we observe that the pruning accuracy is relatively high and the overhead is negligible ($\approx$0.02%).

4) Analyzing Search Time and Space Overhead of the Edge-Based Approach: Although we showed that the proposed edge-based approach improves pruning accuracy and, subsequently, the search precision, the approach comes with an overhead. In this experiment, we elaborate on details of the induced overheads, both in terms of the search time and occupied space. The overall Search time in the edge-based approach is dominated by the time to search abstracts on the edge tier and the time to search clusters on the cloud tier. As such, to evaluate the search time overhead in this experiment, for each benchmark query, we measure the overall search time, and the overhead time on the edge and cloud tiers. To realize the overhead growth, we examined datasets with various volumes. We used ACCC dataset and created subset datasets ranging from 50 GB to 200 GB. To cluster these datasets, we used the method implemented in S3BD for clustering encrypted data. We created 100 benchmark queries that include one to three keywords and run them against the created datasets. Then, we measured the average of the overall search times, plus the average overhead time of on the edge and cloud tiers. The result of this experiment is shown in Figure 7. In this figure, the horizontal axis shows the size of different datasets created in GB. The vertical axis shows both the overall search time and the overhead of edge tier (in seconds).

Figure 7 shows the average of overall search time taken to complete the search queries. We observe that as the dataset size increases, the overall search time also increases. We also observe that the overall search time is dominated by the edge tier and not the cloud tier. The reason is that in S3BD, always three clusters are chosen to be searched, however, to find those three clusters, all abstracts must be checked against each search query. As the size of datasets increases, the number of terms in abstracts also increases, hence, the pruning operation dominates the search time.

In the second part of this experiment, we analyze the space overhead of the edge-based approach. The only space overhead imposed in this approach is in the edge tier, thus, we measured memory consumption in the edge tier for various dataset sizes (ranging from 50 GB to 200 GB). The result of this analysis is presented in Figure 8.

We observe that there is a linear relationship between the dataset size and the memory overhead of the edge tier because of the space consumed to maintain the users' search history that is used to analyze their search patterns and constructing effective abstracts. We note that, the space overhead of the edge tier is only about $\approx$0.01% of the size of the dataset stored in the cloud.

VII. Conclusion and Future Work

In this research, we developed an edge-computing-based framework that offers a user-centric search ability on encrypted big data in cloud. The framework is composed of three tiers namely, user tier, edge tier, and cloud tier. It enables real-time search over encrypted big data by pruning the search space and limiting the search to only relevant clusters of data. For accurate pruning, the edge tier leverages the users' search patterns and creates dynamic samples (aka abstracts) for each cluster. The framework determines quantity of items (terms) in each abstract and populates the abstract with terms that qualitatively represent topics of its corresponding cluster. We evaluated the pruning quality and navigating the search queries to relevant clusters by comparing the proposed framework against the one used in S3BD search system. Experimental results from different datasets show that the pruning quality is improved by $\approx$27%. This gain in performance is attained without imposing a major overhead to the system.

There are several avenues that this research can be extended. One avenue will be incorporating the impact of aging on historic users' search queries. Another avenue will be providing a method to evaluate quality of samples created from an encrypted cluster.

Acknowledgments

We thank reviewers of the manuscript. This research was supported by the Louisiana Board of Regents under grant number LEQSF(2017-20)-RD-B-06, and Perceptive Intelligence, LLC.

References

[1] Tx Zhuo, "https://www.entrepreneur.com/article/273561," Accessed Nov., 2018.

[2] N. Clark, "How a cryptic message, interested in data?, Led to the panama papers," Accessed December, 2018.

[3] M. Javanmard, M. A. Salehi, and S. Zonouz, "Tsc: Trustworthy and scalable cytometry," in *Proceedings of the 1st IEEE Big Data Security Conference*, Aug. 2015, pp. 1356–1360.

[4] R. Guha, R. McCool, and E. Miller, "Semantic search," in *Proceedings of the 12th international conference on World Wide Web*, May. 2003, pp. 700–709.

[5] M. A. S. J. Woodworth, "S3BD: Secure semantic search over encrypted big data in the cloud," *Journal of Concurrency and Computation:Practice and Experience (CCPE)*, vol. 28, Dec. 2018.

[6] L. ltd., "https://www.perceptiveintelligence.com/features," Accessed Nov. 21, 2018.

[7] J. Woodworth, M. A. Salehi, and V. Raghavan, "S3C: An architecture for space-efficient semantic search over encrypted data in the cloud," in *Proceedings of the 4th IEEE International Conference on Big Data*, ser. Big Data'16, Dec. 2016, pp. 3722–3731.

[8] M. A. Salehi, T. Caldwell, A. Fernandez, E. Mickiewicz, E. W. Rozier, S. Zonouz, and D. Redberg, "Reseed: Regular expression search over encrypted data in the cloud," in *Proceedings of the 7th International Conference on Cloud Computing (CLOUD)*, June 2014, pp. 673–680.

[9] N. Kaaniche and M. Laurent, "A secure client side deduplication scheme in cloud storage environments," in *Proceedings of the 6th International Conference on New Technologies, Mobility and Security*, Mar. 2014, pp. 1–7.

[10] W. Litwin, S. Jajodia, and T. Schwarz, "Privacy of data outsourced to a cloud for selected readers through client-side encryption," in *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, ser. WPES '11, Oct. 2011, pp. 171–176.

