

RESEARCH ARTICLE

# Privacy-Aware Relevant Data Access with Semantically Enriched Search Queries for Untrusted Cloud Storage Services

**Zeeshan Pervez<sup>1</sup>, Mahmood Ahmad<sup>2</sup>, Asad Masood Khattak<sup>3</sup>, Sungyoung Lee<sup>2</sup>, Tae Choong Chung<sup>4\*</sup>**

**1** School of Engineering and Computing, University of the West of Scotland, Paisley, PA1 2BE, United Kingdom, **2** Ubiquitous Computing Lab, Department of Computer Engineering, Kyung Hee University, Global Campus, 1 Seocheon-dong, Giheung-gu, Yongin-si, Gyeonggi-do 446-701, South Korea, **3** College of Technological Innovation, Zayed University, Abu Dhabi, United Arab Emirates, **4** Artificial Intelligent Lab, Department of Computer Engineering, Kyung Hee University, Global Campus, 1 Seocheon-dong, Giheung-gu, Yongin-si, Gyeonggi-do 446-701, South Korea

\* [tcchung@khu.ac.kr](mailto:tcchung@khu.ac.kr)



## Abstract

Privacy-aware search of outsourced data ensures relevant data access in the untrusted domain of a public cloud service provider. Subscriber of a public cloud storage service can determine the presence or absence of a particular keyword by submitting search query in the form of a trapdoor. However, these trapdoor-based search queries are limited in functionality and cannot be used to identify secure outsourced data which contains semantically equivalent information. In addition, trapdoor-based methodologies are confined to pre-defined trapdoors and prevent subscribers from searching outsourced data with arbitrarily defined search criteria. To solve the problem of relevant data access, we have proposed an index-based privacy-aware search methodology that ensures semantic retrieval of data from an untrusted domain. This method ensures oblivious execution of a search query and leverages authorized subscribers to model conjunctive search queries without relying on predefined trapdoors. A security analysis of our proposed methodology shows that, in a conspired attack, unauthorized subscribers and untrusted cloud service providers cannot deduce any information that can lead to the potential loss of data privacy. A computational time analysis on commodity hardware demonstrates that our proposed methodology requires moderate computational resources to model a privacy-aware search query and for its oblivious evaluation on a cloud service provider.

## OPEN ACCESS

**Citation:** Pervez Z, Ahmad M, Khattak AM, Lee S, Chung TC (2016) Privacy-Aware Relevant Data Access with Semantically Enriched Search Queries for Untrusted Cloud Storage Services. PLoS ONE 11 (8): e0161440. doi:10.1371/journal.pone.0161440

**Editor:** Yeng-Tseng Wang, Kaohsiung Medical University, TAIWAN

**Received:** March 11, 2016

**Accepted:** August 7, 2016

**Published:** August 29, 2016

**Copyright:** © 2016 Pervez et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** Data are available from <https://github.com/EDSResearch/SemanticallyEnrichedSearch>.

**Funding:** This work was supported by a grant from the Kyung Hee University in 2013 (KHU-20130439).

**Competing Interests:** The authors have declared that no competing interests exist.

## Introduction

We are living through a post-PC era in which computing facilities are regarded as the fifth utility [1]. These facilities, which are primarily related to computational and storage services, are provisioned to subscribers on a pay-as-you-go basis. This new service provisioning model is known as cloud computing [2]. Advances in virtualization technologies and the availability of

high-speed Internet have fostered this on-demand computing paradigm. It provides an abstraction of unlimited computational and storage facilities to its subscribers, enabling them to dynamically scale services or applications according to their specific requirements [3]. These on-demand and virtualized services are provisioned by a cloud service provider (CSP). The underlying cloud infrastructure (processing power, storage capacity, and networking facility) is owned, managed, and operated by a CSP. Subscribers do not need to take care of the cloud infrastructure, the assurance related to uninterrupted service provisioning is delineated in a service contract that is signed between the CSP and its subscribers.

Cloud-based storage service is a generalization of cloud-enabled data sharing, archiving, collaboration, and synchronization services [4]. These services leverage their subscribers to store their data for much a longer duration without the concerns of data availability and accessibility from varied devices, i.e., desktop computers, laptops, and smartphones. As lucrative as it sounds, there are data privacy concerns when confidential and personal data are outsourced to cloud-based storage services owned and managed by a CSP [5], [6], [7], [8]. Since these services are provisioned beyond the federated domain of subscribers over which they do not have any control, the CSP is considered to be an untrusted entity [9]. The most obvious solution to ensure data confidentiality in untrusted domain is to encrypt personal and confidential data before it can be outsourced to a cloud-based storage service. Since these services are provisioned on a pay-as-you-go basis, each data access request is charged according to the amount of data transferred between the subscriber and the CSP. Thus, the capability of a subscriber to access relevant encrypted data is very important. It ensures data privacy and can also increase the utility of the cloud-based storage services.

To access relevant data within the untrusted domain of a CSP, two main methodologies are employed, namely a search over encrypted data [10], [11], [12] and an index-based data search [13], [14], [15]. A search over encrypted data exploits the mathematical properties (trapdoors) of the cryptographic protocol to identify encrypted data that contain a particular keyword. These methodologies ensure the privacy of the outsourced data and the search query, preventing the CSP from deducing any information about the outsourced data that can lead to a potential loss of privacy. An index-based data search employs a different methodology than an encrypted data search. Instead of executing a search query over encrypted data, the search query is evaluated for the index (inverted index) associated with the outsourced data. Trusted entities can be employed to persist the index and evaluating the search query. In contrast to that, index can be stored in cloud storage in encrypted form along with the outsourced data and concealed search queries can be used to search the cloud.

The aforementioned methodologies provide accessibility to relevant data and also ensure data privacy. However, these methodologies are fairly limited in their functionality and greatly affect the utility of cloud-based storage services. A search over encrypted data can only search for predefined keywords for which trapdoors are defined, and in the case of data sharing and collaborative services, these trapdoors are shared among subscribers. In an index-based data search, where the index is used by a trusted entity, the cloud storage is underutilized for only the outsourced data, whereas the search queries that are handled by a trusted entity can only retrieve the data that have an exact match between the search criterion and index entries. Similarly, when the index is outsourced to cloud storage, the CSP can learn the access patterns of subscribers and can deduce confidential information about the outsourced data and subscribers. For instance, if the outsourced data of a patient is searched by a medical doctor specializing in diabetes mellitus, this leads to a possibility that either the outsourced data contains information regarding diabetes mellitus or that the patient is suffering from diabetes mellitus.

Considering the limitations of conventional methodologies to efficiently retrieve outsourced data taking data privacy into consideration, there is a need for a searching methodology that

can achieve semantic data retrieval and for an oblivious data search. Semantic data retrieval will ensure that relevant data can be discovered even if there is no exact match between the outsourced data and the search criteria defined by a subscriber. This will greatly increase the efficacy of the searching methodology in data sharing and collaboration services where the exact contents of the outsourced data are not known to participating subscribers, and only abstract ideas/concepts are communicated between them. For example, the employees of an insurance company who are collaborating on a task to define premium rates for next year's insurance policy, do not know the actual contents of the survey reports shared by their colleagues. However, they want to determine if there are any surveys on viral diseases in a certain vicinity. An oblivious search will lead to maximized utilization of the cloud infrastructure without relying on a trusted third party and will enable the CSP to evaluate the encrypted search queries.

In this research, we propose a privacy-aware content discovery methodology that enables subscribers of a cloud storage service to locate relevant data contents without using actual keywords from the outsourced data. It is an index-based privacy-aware data searching methodology that does not rely on a trusted third party to evaluate the search query. It realizes privacy-aware content discovery, which ensures that only authorized subscribers are able to search the outsourced data. It also prevents the CSP and unauthorized subscribers from learning the presence or absence of any keywords and deducing information that can lead to a potential loss of privacy, encompassing the outsourced data and the subscribers' personal information.

With the proposed methodology of privacy-aware content discovery, we make the following contributions in the area of cloud-based storage services:

- Privacy-aware search for encrypted data by utilizing semantic information to identify similarities between search criteria and outsourced data. The search criteria defined by a subscriber need not be exactly the same as in the outsourced data. If there exists a semantic relation between the search criteria and outsourced data, the relevant data contents can be retrieved;
- Privacy-aware data search without the need to share trapdoor information, and authorized subscribers can define their own search criteria. Their ability to access relevant data is not restricted to the information communicated by the data owner who outsources the data to the cloud storage;
- Maximized utilization of cloud storage services by persisting encrypted index and evaluated encrypted search queries within the domain of untrusted CSP; and
- Index and search query expansion by using semantic technologies to realize an encrypted data search similar to data contents for searching over the Internet.

The rest of this paper is organized as follows: Section 2 presents the related work. Section 3 defines the system design goals and the architectural and security model along with the assumptions. Section 4 is dedicated to the descriptive details of the proposed methodology. Section 5 discusses the implementation details, followed by evaluation of the results in Section 6. Section 7 presents the discussion on security, and Section 8 concludes the paper and discusses future directions.

