Hindawi Publishing Corporation
Scientific Programming
Volume 2016, Article ID 8057208, 9 pages
http://dx.doi.org/10.1155/2016/8057208

Research Article

Revisiting the Practicality of Search on Encrypted Data: From the Security Broker's Perspective

Peiyi Han,[1,2] Chuanyi Liu,[1] Binxing Fang,[1,3] Guofeng Wang,[1,2] Xiaobao Song,[4] and Lei Wan[4]

[1]*Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China*
[2]*School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[3]*Institute of Electronic and Information Engineering in Dongguan, University of Electronic Science and Technology of China, Dongguan 523000, China*
[4]*Shenzhen Yunanbao Technology Co., Ltd., Shenzhen 518057, China*

Correspondence should be addressed to Chuanyi Liu; cy-liu04@mails.tsinghua.edu.cn

Received 28 February 2016; Accepted 14 September 2016

Academic Editor: Ligang He

The primary business challenge for the customers to use outsourced computation and storage is the loss of data control and security. So encryption will become a commodity in the near future. There is big diffusion with the above scenario: take advantage of current application's full functionalities at the same time ensuring their sensitive data remains protected and under customers' control. Prior works have achieved effective progress towards satisfying both sides. But there are still some technical challenges, such as supporting file or data-stream based applications and supporting full-text and advanced searches. In this paper, a novel security broker based encrypted data search scheme, called Enc-YUN, is proposed, which transparently builds a reverse index at the security broker when the data flow is transmitted to the cloud. And search firstly takes place on the index, in which the mapping structure corresponds to and retrieves the very encrypted data in the cloud on behalf of the client. With this scheme, updated-to-date full-text search techniques can be easily integrated to carry out the most advanced search functionalities, at the same time, maintaining the strongest levels of data protection from curious providers or third parties. Experimental results show that Enc-YUN is effective with broad categories of cloud applications, and the performance overhead induced is minor and acceptable according to user's perceptual experience.

1. Introduction

Specialization and outsourcing make society more efficient and scalable, and computing is not any different. According to Cisco global mobile data traffic report [1], cloud applications account for 83% of total mobile data traffic by 2015 and will account for 90% by 2019. The primary business challenge for the customers to use outsourced computation and storage is the loss of data control and security, especially with mission-critical applications or privacy-sensitive applications.

A promising solution is encryption, providing only encrypted data to the cloud. However, there exists diffusion: take advantage of current application's full functionalities at the same time ensuring their sensitive data remains protected and under customers' control. Take data search as an example; can customers still search the contents based on encrypted data?

Against the above problem, current efforts can be summarized into two categories: the first approach focuses on the Searchable Encryption (SE) Algorithms [2], which allow the data owner to delegate search capabilities to the cloud provider without decrypting the documents. However, this approach should modify the legacy cloud provider's interface to adopt the very SE library. And the search capabilities are limited to keyword granularity. The second approach often uses a proxy, namely, data security broker, which transparently sits between the cloud application and its users, intercepting critical data before it is passed into the cloud and replacing it with a random token or encryption value that is meaningless for the cloud.

This paper proposes security broker based architecture to protect the data privacy and search on encrypted data, called Enc-YUN, which builds a reverse index [3] to map the data transmitted and data in the cloud, and search firstly takes place on the index and then retrieves the corresponding encrypted data in the cloud on behalf of the client. With this scheme, updated-to-date full-text search techniques can be easily integrated to carry out the most advanced search functionalities, at the same time, maintaining the strongest levels of data protection from curious providers or third parties.

There is no free lunch, though. With the above scheme, some technical challenges are still waiting to be solved, which can be summarized as follows.

No Modifications on the Legacy Applications. Actually, rewriting or modifying applications in order to implement the Searchable Data Encryption Algorithms is always impossible for the application providers. Broker is located between the application and the user, which can intercept the information to the plaintext of the user before it is transferred to the applications and convert the ciphertext into plaintext before the information is sent to the user. The user should trust broker which is deployed on the user's premises. The broker and sensitive data are in the control of user, even though broker works by intercepting user data to cloud service. In fact, the broker provides an ability to adapt various cloud services transparently without modifying or even sacrificing usability.

Supporting More Categories of Applications While Being Non-Custom-Crafted. With Enc-YUN, it is necessary to recognize the protocol of the application to determine whether the application requires encrypted and semantic analysis on the content of protocol for obtaining the positions of sensitive data which need to be encrypted. However, lots of various SaaS applications need to be analyzed which is a big challenge! Enc-YUN should support popular SaaS applications as much as possible but should not analyze protocol of applications one by one. Then, we classified SaaS applications and had a protocol analysis about typical application in every category. If a new application needs to be protected, Enc-YUN would find the corresponding category and only require changing fewer codes. In the future, we will investigate mechanisms that fully automate analysis about protocol of application and semantic content of protocol.