[11] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceedings of the Symposium on Security and Privacy*, ser. SP '00, May. 2000, pp. 44–58.

[12] J. Chou, M. Howison, B. Austin, K. Wu, J. Qiang, E. W. Bethel, A. Shoshani, O. Rübel, Prabhat, and R. D. Ryne, "Survey on secure search over encrypted data on the cloud," *Accepted in Journal of Concurrency and Computation:Practice and Experience (CCPE)in press*.

[13] J. Chou, M. Howison, B. Austin, K. Wu, J. Qiang, E. Bethel, A. Shoshani, O. Rübel, R. D. Ryne *et al.*, "Parallel index and query for large scale data analysis," in *Proceedings of international conference for high performance computing, networking, storage and analysis*, ser. SC '11, Nov. 2011, pp. 30:1–30:11.

[14] S. Zobaed, S. Ahmad, R. Gottumukkala, and M. Salehi, "Clustcrypt: Privacy-preserving clustering of unstructured big data in the cloud," in *Proceedings of the 21st International Conference on High Performance Computing and Communications, HPCC'19*.

[15] J. G. Dias and J. K. Vermunt, "Latent class modeling of website users search patterns: Implications for online market segmentation," *Journal of Retailing and Consumer Services*, vol. 14, no. 6, pp. 359–368, 2007.

[16] P. Patel, M. I. Ali, and A. Sheth, "On using the intelligent edge for iot analytics," *Journal of Intelligent Systems*, vol. 32, no. 5, pp. 64–69, 2017.

[17] "https://www.fortinet.com/," Accessed March, 2019.

[18] K. Athukorala, A. Medlar, A. Oulasvirta, G. Jacucci, and D. Glowacka, "Beyond relevance: Adapting exploration/exploitation in information retrieval," in *Proceedings of the 21st International Conference on Intelligent User Interfaces*, Mar. 2016, pp. 359–369.

[19] B. Mitra and N. Craswell, "Neural models for information retrieval," *arXiv preprint arXiv:1705.01509*, 2017.

[20] J. Palotti, A. Hanbury, H. Müller, and C. E. Kahn, "How users search and what they search for in the medical domain," *Journal of Information Retrieval*, vol. 19, no. 1-2, pp. 189–224, Oct. 2016.

[21] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay, "Accurately interpreting clickthrough data as implicit feedback," in *ACM Special Interest Group on Information Retrival SIGIR Forum*, vol. 5, Aug. 2005, pp. 154–161.

[22] Q. Ai, S. T. Dumais, N. Craswell, and D. Liebling, "Characterizing email search using large-scale behavioral logs and surveys," in *Proceedings of the 26th International Conference on World Wide Web, WWW '17*, Apr. 2017, pp. 1511–1520.

[23] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida, "Characterizing user behavior in online social networks," in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, Nov. 2009, pp. 49–62.

[24] D. E. Rose and D. Levinson, "Understanding user goals in web search," in *Proceedings of the 13th international conference on World Wide Web*, May. 2004, pp. 13–19.

[25] J. Sim, B. Saunders, J. Waterfield, and T. Kingstone, "Can sample size in qualitative research be determined a priori?" *Journal of Social Research Methodology*, pp. 1–16, Jun. 2018.

[26] S.-C. Chow, J. Shao, H. Wang, and Y. Lokhnygina, *Sample size calculations in clinical research*, 3rd ed. Chapman and Hall/CRC, 2017.

[27] R. C. MacCallum, M. W. Browne, and H. M. Sugawara, "Power analysis and determination of sample size for covariance structure modeling." vol. 1, no. 2, May. 1996, pp. 130–149.

[28] J. Eng, "Sample size estimation: how many individuals should be studied?" *Journal of Radiology*, vol. 227, no. 2, pp. 309–313, 2003.

[29] C. Liu, L. Zhu, L. Li, and Y. Tan, "Fuzzy keyword search on encrypted cloud storage data with small index," in *Proceeding of Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on*, Sep. 2011, pp. 269–273.

[30] Z. Fu, L. Xia, X. Sun, A. X. Liu, and G. Xie, "Semantic-aware searching over encrypted data for cloud computing," *Journal of Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2359–2371, Mar. 2018.

[31] A. Endert, S. Fox, D. Maiti, and C. North, "The semantics of clustering: analysis of user-generated spatializations of text documents," in *Proceedings of the 11th International Working Conference on Advanced Visual Interfaces*, May 2012, pp. 555–562.

[32] S. Wang and R. Koopman, "Clustering articles based on semantic similarity," *Journal of Scientometrics*, vol. 111, no. 2, pp. 1017–1031, 2017.

[33] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[34] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics,ACL*, Jun. 1994, pp. 133–138.

[35] Common Crawl on Amazon Web Services (AWS), "https://aws.amazon.com/public-datasets/common-crawl/," Accessed February, 2018.

[36] Request for Comments (RFC) Document Series, "http://www.ietf.org/rfc.html," Accessed February, 2018.

[37] O. Medelyan, E. Frank, and I. H. Witten, "Human-competitive tagging using automatic keyphrase extraction," in *Proceedings of the 14th Conference on Empirical Methods in Natural Language, EMNLP '09*, Aug. 2009, pp. 1318–1327.