## Related Work

In this section, we present methodologies to search encrypted data within an untrusted domain. Throughout this section, we mainly focus on cryptosystems, which exploit the mathematical properties of underlying cryptographic primitives to search encrypted data (i.e., trapdoor functions), and enterprise products, which define their protocols to match encrypted

search queries and data. We mainly discuss the effects of these conventional methodologies on the efficacy and utility of cloud-based storage services within the context of data sharing and collaboration services.

Symmetric key cryptography (SKC) enables a search over encrypted data [10] by utilizing a trapdoor defined for a particular key only to identify a match between a search query (trapdoor function) and encrypted data. SKC has been used in various schemes for searching over encrypted data, in which trapdoors are used to identify a match between the index of encrypted keywords instead of encrypted data [13], [14], [15]. However, the basic principles of the trapdoor's definition and the matching remain the same. A trapdoor-based search for public key cryptography (PKC) was proposed by Boneh et al. [11]. It leverages an untrusted server to search encrypted data using a public key, without the need to decipher concealed data. Schemes to search encrypted data that are based on SKC and PKC are limited in functionality because encrypted data can only be searched for keywords having corresponding trapdoors that are defined by the data owner who encrypts the data. Also, these trapdoors must be transmitted to authorized users, enabling them to access relevant data using search queries. Thus, methodologies relying on trapdoor-based cryptography assume guaranteed availability of the data owner or a trusted third party (TTP) to transmit a trapdoor to authorized users according to their access privileges.

To search confidential personal healthcare records, Li et al. proposed the Authorized Private Keyword Search (APKS) [12]. APKS utilizes Hierarchical Predicate Encryption (HPE) to realize a search over encrypted data [16] and employs a TTP to distribute capabilities (trapdoors) to authorized users according to their access privileges. These capabilities are then submitted to the CSP to evaluate the search query. Wang et al. proposed a methodology to rank search results according to their relevance with the selection criteria (trapdoor) [17]. However, it only supports a single trapdoor-based search query, greatly reducing its efficacy in defining the complex selection criterion and lacking the realism to search a large amount of data. A Searchable Cryptographic Cloud Storage System (CS2) focusing on dynamic data updates also provides a search over encrypted data [18]. Instead of searching the entire encrypted data repository, CS2 utilized the inverted index. However, CS2 is limited to cloud-based storage services and is not applied for cloud-based data sharing and collaboration services. Recently, Wenhai Sun et al. presented a privacy-preserving multi-keyword text search (MTS) with similarity-based ranking [19]. MTS utilizes tree-based indexing with adaption methods for a multi-dimensional algorithm. It ensures the confidentiality of the search query and the index data structure. However, it assumes that the user searching the cloud storage always behaves honestly, whereas the cloud server is honest but curious. This assumption can greatly affect the practicality of MTS for cloud-based storage services, focusing on data sharing and collaboration, in which users can behave maliciously to determine the presence or absence of a particular keyword(s). Oblivious Term Matching (OTM) realizes an encrypted index search, where the index is computed over encrypted outsourced data. OTM obviously evaluates encrypted search queries, where it does not consider relevant data access with consideration of semantic enrichment of the encrypted index or search queries [20]. The proposed methodology of semantic data search uses OTM to identify similarities between search criteria and outsourced data.

To achieve efficient data retrieval over large data contents, enterprises rely on search products that are customized to their specific needs and requirements. The Google search appliance [21] and Windows enterprise search products [22] offer such search solutions. These products create a searchable centralized enterprise-wide index that is used within the enterprise's data center or can be configured to use cloud repositories. The search queries are evaluated and the results are filtered according to the access privileges of a user. Since these products evaluate

access privileges after the execution of a search query, they require search services to be hosted within the federated domain of an enterprise. Thus, these search services retrain the migration of an enterprise-to-cloud ecosystem as it has to engage its own dedicated computation and storage resources for customizable search services. The authors in [23] have shown that, by carefully modeling search queries, malicious users can deduce confidential information from the centralized index, even if their access privileges do not allow them to access encrypted data.

The aforementioned methodologies for searching encrypted data focus on the confidentiality of the search query and the outsourced data. However, these methodologies do not consider the privacy of the query evaluation process that is employed to identify the relevant data contents. It can be exploited by a malicious CSP to deduce information that can lead to a potential loss of privacy. In cloud-based data sharing services, if multiple users are searching for a similar keyword, the CSP can effortlessly identify the importance of the outsourced data and can concentrate its malicious intents to deduce confidential information from the data. For instance, if the employees from the accounts and planning departments of an organization are searching data that has been outsourced to a folder called projected income statements, the CSP can determine the irregularity in access patterns and consequently affect the highly sensitive stock market, thereby disrupting the stock prices. Thus, a methodology that can obliviously search cloud-based repositories is of great importance, as it restrains the capability of a CSP to deduce or infer confidential information.

Fig 1 highlights the important features of existing methodologies for encrypted data search i.e., availability requirement for involved entities, entity responsible for evaluating the search query, and capability of a user to define arbitrary search queries. Although these methodologies realize encrypted data search, however their functionality is limited to exact matching between the search query and encrypted data i.e., trapdoors and encrypted index. Also these methodologies restrain authorized users to define their own search queries. In the subsequent sections, relevant data access with semantically enriched search queries is presented. The proposed methodology realizes semantic search enabling authorized users to define their own conjunctive search queries without compromising privacy of the outsourced data and search queries as well.

## Design Goals, System and Security Model, Main Idea, Assumptions and Notations

### System Design Goals

A data search within a cloud storage service allows subscribers to locate the required data contents. However, when data is outsourced to an untrusted domain of a public cloud service provider in encrypted form, standard search queries do not work, as the search criteria cannot be mapped to encrypted data. These search queries can also reveal confidential information about the outsourced data and the data owner. The design goal of our proposed system is to allow the subscribers of a public cloud storage service to search encrypted data in a similar way as contents are discovered over the Internet. However, search queries should not reveal any information to the cloud service provider, which can lead to the potential loss of privacy, affecting the outsourced data and personal information.

### System Model

To search encrypted data similar to content discovery works over the Internet, the public cloud storage service provider, repository owner, and content contributor are considered as the involved entities. For the sake of simplicity in the subsequent descriptive details, we refer to

	Availability requirement			Query execution		User defined search criteria
	Data owner	Storage service provider	Third party services / dedicated resources	Storage service provider	Third party services	
Searchable symmetric key cryptography [14]	X	X		X		
Privacy-preserving queries on encrypted data [15]	X	X		X		
Searchable public key cryptography [11]	X	X		X		
Authorized private keyword search [12]		X	X	X		
Secure ranked search over encrypted data [17]		X	X	X		
Searchable cryptographic cloud storage system [18]		X		X		
Privacy-preserving multi-keyword text search [19]	X	X		X		
Oblivious term Matching [20]		X		X		X
Google search Appliance [21], Windows enterprise search [22]		X	X		X	

**Fig 1. Features of conventional encrypted data search methodologies.**

doi:10.1371/journal.pone.0161440.g001

these entities as the cloud server, owner, and subscriber, respectively. The cloud server owns the cloud infrastructure (i.e., storage, computation, and network) and provisions its access on a subscription basis. The owner is a cloud storage subscriber who creates a shared repository that is accessible to other authorized subscribers. Subscribers contribute to the shared repository by outsourcing data contents. The owner and authorized subscribers search the cloud storage (shared repository) by submitting search queries to the cloud server. Search queries are obliviously evaluated by the cloud server, and the search results are provided to the respective entity according to its access privileges.

### Security Model

We consider the cloud server to be an untrusted entity that can collude with unauthorized subscribers to compromise the privacy of the outsourced data. It can assist unauthorized subscribers to search the outsourced data. Since the search query is evaluated by the cloud server, its result can be exploited to deduce confidential information about the outsourced data. To ensure the confidentiality of the outsourced data, only encrypted data is outsourced to the cloud server. In addition, to prevent the cloud server from inferring confidential information about the outsourced data, the encrypted search query is obliviously evaluated. This restrains the cloud server and unauthorized subscribers from learning of the presence or absence of a particular word or concept in the outsourced data.

## Main Idea

Suppose the Daily News is a nationwide newspaper that provides coverage of national and international events. Alice is a subeditor working for the Daily News. She oversees the department that focuses on financial corruption. At any particular point in time, she is working on multiple cases. She has assigned evidence collection and report compilation tasks to her subordinate journalists. To deal with the problem of content accessibility on her office and mobile devices, she has subscribed to a public cloud storage service that is provisioned by Eve. Since her subordinate journalists share confidential information with her, she does not want Eve to learn or deduce any information about the outsourced data. To ensure the privacy of the data, each journalist outsources encrypted data to the repository shared by Alice.