Selective and Searchable Data Encryption. First, if we would encrypt simply all the content of protocol, this leads to the server-side of application parse data error and possibly denial of service. And the format of SaaS applications data such as phone number and email needs to be verified by cloud services. Thus, these data also should not be encrypted in case of impacting the functionality of SaaS applications. Second, keyword search is a common operation in SaaS applications, but it is often impractical to run on the client because it would require downloading large amounts of data to the user's machine. While there exist practical cryptographic schemes for keyword search, they require that cloud provider modify or rewrite the application code and interface.

The challenge facing Enc-YUN is how to implement selective and searchable data encryption. Enc-YUN could determine what data should be encrypted by policies related to the attribute of data. That greatly preserves the usability and user experience of cloud services.

2. Related Work

In this section, previous work has been well systematized and summarized into two parts. The first part is mainly about Searchable Data Encryption Algorithms by which user can search documents without decrypting them. Client controlled search on encrypted data is lucidly elaborated in the second part of this section. This approach is often implemented by a proxy called data security broker to intercept critical data before it is passed into the cloud and replace it with a random token or encryption value.

2.1. Searchable Data Encryption Algorithms. Traditional ciphertext search technologies can be summarized into two typical categories.

Linear search compares the words of ciphertext in turn to confirm whether or not the keyword exists and count the frequency of this keyword. For instance, Song et al. [4] proposed a solution based on searchable symmetric-key encryption (SSKE) scheme which adopts stream cipher method to encrypt character data.

Security index-based ciphertext search establishes a keyword index [5] according to the document and then encrypts and uploads index and document to the cloud. And keywords will be compared from the index instead of the whole document. Based on this approach, Boneh et al. [6] provided a method, namely, public-key encryption with keyword search (PEKS) [7, 8], such that the recipient searches keywords from the file sent by the sender. And Agrawal et al. [9] in IBM Research Center proposed an order preserving encryption scheme (OPES) algorithm [10] that keeps the values in order during the encryption process.

However, the conventional ciphertext search methods proposed above are based on the premise that the cloud provider needs to change the interface of existing cloud services. In fact, it is difficult for providers to do this in practice.

2.2. Client Controlled Search on Encrypted Data. The broker is the intermediary between cloud providers and users that encrypt and protect sensitive data of users. Consider several deployment locations of broker based on a simplified architecture of typical SaaS applications, outlined in Figure 1.

2.2.1. Between the Application's Client-Side and User. The broker can encrypt data at the point (Figure 1(a)) before the application code (including the client-side code) can access it. The application can only view an encrypted version of the data.

He et al. proposed a general solution which is ShadowCrypt [11] for encrypting textual data for existing web applications. ShadowCrypt runs as a browser extension, replacing

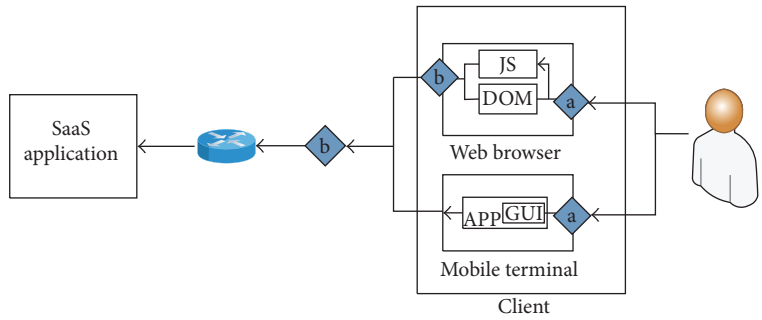

Figure 1: Architecture of typical SaaS applications and chokepoints for data encryption.

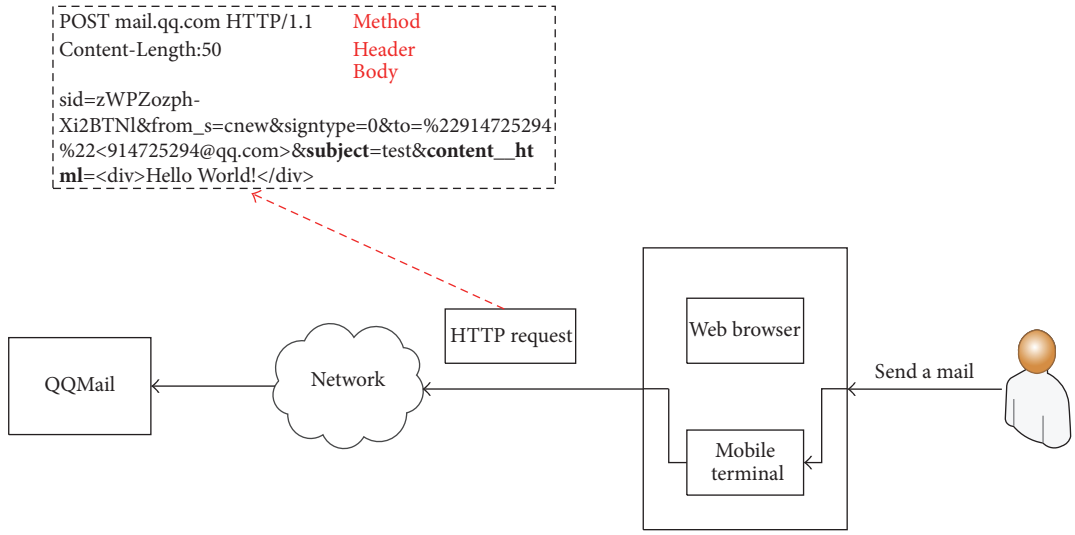

Figure 2: The procedure of sending mails to QQMail.

input elements in a page with secure, isolated shadow inputs and encrypted text with secure, isolated clear text. According to Figure 2, ShadowCrypt cannot encrypt data in mobile application. And Figure 3 shows file data which is uploaded to Dropbox by users is not stored in DOM nodes. Thus, ShadowCrypt could not support encrypting file data. In addition, it only applies to several web browsers in PC platforms such as Chrome and Firefox.

Mimesis Aegis [12] which is suitable for mobile platforms not only provides isolation but also preserves the user experience through the creation of a conceptual layer called Layer 7.5 (L-7.5), which is interposed between the application (OSI Layer 7) and the user (Layer 8). However, Mimesis could not support encrypting file data.

2.2.2. Between the Application's Client-Side and Network. The broker also can be deployed at the point (Figure 1(b)) to encrypt data after the application's client-side (i.e., JavaScript/HTML) would send data to the cloud services. Therefore, we can adopt the extension of browser and proxy as the broker.

Virtru [13] offers a browser plugin that performs email encryption, such that web-mail providers like Gmail cannot see users' data in the clear. But Virtru provides only a point solution for a handful of web-mail providers and does not generalize to other web applications and mobile applications.

Mylar [14] is an extension of the Meteor JavaScript framework for building applications that encrypt all their data sent to the server. Developers need to write their applications in Meteor (affecting backwards compatibility) and tell Mylar what data needs encryption.

3. Enc-YUN: Security Broker Based Search on Encrypted Data

There are three different parties in Enc-YUN: the users, the security broker, and the cloud services. Enc-YUN aims to protect the users' confidential data from attacking by hackers or intercepting by cloud providers.

3.1. Architecture. The architecture of Enc-YUN is shown in Figure 5. Enc-YUN consists of the five following components.

Parser. It intercepts data sent to and from the server, and it is responsible for recognizing application protocol and analyzing semantic content of protocol between the user and the