Bob and Mallory are Daily News journalists who work with Alice. Bob is an expert at retrieving information from online resources. Mallory's expertise is in finding the ground truth by contacting the concerned authorities. Both are directed to submit their findings on a financial scam that was recently exposed by the Fraud and Financial Crime Division of the State. Alice has provisioned access to both Bob and Mallory to a cloud-based shared repository. Bob retrieves all of the related information from online resources, whereas Mallory compiles her report using the information she has collected from the appropriate authorities. Before outsourcing their findings to a shared repository, they index the information. The index is then further enhanced by augmenting it with missing relevant information. After that, the findings and the enhanced index are encrypted and outsourced to the shared repository.

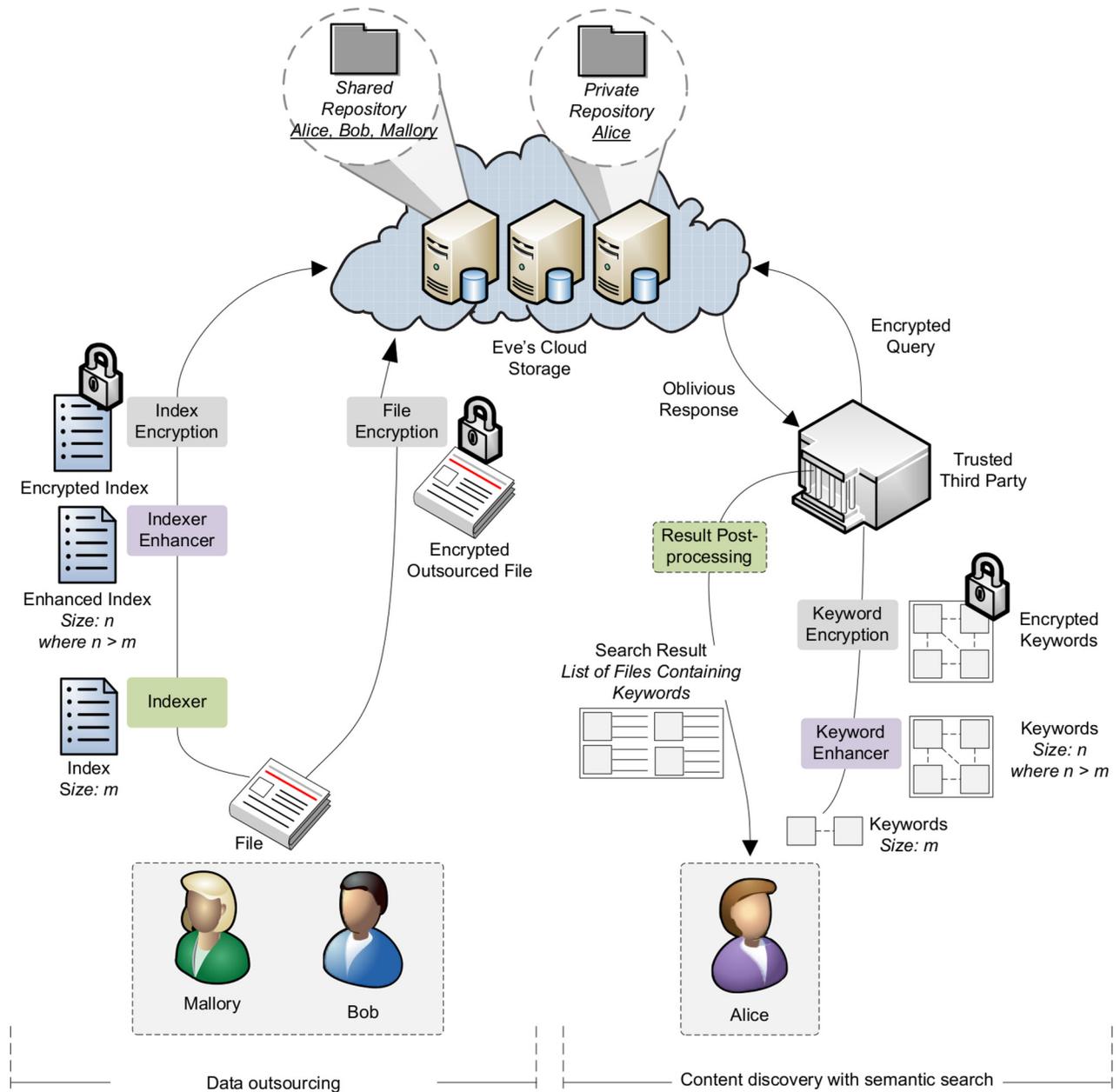
Whenever Alice needs to search for a file containing particular information, she defines a search criterion. The search criterion is then enriched by adding missing relevant information. The augmented search criterion is then encrypted with the secret key, and after that, Alice models an oblivious search query using the encrypted search criterion. The oblivious query is then submitted to the cloud server, which replies with the response.

Alice processes the cloud server response and determines the presence / absence of keywords that were defined in the search criterion. From the processes of query formation, evaluation, and post-processing of the result, the cloud server learns nothing about the outsourced data or the search query; the search evaluation is oblivious to the cloud server. If an unauthorized subscriber tries to search the repository, the proposed system generates a randomized response. [Fig 2](#) illustrates the conceptual model of our proposed system for searching encrypted data in an untrusted domain.

## Assumptions and Notations

The proposed system focuses on semantic search for encrypted data. We assume that the owner has shared a symmetric encryption key with the authorized subscribers. The data that is outsourced to a shared repository is always encrypted with that key. In the subsequent descriptive details of the proposed system, the specifics of sharing data within an untrusted domain are intentionally neglected for the sake of simplicity. Readers can refer to [\[24\]](#) and [\[25\]](#) for descriptions of efficient and secure data sharing within public cloud storage services. The proposed system address the problem of privacy-aware relevant data access in untrusted cloud storage services. Ensuring data integrity and correctness is beyond the scope of research undertaken in this work; interested reader can refer to [\[26\]](#) for more details on public auditing.

[Table 1](#) presents the notations used in the descriptive details of our proposed system to semantically search the encrypted data.  $\mathcal{F}$  represents a file that is outsourced to cloud storage.  $\mathcal{I}$  stands for an index computed over  $\mathcal{F}$  that contains the keywords and their respective frequencies i.e.,  $\mathcal{I} = \{ \langle kw_0, f_0 \rangle \dots \langle kw_n, f_n \rangle \}$ , where  $n$  is the size of the index.  $\mathcal{I}_s$  represents a semantic index that is generated by identifying synonyms and the root word for each  $kw_{0\dots n} : kw_i \in \mathcal{I}$



**Fig 2. Semantic search over encrypted data—conceptual model.**

doi:10.1371/journal.pone.0161440.g002

i.e.,  $\mathcal{I}_s = \{\langle kw_0, syn_{0,0}, rw_0, f_0 \rangle \dots \langle kw_n, syn_{n,0}, rw_n, f_n \rangle\}$ , where  $syn_{i,0}$  is the list of synonyms of  $kw_i$ , and  $rw_i$  is its root word.  $\mathcal{H}$  is an encoding function that is publicly known and encodes variable sized keywords into integer values of fixed length.  $\mathcal{E}_H$  and  $\mathcal{D}_H$  are homomorphic encryption algorithms.  $\sigma_{pk}$  and  $\sigma_{sk}$  are public and secret keys respectively, that are used by homomorphic encryption algorithms. These algorithms enable the processing of encrypted values (search query and encrypted index) without the need to decrypt them.  $\mathcal{E}_s$  and  $\mathcal{D}_s$  are symmetric encryption algorithms with a secret key  $k$ .  $\mathcal{F}$  and  $\mathcal{I}_s$  are encrypted with symmetric encryption algorithms before they can be outsourced to a cloud server.  $\mathcal{E}_A$  and  $\mathcal{D}_A$  are asymmetric encryption algorithms associated with  $k_{pub}$  and  $k_{pri}$  public and private keys, respectively.

**Table 1. Notations used in the descriptive detail of semantically enriched encrypted data search.**

Notation	Description
$\mathcal{F}$	File outsourced to a shared repository.
$\mathcal{I} = \{\langle kw_0, f_0 \rangle \dots \langle kw_n, f_n \rangle\}$	Index file that contains $n$ keywords.
$\mathcal{I}_s = \{\langle kw_0, syn_{0\dots v}, rw_0, f_0 \rangle \dots \langle kw_n, syn_{n\dots v}, rw_n, f_n \rangle\}$	Semantic index—an enriched form of $\mathcal{I}$ . $kw$ is a keyword from $\mathcal{I}$ , $syn_{0\dots v}$ is a list of its synonyms and $rw$ is its root/parent word.
$\mathcal{H}$	Publicly known encoding function that transforms an arbitrary-sized string to an integer value of $q$ modulo, where $q$ is a large prime.
$\mathcal{E}_H, \mathcal{D}_H$	Homomorphic encryption and decryption algorithms.
$\sigma_{pk}, \sigma_{sk}$	Public and secret key pair for homomorphic encryption algorithms.
$\mathcal{E}_A, \mathcal{D}_A$	Asymmetric encryption and decryption algorithms.
$k_{pub}, k_{pri}$	Public and private key pair for asymmetric encryption algorithms.
$\mathcal{E}_S, \mathcal{D}_S$	Symmetric encryption and decryption algorithms.
$k$	Secret key of symmetric encryption algorithms. It is shared with authorized users only.
$\alpha_{0\dots n}$	List of coefficients of a polynomial $P$ which defines a search query.
$\Delta_{y_{0\dots n}}$	List of oblivious values generated as a result of query execution by the cloud server.

doi:10.1371/journal.pone.0161440.t001

$\alpha_{0\dots n}$  represents a list of polynomial coefficients that are used to formulate an oblivious search query.  $\Delta_{y_{0\dots n}}$  is a list of oblivious values that are obtained as a result of the oblivious search query execution by the cloud server.