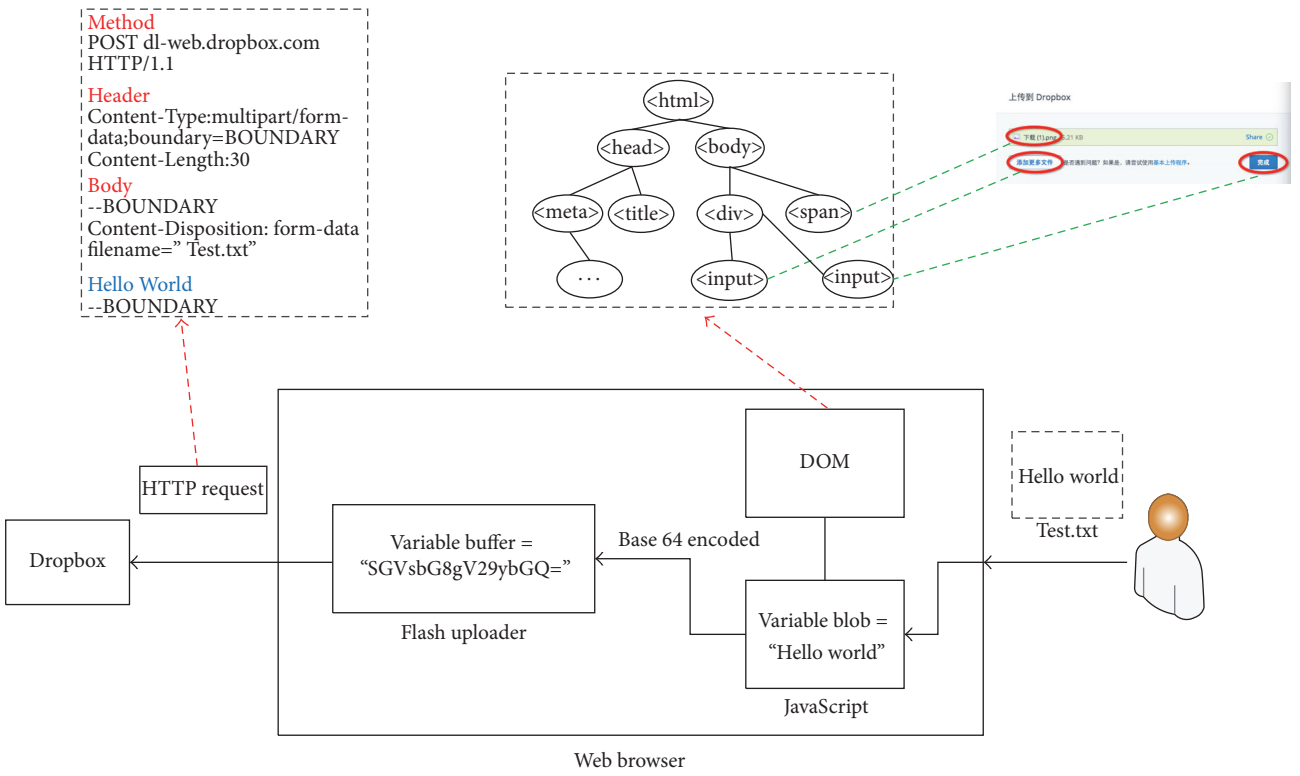

Figure 3: The procedure of uploading files to Dropbox.

cloud service in order to obtain what data require to be encrypted or decrypted. It supports multiple application protocols and various data formats. According to Figure 4, the broker could also intercept a secure channel such as SSL/TLS between users and cloud servers. The client sets SSL connection with the broker, and the broker establishes another SSL connection with the cloud server. As the broker works on behalf of clients, it is trusted by clients even when it decrypts the original data sent by clients.

Encrypter. It supports selective and searchable data encryption and integrates with several encryption algorithms such as AES and DES. Encrypter calls for secret keys from Key Manager before encrypting or decrypting and protects user sensitive data with distributed keys.

Searcher. It would transform metadata to MetaData Manager for ciphertext search. Searcher receives search keywords from users and searches content from cache which stores confidential data and return the results to users.

Transmitter. It is responsible for transmitting encrypted data from users to cloud services and data from SaaS applications to users.

Key Manager. It performs generation, storage, and management on keys used to encrypt or decrypt.

MetaData Manager. It stores and manages metadata.

3.2. Application Analysis. Enc-YUN not only supports popular SaaS applications as much as possible but also does not analyze protocol of applications one by one. In regard of the protocol recognition and semantic analysis when applying SaaS, we encountered two challenges:

(i) How to achieve the automatic adaptation of new application protocol if there are various protocols?

(ii) How to achieve the match of protocols without changing broker when the protocol changes?

The application of SaaS can be divided into the following categories: email, cloud storage, CRM, ERP, office 365, social category and so on. Analyzing the protocol of two typical applications of each category, we can find that the basic protocol is the same.

Figure 2 presents the content of the protocol about sending mail. Enc-YUN could obtain attributes and content of data by analyzing this typical format of key-value. For example, "subject" means mail subject and "content_html" means mail content. Then Enc-YUN encrypts the content of the subject and body of mail. Furthermore, Figure 3 shows the cloud storage applications adopt uploading content of files in multipart format. And parser parses the content with finite-state machine, encrypts data, assembles it into a new standard multipart format, and then forwards the request to the application. So we achieved the protocol adaptation and semantic identification of typical applications according to its category. If adding the new application is necessary,

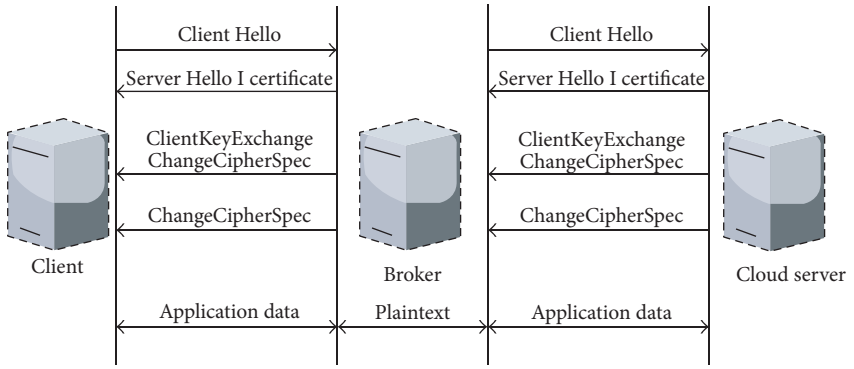

Figure 4: SSL/TLS interception in Enc-YUN.