## Proposed System

The proposed methodology of encrypted data search based on semantically enriched index and search queries is presented in this section. It is divided into five cohesive steps: indexing, data outsourcing, query formulation, query execution, and post-processing of results.

### Indexing

A semantic search over encrypted data is achieved by evaluating search queries for an enriched inverted index ( $\mathcal{I}_s$ ) associated with the outsourced data ( $\mathcal{F}$ ). Since, we want subscribers to search outsourced data using search queries that are semantically equivalent, the inverted index ( $\mathcal{I}$ ) is augmented with extra information. This extra information enables us to identify outsourced data that contains relevant information instead of finding an exact match between the search query and the keywords extracted from the outsourced data.

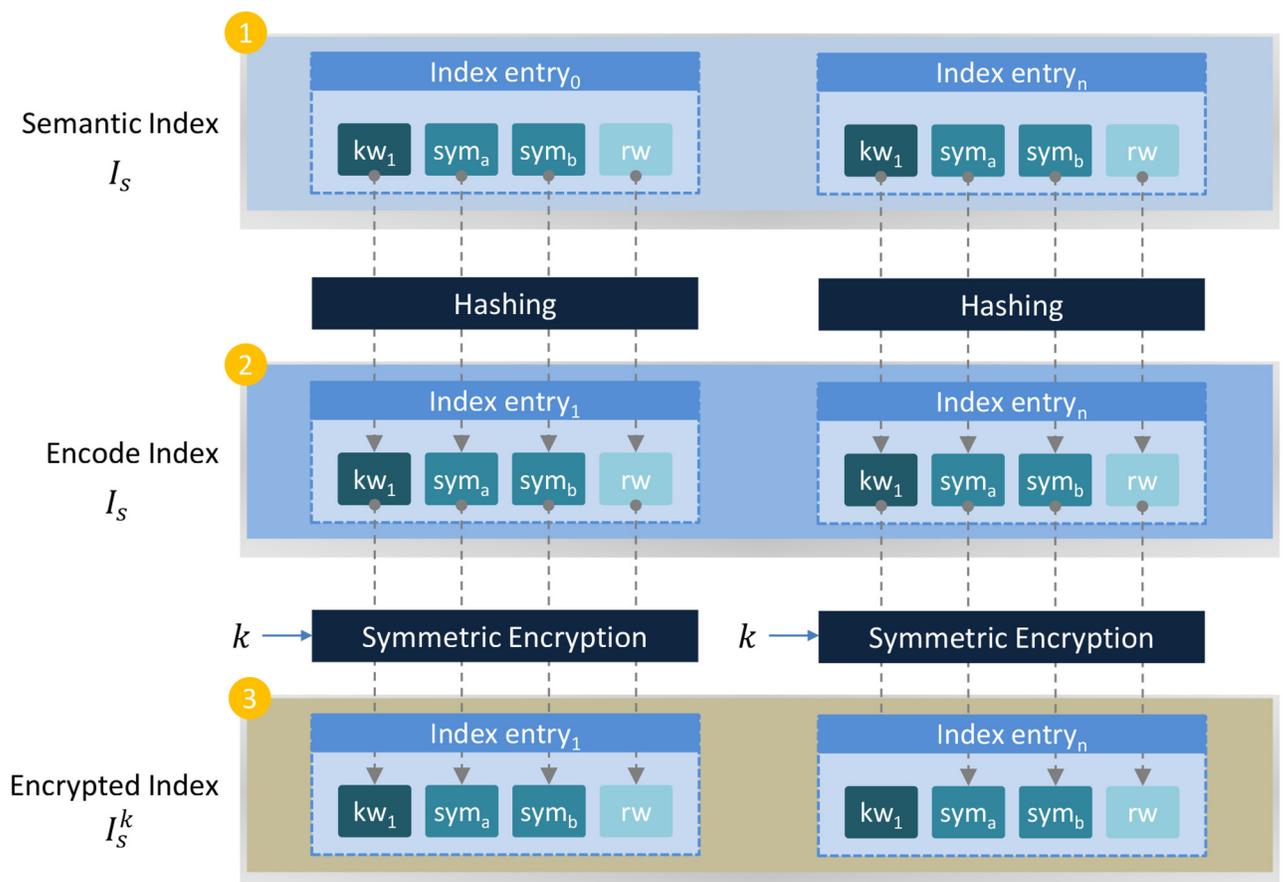
To achieve this, the indexing is further divided into two phases. In the first phase, for each  $\mathcal{F}$  that needs to be stored in cloud storage,  $\mathcal{I}$  is generated. It contains all of the keywords ( $kw_0 \dots kw_n$ ) that appear in  $\mathcal{F}$ , along with their respective frequencies i.e.,  $\mathcal{I} = \{\langle kw_0, f_0 \rangle, \dots \langle kw_n, f_n \rangle\}$ . After that  $\mathcal{I}$  is further processed to augment it with semantic information. For that,  $kw_{0\dots n} : kw_i \in \mathcal{I}$  are searched in a lexical database. This enables us to identify synonyms ( $syn_{0\dots v}$ ) of  $kw_i$  that do not exist in  $\mathcal{I}$  but where  $syn_{0\dots v}$  and  $kw_i$  semantically equivalent. Further root word ( $rw_i$ ) of each  $kw_i$  is also extracted from the lexical database.  $rw$  assists us in finding the keywords that share the same root word, consequently identifying the relevancy between the search query and the encrypted outsourced data. Once  $syn_{0\dots v}$  and  $rw$  are identified,  $\mathcal{I}$  is augmented with this extra information and is transformed into semantic

index i.e.,  $\uplus(\mathcal{I}, \text{syn}_{0\dots v}, \text{rw}) \rightarrow \mathcal{I}_s$ ; where  $\uplus(\cdot)$  is a function that appends  $\text{syn}_{0\dots v}$  and  $\text{rw}$  to  $\mathcal{I}$  removing any duplicate values, where  $\mathcal{I}_s = \{\langle kw_0, \text{syn}_{0\dots v}, \text{rw}_0, f_0 \rangle \dots \langle kw_n, \text{syn}_{n_0\dots v}, \text{rw}_n, f_n \rangle\}$ .

### Data Outsourcing

To ensure that the cloud server cannot exploit  $\mathcal{I}_s$  by deducing confidential information about the outsourced data.  $\mathcal{I}_s$  is concealed using a symmetric encryption algorithm before it can be outsourced to a cloud server. The secret key ( $k$ ) for the symmetric encryption algorithm is shared among all of the authorized subscribers by the owner who having ownership rights over the shared cloud based repository. The scope of this paper is limited to encrypted data search, readers may refer to [27] for more details secret key sharing and user revocation in untrusted domain.

In order to ensure that the search query can be obviously evaluated by the cloud server, the owner encodes each keyword ( $kw_i \in \mathcal{I}_s$ ) using a publicly known encoding function  $\mathcal{H}(\mathcal{I}_s) \rightarrow \hat{\mathcal{I}}_s$ .  $\hat{\mathcal{I}}_s$  is then encrypted using a symmetric encryption algorithm  $\mathcal{E}_s(\hat{\mathcal{I}}_s, k) \rightarrow \hat{\mathcal{I}}_s^k$ . Once the confidentiality of  $\mathcal{I}_s$  is ensured,  $\hat{\mathcal{I}}_s^k$  along with  $\mathcal{F}^k$  are outsourced to the cloud server. Since,  $k$  is only shared among the authorized subscribers and the owner, unauthorized subscribers cannot deduce any information about the outsourced data, even if they conspire with the cloud server. Fig 3 illustrates the entire process of securing inverted index with symmetric encryption.



**Fig 3. Encoding semantically enriched index and securing its confidentiality through symmetric encryption.**

doi:10.1371/journal.pone.0161440.g003

### Query Formulation

To search encrypted data, a subscriber defines a search criterion ( $\mathcal{C}_{kw_{0..j}}$ ), which consists of a set of keywords ( $kw_{0..j}$ ) that are used to search the relevant encrypted outsourced data. Since, we want to realize a semantic search over encrypted data, the search criteria defined by the subscriber is further enriched by identifying synonyms of  $kw_{0..j}$  :  $kw_i \in \mathcal{C}_{kw_{0..j}}$ . Once the relevant keywords ( $syn_{0..v}$ ) are identified  $\mathcal{C}_{kw_{0..j}}$  is enriched by adding  $syn_{0..v}$  to  $\mathcal{C}_{kw_{0..j}}$  i.e.,  $\uplus(\mathcal{C}_{kw_{0..j}}, syn_{0..v}) \rightarrow \mathcal{C}_{kw_{0..l}}$ , where  $j < l$ , and  $\uplus(\cdot)$  is a function that appends  $syn_{0..v}$  to  $\mathcal{C}_{kw_{0..j}}$ .