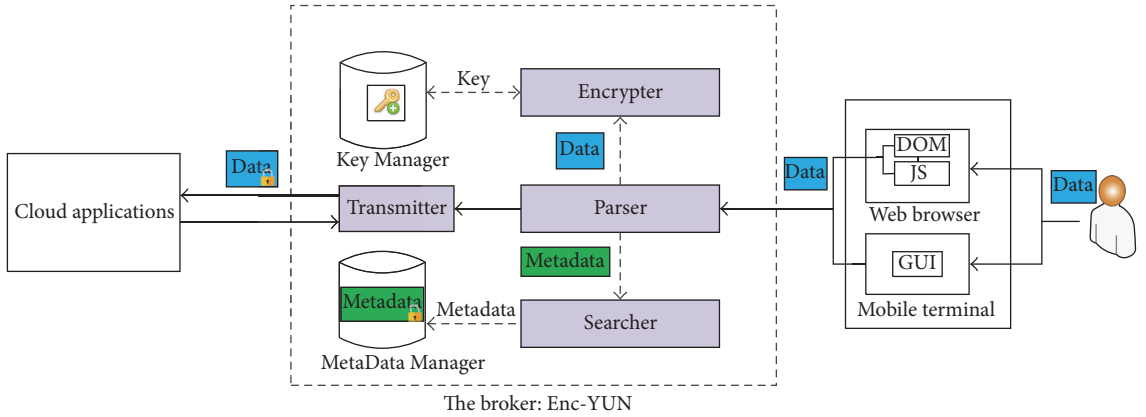

Figure 5: Architecture of the Enc-YUN.

we only need to find the category the application belongs to and achieve the protection of sensitive data in the new applications with little code modification.

Meanwhile, we will maintain an updated application feature library. Table 1 shows several features of QQMail in the library. If the protocol of the application changed, the application feature library will update the protocol feature and rules of semantic analysis of the application. And Enc-YUN could obtain new positions of sensitive data, extract data, and encrypt data by synchronizing the feature library.

3.3. Selective and Searchable Data Encryption. Cloud services would return error message when receiving request data which is encrypted or has incorrect format from users. And traditional keyword search is invalid on encrypted data. Selective and searchable data encryption [15] is another challenge for Enc-YUN.

Users have the right to choose if they want the data to be encrypted and can set some policies to control what data should be encrypted and what data should keep clear. Enc-YUN carries out selective encryption based on policies associated with attributes of confidential data.

Broker is deployed in the internal network of enterprise or organization controlled by users. Getting control of encryption keys to sensitive data, permission of selective data encryption updating policy, and metadata access, the broker is credible for the users. Under Enc-YUN, users get the control of sensitive data stored in SaaS application.

Then the paper proposed an approach of ciphertext search based on broker. The architecture of ciphertext search is shown in Figure 6:

(1) The user makes a keyword search request to the cloud application.

(2) Broker would intercept this request and receive the keyword from user. Then, the search query is executed against the local index.

(3) The local index, which stores a reverse index to map the keywords and data in the cloud, returns all of the associated metadata to the broker.

(4) The broker forwards the result to the user.

(5) The user retrieves the encrypted data or records according to the metadata ID which is the identity of encrypted data in cloud applications.

(6) Cloud applications return encrypted data which contains the keyword to the broker.

(7) Broker intercepts and decrypts ciphertext and then returns plaintext to the user.

Table 1: Example of QQMail features in the library.

Application	Function	Request method	Request URI	Request content type	Encryption field	Encryption algorithm
QQMail	sendMail	POST	set3.mail.qq.com/cgi-bin/compose_send?sid=JIq-6XB24WTmf0ke	Key-value format	Subject content	AES-256
QQMail	receiveMail	GET	set3.mail.qq.com/cgi-bin/readmail?folderid=1&folderkey=1&t=readmail&mailid=ZC0010-3nAORN1KF5ITb1kOXm	HTML format	Response data	AES-256