Since, search queries are evaluated by the cloud server, there is a need to conceal  $\mathcal{C}_{kw_{0..l}}$  using an appropriate symmetric encryption algorithm. To prevent the cloud server from deducing any information about the encrypted outsourced data, the owner encodes  $\mathcal{C}_{kw_{0..l}}$  using a publicly known encoding function. For example,  $\mathcal{H}(\mathcal{C}_{kw_{0..l}}) \rightarrow \hat{\mathcal{C}}_{kw_{0..j}} - \mathcal{H}(\cdot)$  must be the same encoding function as that used in the data outsourcing; otherwise, an oblivious search query cannot be successfully evaluated. After that  $\hat{\mathcal{C}}_{kw_{0..l}}$  is encrypted with the symmetric encryption i.e.,  $\mathcal{E}_s(\hat{\mathcal{C}}_{kw_{0..l}}, k) \rightarrow \hat{\mathcal{C}}_{kw_{0..l}}^k$ , where  $k$  is a shared symmetric encryption key, which is the same as that used in the data outsourcing to conceal  $\hat{\mathcal{I}}_{kw_{0..n}}$ .

To this stage,  $\hat{\mathcal{C}}_{kw_{0..l}}^k$  has been concealed, however in order to realize an oblivious query evaluation there is a need to further process  $\hat{\mathcal{C}}_{kw_{0..l}}^k$ . A polynomial ( $P(x)$ ) is defined such that the concealed  $kw_{0..n}$  :  $kw_i \in \hat{\mathcal{C}}_{kw_{0..l}}^k$  are the root of  $P(x)$  i.e.,  $P(x \in \hat{\mathcal{C}}_{kw_{0..l}}^k) = \sum_{i=0}^l \alpha_i x^i = 0$ , where  $\alpha_{0..l}$  are the coefficients of  $P(x)$ .

Once the polynomial  $P(x)$  has been defined, a homomorphic encryption key pair ( $\sigma_{pk}, \sigma_{sk}$ ) is initialized. Homomorphic encryption enables the cloud server to process the encrypted search query and also restrains its ability to learn the result of the query evaluation. After that,  $\alpha_{0..l}$  are encrypted i.e.,  $\mathcal{E}_H(\alpha_{0..l}, \sigma_{sk}) \rightarrow \alpha_{0..l}^{\sigma_{sk}}$ ,  $\alpha_{0..l}^{\sigma_{sk}}$  along with  $\sigma_{pk}$  are transferred to the cloud server.  $\alpha_{0..l}^{\sigma_{sk}}$  are used as the encrypted search query whereas  $\sigma_{pk}$  enables the evaluation of an encrypted search query without the need to decipher  $\hat{\mathcal{I}}_s^k$  and  $\alpha_{0..l}^{\sigma_{sk}}$ . Fig 4 describes the entire process of query formation.

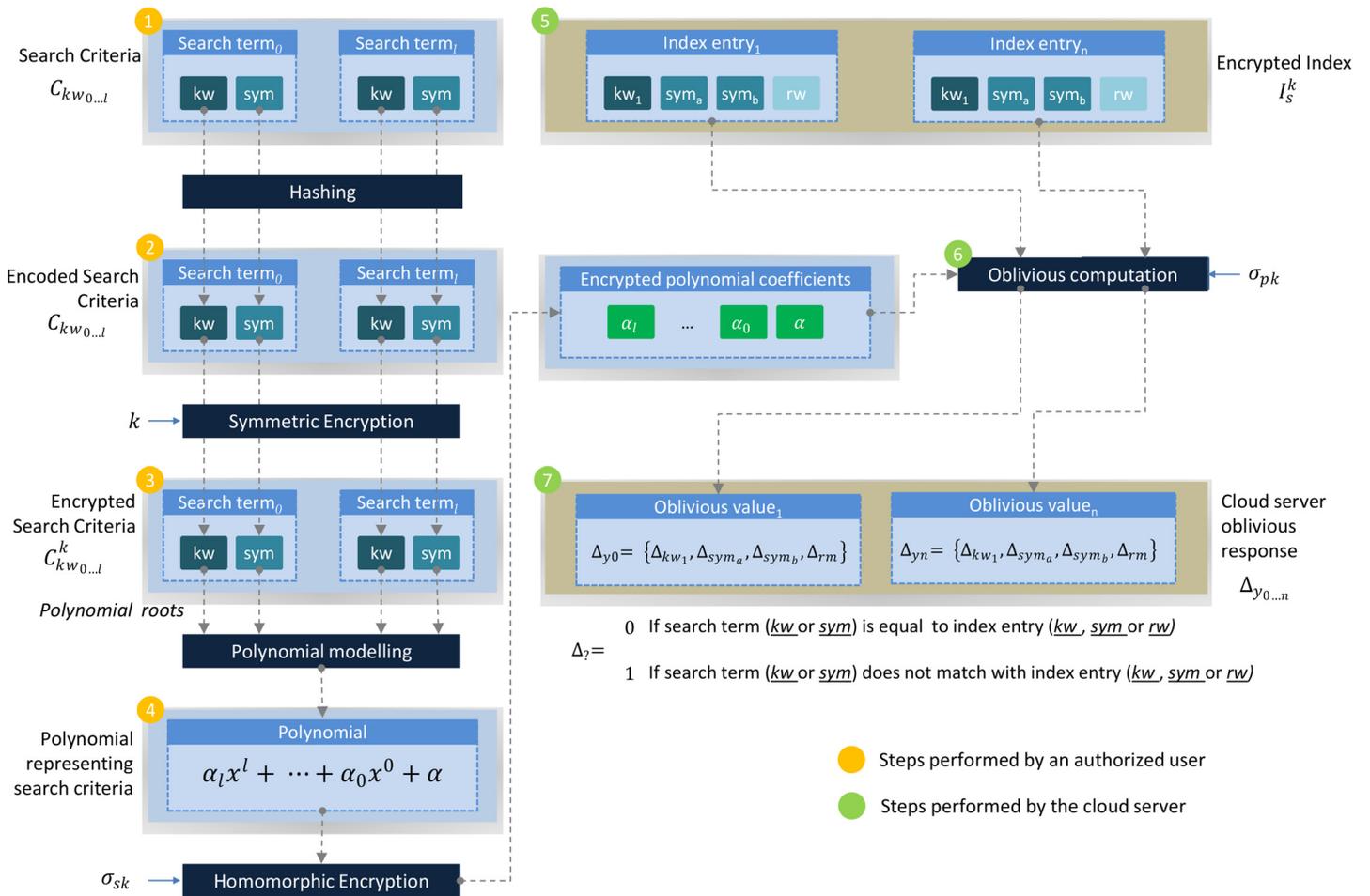
### Query Execution

To semantically identify the encrypted data, the search query is obliviously executed by the cloud server.  $\alpha_{0..l}^{\sigma_{sk}}$ , which is submitted by a subscriber, is evaluated for  $\hat{\mathcal{I}}_s^k$ . By using  $\sigma_{pk}$ , for  $kw_{0..n}$  :  $kw_i \in \hat{\mathcal{I}}_s^k$ , the cloud sever computes the oblivious value i.e.,  $\Delta_{y_{0..n}} = r.P(y_i \in \hat{\mathcal{I}}_s^k)$ , where  $y_i = kw_i \in \hat{\mathcal{I}}_s^k$  and  $r$  is a random number. The computation of the oblivious value ensures that the owner can identify the match between  $\alpha_{0..l}^{\sigma_{sk}}$  and  $\hat{\mathcal{I}}_s^k$ . Fig 4 describes the oblivious query execution process.

Since, we are employing homomorphic encryption, the cloud server cannot learn whether  $kw_i \in \hat{\mathcal{I}}_s^k$  is a root of  $\alpha_{0..l}^{\sigma_{sk}}$ . Thus, it cannot identify a match between the encrypted search criterion and the encrypted index. Once  $\Delta_{y_{0..n}} = r.P(y_{0..n})$  are computed, the cloud server transfers the result of the search query evaluation to the subscriber.

### Post-processing of results

The oblivious values that the subscriber receives from the cloud server can only be deciphered using the valid homomorphic key, which is the secret key,  $\sigma_{sk}$ . This ensures that the cloud server cannot collude with malicious subscribers to exploit the oblivious query evaluation



**Fig 4. Encoding semantically enriched search criteria and modeling search query for oblivious computation.**

doi:10.1371/journal.pone.0161440.g004

process. On receiving the cloud server’s response, the subscriber deciphers  $\Delta_{y_{0...n}}$  i.e.,  $D_H(\Delta_{0...n}, \sigma_{sk}) = \psi_{0...n}$ , where  $\psi_i$  can be a zero or non-zero randomized value.

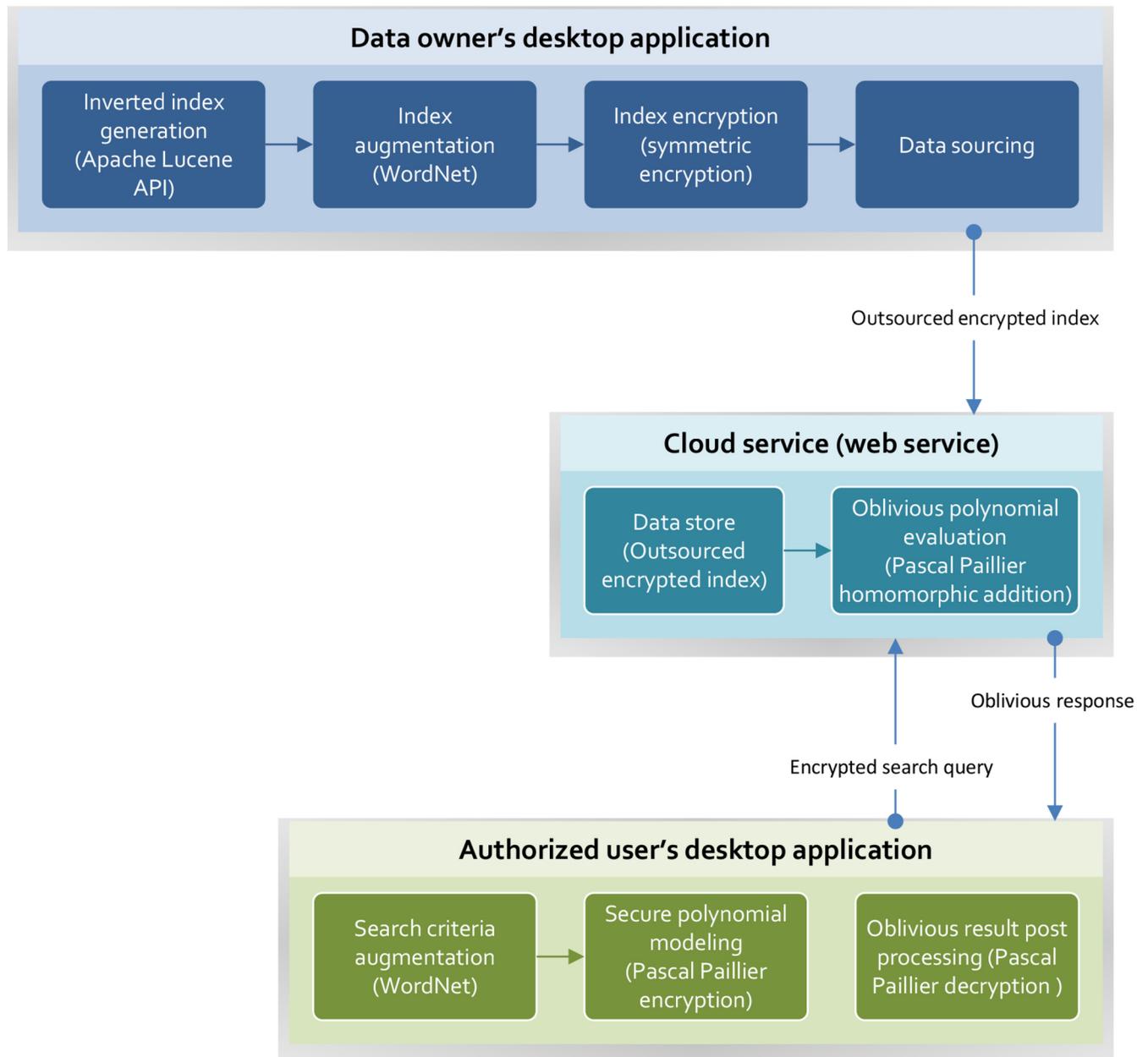
Since, the search query  $\sigma_{sk}^k$  submitted by the subscriber is constituted of root values from  $\hat{C}_{kw_{0...l}}^k$  the decryption of  $\psi_i$  turns out to be zero for all those  $y_i = kw_i \in \hat{I}_S^k$  that are equal to the root value of  $P(y_i)$ , i.e.,  $kw_i \in \hat{C}_{kw_{0...l}}^k \wedge kw_j \in \hat{I}_{kw_{0...n}}^k$  where  $kw_j = kw_i$ . For all other values where  $kw_j \neq kw_i$ ,  $\psi_i$  would turn out to be a random value (see Eq 1).

$$P(y) = \sum_{i=0}^l \alpha_i y^i \begin{cases} = 0 & \text{if } y \text{ is root of } P(y). \\ \neq 0 & \text{a random value } r \text{ for all other index entries.} \end{cases} \quad (1)$$

Thus, only by deciphering  $\Delta_{y_{0...n}}$  with valid  $\sigma_{sk}$  owner can learn the result of encrypted search query. However, for the cloud server the evaluation of the search query will remain oblivious.

### Implementation

The proposed methodology for a semantically enabled search of encrypted data is realized using jdk 1.7. We implemented a Java based desktop application and web service. The desktop



**Fig 5. Core functionalities—desktop application and web service.**

doi:10.1371/journal.pone.0161440.g005

application performs keyword extraction, indexing and search query augmentation, and post-processing of the result, whereas the web service is solely responsible for the oblivious evaluation of the encrypted search queries. Fig 5 illustrates the core functionalities of desktop application (data owner and authorized users) and web service.

In desktop application for data owner we generates an inverted index from the plain text, i.e., the data that needs to be outsourced to cloud storage. For this, we employ Apache Lucene API [28], which is a fully-featured text search engine that is focused on high performance. Apache Lucene enables us to extract all keywords, avoiding indexing of the stop word and repeated keywords. Once the keywords are extracted from the plain text in the form of the

inverted index, we augment them with semantic information, i.e., synonyms and root words, using WordNet [29]. To use the augmented keywords to search the encrypted data, a hash of the individual keywords is computed using the SHA-512 hashing algorithm. The hashed keywords are then encoded into BigInteger values of arbitrary size. Once encoded, the inverted index entries are encrypted with the symmetric encryption algorithm and are outsourced to the web service.

User desktop application authorized users model their search query in the form of a polynomial and learn the semantic map between their search criteria and the outsourced encrypted index. The search criteria defined by a user is augmented and encoded in a similar way, as discussed for the inverted index. To evaluate the encrypted search query, we utilize the Pascal Paillier cryptosystem [30]. The secret key of Pascal Paillier is used to conceal the search criteria, whereas a public key is used by the cloud server to evaluate the search query. For each encrypted keyword in the encrypted index, the search is evaluated and the result is transmitted back to the user.

### Evaluation

The proposed methodology for the encrypted data search was evaluated on a 2.60 GHz Windows 7 PC with 2.0 GB of main memory. We opted for a relatively low-end machine to demonstrate that the proposed methodology can be realized for any public cloud-based storage service, since it does not have any special computational requirements.

In the subsequent section, we first present the computational complexity of the semantic search for encrypted data. We then discuss the computational analysis of augmenting the inverted index and the search criteria with the semantic information. In the last section, we show the computational load of the oblivious query evaluation, where the search query is composed of multiple search criteria.

### Complexity Analysis

The computational complexity and the amount of data transmitted between the entities are analyzed in order to illustrate the efficacy of our proposed encrypted data search. Both of these parameters are directly proportional to the size of the encrypted index outsourced to the cloud storage and the size of the encrypted search query. Table 2 shows the set of operations performed in each step of our proposed methodology, along with the input size and the amount of transmitted data. For the sake of simplicity, we regarded the cryptographic and hashing operations as constant time operations, because the proposed methodology is not confined to any particular encryption or hashing algorithm.

**Indexing:** To extract keywords and to identify the semantically equivalent words for each extracted keyword, we utilize freely available libraries, such as Apache Lucene and Wordnet. Since these libraries complement our proposed system, we consider their execution at a

**Table 2. Complexity analysis of semantic search for encrypted data.**

Steps	Operations	Input Size	Computational Complexity	Transmitted Values
Indexing	Public encoding & Symmetric encryption	$N$	$O(N)$	–
Data outsourcing	–	$N$	$O(N)$	$N$
Query formulation	Asymmetric encryption & Polynomial modeling	$n$	$O(n^3)$	$n + 2$
Query execution	Polynomial evaluation	$(n + 1)N$	$O(n^2.N)$	$N$
Post-processing of results	Asymmetric decryption	$N$	depends on $n$	depends on $n$

doi:10.1371/journal.pone.0161440.t002

constant time. Thus, the computational complexity of the index is  $O(N)$ , where  $N$  is the size of the augmented index containing both the synonym and the root word.

**Data outsourcing:** We regard the computational complexity of the data outsourcing to be  $O(N)$ , where  $N$  is the size of the augmented index. In total,  $N$  values are transmitted to the cloud server.