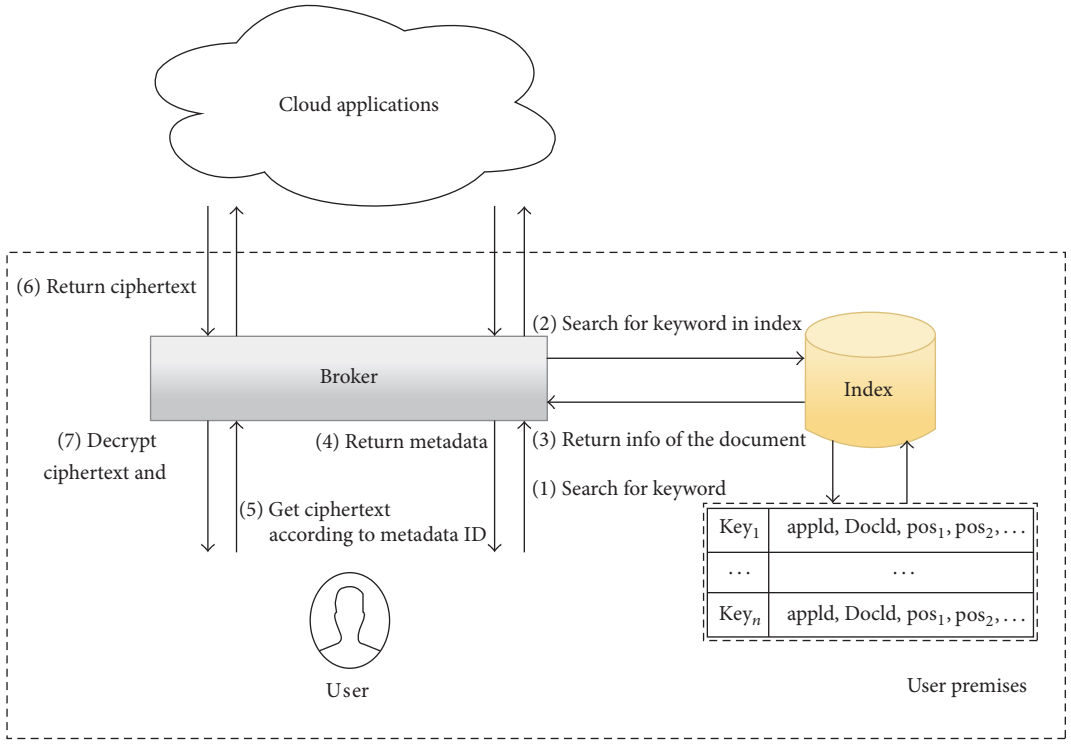

Figure 6: Architecture of ciphertext search in Enc-YUN.

3.4. Key and Metadata Management. Encrypted data sharing between users becomes difficult because the users do not share their own private key. Enc-YUN resolves this challenge by wrapped key. Each user has a private/public-key pair. The Key Manager stores the private key of the user, encrypted with the user's password. When the encrypter encrypts sensitive data, the Key Manager generates a random file-key which is used to encrypt the data. And Enc-YUN creates a wrapped key: an encryption of file-keys under the public key of users. If Alice wants to share a sensitive document with Bob, the Enc-YUN which needs to be authorized to obtain the private key of Alice unlocks the wrapped key and creates another wrapped key with Bob's public key. Thus, Bob will get plaintext of the document shared by Alice by unlocking the wrapped key with his private key.

The metadata linking the application functions and encrypted data is critical when the user in Enc-YUN authorizes another user outside of Enc-YUN to view encrypted mails and files. The metadata of a mail mainly includes components like mail id, sender id, recipient id, attachment id, ciphertext id, and so on.

4. Evaluation

In this section, we discuss the performance overhead of Enc-YUN. We conducted the test on an Intel 2.5 GHz × 2 with 2 GB of RAM. And we made some comparison tests between cloud storage and mail application under the circumstances of having Enc-YUN and not having Enc-YUN. The test could estimate the performance of the Enc-YUN by time-consuming brought from Enc-YUN.

Figure 7 shows that sending and receiving mails in Enc-YUN have more network overhead than in the normal network. Because the broker which is proxy would cost time to establish connections between users and cloud services, the time in which Enc-YUN recognizes protocols and analyzes semantic data from users had a small proportion in the entire

Table 2: Time cost in adding and viewing info in http://www.youshang.com/. "In" is the number of inputs which require encryption on the page. "Out" is the number of outputs which require decryption on the page.