**Query formulation:** The query formation is comprised of two steps. In the first step, the user defines the search criteria, which is then expanded with semantic information and finally encoded into a fixed length integer value using a hashing algorithm. In the second step, encoded values are used to model a polynomial, which is then concealed using the Pascal Pailier homomorphic encryption algorithm. Since the retrieval of the synonyms and root word and the encoding of the expanded search criteria are regarded as constant time operations, the computational complexity of the first step is  $O(n)$ , where  $n$  is the size of the expanded search criteria. For the second step, the first individual encoded search criterion is modeled as a polynomial (where the search criterion is a root of the polynomial), the individual polynomials are multiplied together, and the coefficients of the resultant polynomial are concealed with the private key of the homomorphic encryption algorithm. Since the encrypted search query is modeled in three steps, its computational complexity is  $O(n^3)$ , where  $n$  is the size of the encoded search criteria. In total,  $n + 2$  values are transmitted to the cloud server, where  $n + 1$  is the number of coefficients, and there is one public key of the homomorphic encryption algorithm.

**Query execution:** The encrypted search query is evaluated for each encrypted entry in the index outsourced to a cloud server. The computation complexity of the query execution depends on two factors: the size of the index,  $N$ , and the size of the polynomial that models the search query,  $n + 1$ . Thus, the computational complexity of an oblivious search query evaluation in terms of the Big- $O$  notation can be expressed as  $O(n^2 N)$ . The size of search query results is also directly proportional to the size of the index. In total,  $N$  values are transmitted to the user as a result of the search query execution.

**Post-processing of results:** Post-processing of the results is relatively a simple process, and it only deciphers the number of oblivious values,  $N$ , sent by the cloud server. Since we consider the computational load of the cryptographic operations as a constant, the computational complexity of the post-processing of the result can be regarded as  $O(N)$ . The computational time required to post-process an individual oblivious record depends on size of search query i.e., number of keywords used to model conjunctive search query.

## Computational Analysis

The computational time required to enrich the inverted index and the search query with semantic information is presented along with the amount of time required to model an encrypted search query. We studied the computational time of conjunctive search queries and presented the time required to evaluate those encrypted search queries over the enriched inverted index. [Table 3](#) shows the average computational time of the aforementioned steps computed over 100 iterations. To measure computational time we used Java time logging mechanism. System time to the precision of nanoseconds was logged at the beginning and end of the process, difference between logged timestamps was regarded as the time required to completely execute the process.

**Synonym identification:** The index and search query expansion are two important steps which enable a semantic search over encrypted data. For the computational analysis, we evaluated Wordnet API over a batch of 50 words. These words can be regarded as keyword entries in the inverted index and search criteria defined by authorized subscribers. For a batch, the total number of synonyms and the execution time are noted first, and we extracted 872

**Table 3. Computational time analysis of semantic search for encrypted data.**

Query Size (No. of keywords)	Query formulation (ms)	Query execution (ms)
2	238	245
4	411	791
6	590	1169
8	778	2811
10	982	4230
12	1187	6018
14	1405	8796

doi:10.1371/journal.pone.0161440.t003

synonyms in 408 ms. This total number of synonyms is then divided by 50 to calculate the average number of synonyms per word, which is approximately 18 synonyms per word. Finally, the time required to extract these average synonyms per word is calculated as the total execution time divided by the total time multiplied by the average number of synonyms per word, i.e.,  $(408/872) * 18 = 8.42$  ms, which represents the average execution time per word. This exercise is repeated over 10 batches of different word sets for a more realistic time calculation. For the evaluation, we selected the standard implementation of WordNet and did not consider an optimization strategy.

**Query formulation:** The query formulation is comprised of the polynomial modeling and the asymmetric encryption of polynomial coefficients. As discussed in the complexity analysis, the computational cost of the query formulation depends upon the size of the search criteria that constitutes the encrypted search query. Unlike the conventional methodology, the proposed method supports a conjunctive search query, allowing authorized subscribers to set multiple search filters instead of relying on a single search criterion. Table 3 shows the average computational cost to model an encrypted search query with multiple search criteria.

**Query execution:** For each index entry (encrypted keyword), the cloud server evaluates the search query. The query evaluation is merely a process of polynomial evaluation at a certain value, and that value happens to be an individual keyword in the outsourced enhanced index. The computational time of the query execution depends on the size of the enhanced index and the encrypted search query. The entire process of query execution utilizes the homomorphic property of the Pascal Paillier cryptosystem. The result of the query execution is oblivious to the cloud server. The computational time of the encrypted search query comprises a range of two to ten search criteria, as shown in Table 3, which shows how the increase in the number of search criteria affects the computational time required to obviously execute a search query.

## Security Analysis

In this section, we present the security analysis of our proposed methodology. Particularly, we focus on the capabilities of malicious entities to learn the encrypted search query and to deduce confidential information about the encrypted outsourced data. We examine the advantage of an untrusted cloud service provider to learn the result of the search query evaluation and to deduce information that can lead to a potential loss of privacy. We then discuss the scenario in which an unauthorized subscriber attempts to search encrypted data to which it does not have access.

The proposed methodology utilizes a number of cryptographic primitives to ensure execution of the encrypted search queries and to restrain malicious entities from deducing information that assists them in compromising the privacy of the outsourced data. As illustrated in the descriptive details of our proposed methodology, the inverted index is encrypted with

symmetric encryption. To ensure oblivious evaluation of the search queries, homomorphic encryption is utilized along with a private matching protocol [31]. For the security analysis of these cryptographic primitives, readers can refer to [30] and [32]. In the subsequent sections, we examine the capabilities of malicious entities to deduce confidential information within the context of a semantic search over encrypted data.

### Malicious Cloud Server

The proposed methodology for encrypted data search utilizes the computational power and storage facility of a cloud server to execute search queries, instead of relying on a trusted third party. The cloud server uses an encrypted index  $\hat{\mathcal{I}}_s^k$ , that is comprised of encrypted keywords. To compromise the privacy of the outsourced data, the cloud server either has to decipher the inverted index or deduce information from the evaluation of the encrypted search queries. In addition, the search queries are submitted in an encrypted format ( $\alpha_{0...l}^{\sigma_{sk}}$ ), and are evaluated by using a private matching protocol i.e., ( $P(y_{0...n} \in \hat{\mathcal{I}}_{kw_{0...n}}^{\omega_u}) = \Delta_{y_{0...n}}$ ). Since, search queries are encrypted and the result of query evaluation is oblivious to the cloud server, the cloud server cannot learn any information about the keywords concealed in the search query.

In order to compromise the privacy of the outsourced data, the cloud server needs access to the secret key,  $k$ , which is shared by the repository owner. Once the cloud server has access to the secret key it can effortlessly decipher the keywords that comprises the inverted index. However, only authorized subscribers have access to the secret key as it is encrypted with their respective public key ( $\omega_{u_i}^{k_{pub}}$ ). Thus, for a cloud server the computational complexity to compromise the privacy of the outsourced is equivalent to that of asymmetric encryption. However, even if the cloud server manages to gain access to the secret key it can only decipher the encrypted keywords that are associated with the outsourced data—so that confidentiality of the outsourced data is preserved as it is encrypted with a symmetric encryption key, which is only disseminated to authorized subscribers. Since our proposed methodology deals with the encrypted data search, the topic of authorized data access is beyond its scope.

### Malicious Subscriber

The proposed methodology of encrypted data search not only realizes encrypted search in untrusted domain it also tackles the problem of unauthorized data search by malicious users. It ensures that unauthorized subscribers are not able to deduce any information about the encrypted outsourced data by simply learning the presence or absence of keywords. It does provide protection against conspired attacks by unauthorized subscribers and untrusted cloud server. Since, encrypted index is concealed with secret key that is only shared amongst authorized collaborating subscribers, malicious subscriber can not successfully evaluate their search query.

To search the encrypted data, the search criteria ( $\mathcal{C}_{kw_{0...l}}$ ) is concealed with a secret key i.e.,  $\mathcal{E}_S(\hat{\mathcal{C}}_{kw_{0...l}}, k) = \hat{\mathcal{C}}_{kw_{0...l}}^k$ . Once concealed it is then used to the model search query, which is comprised of the encrypted coefficients  $\alpha_{0...l}^{\sigma_{sk}}$ , of polynomial  $P(x \in \hat{\mathcal{C}}_{kw_{0...l}}^k)$ .

Since, only authorized subscribers have secret keys, search queries from unauthorized subscribers cannot be evaluated successfully. Also, unauthorized subscribers cannot intercept valid search query to modify the search criteria. This is because unauthorized subscriber does not have valid secret key to model new or a part of valid search request. The concealed search criteria are only comparable with the encrypted index if the search criteria are also encrypted with the same key. Even if unauthorized subscribers collude with the cloud server, the execution of unauthorized search queries cannot assist them in learning any useful information. The search

criterion encrypted with the arbitrary secret key is not compatible with the concealed inverted index, i.e.,  $\hat{C}_{kw_0..l}^{k_2} \notin \hat{I}_{kw_0..n}^k$ . Thus, for unauthorized subscribers, it is computationally infeasible to deduce any information that can lead to the potential loss of data privacy.