Web page	In	Cost time of encrypter and parser (μs)	Cost time of adding info (μs)	Out	Cost time of encrypter and parser (μs)	Cost time of viewing info (μs)
Customer management	5	24407.8	247292 (9.87%)	10	140146.6	32576 (430%)
Supplier management	5	25025.8	57748 (43.34%)	20	222413.8	32536 (683%)
Good management	10	17043.2	75690 (22.52%)	10	112071.2	38380 (292%)
Warehouse management	1	9545.8	26490 (36.04%)	5	56961.4	25060 (227%)

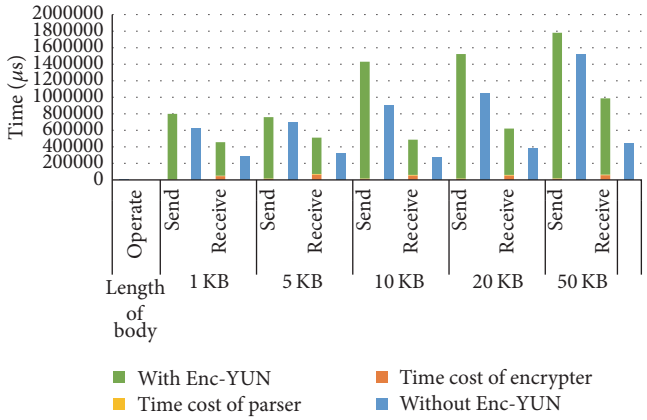

Figure 7: Time cost of Enc-YUN to send mails and receive mails in QQMail.

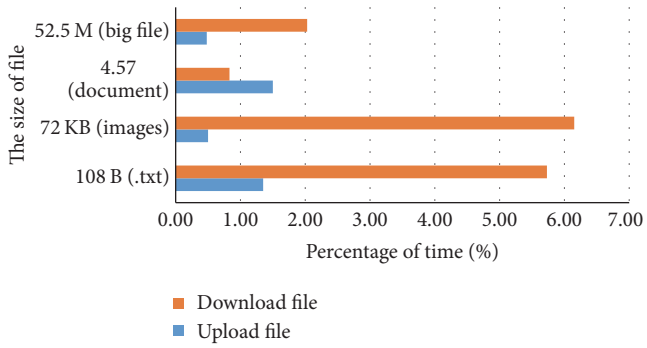

Figure 8: Percentage of time cost in encrypter of Enc-YUN to upload files and download files in Baidu Netdisk.

request completion time. This indicates that the algorithm of protocol recognition and semantic analysis in Enc-YUN is efficient. However, there are other time-consuming events besides the parser and encrypter. The time may be consumed by other modules of the Enc-YUN. We believe that the time cost will be less if every module of Enc-YUN has a good performance. And Figure 8 presents that the time percentage of uploading small files or documents is close to normal time of uploading files outside Enc-YUN. But the larger the file, the more the time it will consume. It indicates that the performance of Enc-YUN depends on the performance of the broker. Table 2 lists the number of input fields which require encryption for each page and the estimated time cost of encrypter and parser increase in microseconds. The page is loaded by browser in several hundred milliseconds. The performance overhead induced by the Enc-YUN is minor and acceptable according to user's perceptual experience in the millisecond level.

We also tested the Enc-YUN on a wide variety of popular SaaS applications such as Salesforce, Gmail, and Google Drive. And the application of message is the only type of application in which Enc-YUN cannot encrypt data because of the fact that the protocol of application is encrypted by providers. While encrypting data always impacts some application functionalities, we find that, for a broad range of applications, encrypting data still retains prominent functionality. As it can be seen from Table 3, Enc-YUN supports more semantic-rich functionalities because Enc-YUN searches in local index and retains the functionalities of normal information retrieval. All of the SSE schemes focus on text-formed data regardless of complex data structures in reality. However, Enc-YUN supports XML, JSON, relational database model, and so on. Thus the real time of search in Enc-YUN is also better than KPR [16] and KP [17] scheme.

5. Conclusion and Future Work

We presented Enc-YUN, a system for transparently encrypting confidential data and supporting selective and searchable data encryption. Without any modification of the cloud and applications, ciphertext search could take place in the Enc-YUN. Moreover, Enc-YUN has ability to support more and more applications with protocol recognition and semantic analysis.

Enc-YUN's contribution lies in providing a new perspective to achieving practical ciphertext search. And Enc-YUN's secure infrastructure and usable interface design provide a basis for implementing wide variety of encryption schemes. In the long run, we will try to improve the performance of transforming large files and we aim at supporting more automated protocol inspection and intelligent protocol analysis.

Competing Interests

The authors declare that they have no competing interests.

Table 3: Comparison between SE schemes and Enc-YUN. PEKS scheme: public-key encryption with keyword search scheme; SEKS scheme: symmetric-key encryption with keyword search scheme. m and n are the maximum # of keywords and files, k is the # of unique keywords included in an updated file (added or deleted), d is the # of incremental keywords in an updated file, p is the # of parallel processors, and t is the network latency introduced due to the interactions.