## Conclusion and Future Directions

This paper addresses the problem of privacy-aware data search within the untrusted domain of a cloud service provider. It proposes an index-based privacy-aware data search methodology which can identify a semantic match between encrypted data and search criteria. Unlike the conventional methodology, the proposed privacy-aware data search leverages authorized subscribers to access relevant data by defining conjunctive search queries without relying on any trapdoors defined by the data owner. It realizes an oblivious data search, which ensures that the cloud service provider can only assist in the execution of encrypted search queries; however, the CSP can not learn or deduce confidential information from the execution of the search query, which can lead to the potential loss of data privacy. The security analysis demonstrated that, for malicious subscribers and untrusted cloud service providers, the proposed methodology always generates a randomized response that restrains them from learning of the presence or absence of a particular keyword in the outsourced encrypted data. Since the proposed methodology is an index-based data search, it does not have a requirement to encrypt the outsourced data with a particular encryption algorithm. The encryption of outsourced data with an arbitrary encryption algorithm does not affect the operation of the proposed methodology. The computational analysis shows that the proposed methodology exerts a reasonable computational load on authorized subscribers to model their encrypted search queries.

So far, the proposed methodology can only identify exact matches between the extended search criteria and the inverted index. In the future, we plan to include wildcard-enabled search queries, which can be used to match a substring while ensuring oblivious execution and privacy-awareness of search queries.

## Acknowledgments

This work was supported by a grant from the Kyung Hee University in 2013[KHU-20130439].

## Author Contributions

**Conceived and designed the experiments:** ZP AMK SL TC.

**Analyzed the data:** MA AMK.

**Wrote the paper:** ZP MA AMK.

Conceptualized the research: ZP MA. Designed and implemented the privacy-aware relevant data access: ZP MA AMK. Contributed in reviewing and revising the manuscript: ZP SL TC.

## References

1. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. vol. 25. Amsterdam, The Netherlands, The Netherlands: Elsevier Science Publishers B. V.; 2009. p. 599–616. Available from: <http://portal.acm.org/citation.cfm?id=1528937.1529211>.
2. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, et al. A view of cloud computing. *Commun ACM*. 2010 April; 53:50–58. Available from: <http://doi.acm.org/10.1145/1721654.1721672>. doi: [10.1145/1721654.1721672](https://doi.org/10.1145/1721654.1721672)

3. Buyya R, Yeo CS, Venugopal S. Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities, in: In: Department of Computer Science and Software Engineering (CSSE), The University of Melbourne, Australia. He; 2008. p. 10–1016.
4. Velte T, Velte A, Elsenpeter R. Cloud Computing, A Practical Approach. 1st ed. New York, NY, USA: McGraw-Hill, Inc.; 2010.
5. Archer J, Cullinane D, Puhlmann N, Boehme A, Kurtz P, Reavis J. Security Guidance for Critical Areas of Focus in Cloud Computing V3.0; 2011. <https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf>. Available from: <http://dl.acm.org/citation.cfm?id=1833515.1833621>.
6. Ristenpart T, Tromer E, Shacham H, Savage S. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In: Proceedings of the 16th ACM conference on Computer and communications security. CCS'09. New York, NY, USA: ACM; 2009. p. 199–212. Available from: <http://doi.acm.org/10.1145/1653662.1653687>.
7. Kaufman LM. Data Security in the World of Cloud Computing. IEEE Security and Privacy. 2009 July; 7:61–64. Available from: <http://dl.acm.org/citation.cfm?id=1591890.1592198>. doi: [10.1109/MSP.2009.87](https://doi.org/10.1109/MSP.2009.87)
8. Hacigümüş H, Iyer B, Li C, Mehrotra S. Executing SQL over encrypted data in the database-service-provider model. In: Proceedings of the 2002 ACM SIGMOD international conference on Management of data. SIGMOD'02. New York, NY, USA: ACM; 2002. p. 216–227. Available from: <http://doi.acm.org/10.1145/564691.564717>.
9. Zhou M, Mu Y, Susilo W, Yan J, Dong L. Privacy enhanced data outsourcing in the cloud. Journal of Network and Computer Applications. 2012; 35(4):1367–1373. Available from: <http://www.sciencedirect.com/science/article/pii/S1084804512000367>. doi: [10.1016/j.jnca.2012.01.022](https://doi.org/10.1016/j.jnca.2012.01.022)
10. Song DX, Wagner D, Perrig A. Practical techniques for searches on encrypted data. In: Security and Privacy, 2000. S P 2000. Proceedings. 2000 IEEE Symposium on; 2000. p. 44–55.
11. Boneh D, Crescenzo GD, Ostrovsky R, Persiano G. Public Key Encryption with Keyword Search. In: EUROCRYPT; 2004. p. 506–522.
12. Li M, Yu S, Cao N, Lou W. Authorized Private Keyword Search over Encrypted Data in Cloud Computing. In: Distributed Computing Systems (ICDCS), 2011 31st International Conference on; 2011. p. 383–392.
13. cheng Chang Y, Mitzenmacher M. Privacy Preserving Keyword Searches on Remote Encrypted Data. In: In Proc. of 3rd Applied Cryptography and Network Security Conference (ACNS); 2005. p. 442–455.
14. Curtmola R, Garay J, Kamara S, Ostrovsky R. Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions; 2006.
15. Yang Z, Zhong S, Wright RN. Privacy-Preserving Queries on Encrypted Data. In: In Proc. of 11th European Symposium On Research In Computer Security (Esorics); 2006. p. 479–495.
16. Okamoto T, Takashima K. Hierarchical Predicate Encryption for Inner-Products. In: Matsui M, editor. Advances in Cryptology– ASIACRYPT 2009. vol. 5912 of Lecture Notes in Computer Science. Springer Berlin Heidelberg; 2009. p. 214–231. Available from: [http://dx.doi.org/10.1007/978-3-642-10366-7\\_13](http://dx.doi.org/10.1007/978-3-642-10366-7_13).
17. Wang C, Cao N, Li J, Ren K, Lou W. Secure Ranked Keyword Search over Encrypted Cloud Data. In: Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on; 2010. p. 253–262.
18. Kamara S, Papamanthou C, Roeder T. CS2: A Searchable Cryptographic Cloud Storage System. Microsoft Research; 2011. MSR-TR-2011-58. Available from: <http://research.microsoft.com/apps/pubs/default.aspx?id=148632>.
19. Sun W, Wang B, Cao N, Li M, Lou W, Hou YT, et al. Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking. In: Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security. ASIA CCS'13. New York, NY, USA: ACM; 2013. p. 71–82. Available from: <http://doi.acm.org/10.1145/2484313.2484322>.
20. Pervez Z, Awan A, Khattak A, Lee S, Huh EN. Privacy-aware searching with oblivious term matching for cloud storage. vol. 63. Springer US; 2013. p. 538–560. Available from: <http://dx.doi.org/10.1007/s11227-012-0829-z>.
21. Google Search Appliance.; 2013. Available from: <http://www.google.co.uk/enterprise/search/gsa.html>.
22. Enterprise Search Server Solutions.; 2013. <http://sharepoint.microsoft.com/en-us/product/capabilities/search/Pages/Search-Server.aspx>.
23. Singh A, Srivatsa M, Liu L. Search-as-a-service: Outsourced search over outsourced storage. ACM Trans Web. 2009 September; 3:13:1–13:33.
24. Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM conference on Computer and communications

- security. CCS'06. New York, NY, USA: ACM; 2006. p. 89–98. Available from: <http://doi.acm.org/10.1145/1180405.1180418>.
25. Kamara S, Lauter K. Cryptographic Cloud Storage. In: Financial Cryptography and Data Security. vol. 6054 of Lecture Notes in Computer Science; 2010. p. 136–149.
  26. Wang C, Wang Q, Ren K, Lou W. Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing. In: INFOCOM, 2010 Proceedings IEEE; 2010. p. 1–9.
  27. Pervez Z, Khattak AM, Lee S, Lee YK. SAPDS: self-healing attribute-based privacy aware data sharing in cloud. *The Journal of Supercomputing*. 2012; 62(1):431–460. doi: [10.1007/s11227-011-0727-9](https://doi.org/10.1007/s11227-011-0727-9)
  28. Lucene. Apache Lucene Core; 2013. <http://lucene.apache.org/core/>.
  29. WordNet. About WordNet—a large lexical database of English; 2013. <http://wordnet.princeton.edu/wordnet/>.
  30. Paillier P. Public key cryptosystems based on composite degree residuosity classes. In: Proceedings of the 17th international conference on Theory and application of cryptographic techniques. EURO-CRYPT'99. Berlin, Heidelberg: Springer-Verlag; 1999. p. 223–238.
  31. Freedman M, Nissim K, Pinkas B. Efficient Private Matching and Set Intersection. Springer-Verlag; 2004. p. 1–19.
  32. Goldreich O, Israel R, Dana T. Foundations of Cryptography; 1995.