Category	Schemes	Query support						Update time
		Single keyword	Multiple keywords	Fuzzy search	Ranked search	Support complex data structures	Dynamic search	
SEKS schemes	SWP [18]	✓	✘	✘	✘	✘	✘	—
PEKS schemes	BCO+ [19]	✓	✘	✘	✘	✘	✘	—
	SLN+ [20]	✓	✘	✓	✘	✘	✘	—
Index-based SEKS schemes	Goh [21]	✓	✘	✘	✘	✘	✘	—
	GSW [22]	✓	✓	✘	✘	✘	✘	—
	KIK [23]	✓	✘	✓	✓	✘	✘	—
	XWS [24]	✓	✓	✘	✓	✘	✘	—
	KPR [16]	✓	✘	✘	✘	✘	✓	$O(k)$
	KP [17]	✓	✘	✘	✘	✘	✓	$O((m/p)\log n) + t$
Index-based PEKS schemes	RT [25]	✓	✓	✘	✘	✘	✘	—
	BW [26]	✓	✓	✘	✘	✘	✘	—
Enc-YUN	Enc-YUN	✓	✓	✓	✓	✓	✓	$O(d)$

Acknowledgments

This paper is supported by the National High Technology Research and Development Program of China (863 Program) under Grant no. 2015AA016001, the National Natural Science Foundation of China under Grant no. 61370068, Innovation Projects in Shandong Province under Grant no. 2014ZZCX03411, and Production-Study-Research Cooperation Project in Guangdong Province under Grant no. 2016B090921001.

References

[1] Cisco Visual Networking Index (VNI) Global Mobile Data Traffic Forecast for 2014 to 2019.

[2] C. Dong, G. Russello, and N. Dulay, "Shared and searchable encrypted data for untrusted servers," *Journal of Computer Security*, vol. 19, no. 3, pp. 367–397, 2011.

[3] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," *Journal of Computer Security*, vol. 19, no. 5, pp. 895–934, 2011.

[4] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 36–49, Berkeley, Calif, USA, May 2000.

[5] Y. C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in *Applied Cryptography and Network Security*, vol. 3531 of *Lecture Notes in Computer Science*, pp. 442–455, Springer, Berlin, Germany, 2005.

[6] D. Boneh, G. D. Crescenzo, and R. Ostrovsky, "Public key encryption with keyword search," in *Proceedings of the EUROCRYP '04*, pp. 506–522, Interlaken, Switzerland, May 2004.

[7] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology—EUROCRYPT 2004*, pp. 506–522, Springer, Berlin, Germany, 2004.

[8] B. Zhang and F. Zhang, "An efficient public key encryption with conjunctive-subset keywords search," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 262–267, 2011.

[9] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '04)*, pp. 563–574, Paris, France, 2004.

[10] A. Boldyreva, N. Chenette, and A. O'Neill, "Order-preserving encryption revisited: improved security analysis and alternative solutions," in *Advances in Cryptology—CRYPTO 2011*, vol. 6841 of *Lecture Notes in Computer Science*, pp. 578–595, Springer, Berlin, Germany, 2011.

[11] W. He, D. Akhawe, S. Jain, E. Shi, and D. Song, "ShadowCrypt: encrypted web applications for everyone," in *Proceedings of the 21st ACM Conference on Computer and Communications Security (CCS '14)*, pp. 1028–1039, ACM, Scottsdale, Ariz, USA, November 2014.

[12] B. Lau, S. Chung, C. Song et al., "Mimesis aegis: a mimicry privacy shield—a system's approach to data privacy on public cloud," in *Proceedings of the 23rd USENIX Security Symposium (USENIX Security '14)*, pp. 33–48, August 2014.

[13] Virtru, https://www.virtru.com/.

[14] R. A. Popa, E. Stark, S. Valdez et al., "Building web applications on top of encrypted data using Mylar," in *Proceedings of the ACM 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, pp. 85–100, Seattle, Wash, USA, April 2014.

[15] M. Abdalla, M. Bellare, D. Catalano et al., "Searchable encryption revisited: consistency properties, relation to anonymous IBE, and extensions," *Journal of Cryptology*, vol. 21, no. 3, pp. 350–391, 2008.

[16] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS '12)*, pp. 965–976, New York, NY, USA, 2012.

[17] S. Kamara and C. Papamanthou, "Parallel and dynamic searchable symmetric encryption," in *Financial Cryptography and Data Security*, vol. 7859 of *Lecture Notes in Computer Science*, pp. 258–274, Springer, Berlin, Germany, 2013.

[18] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 44–55, May 2000.

[19] D. Boneh, G. Di Crescenzo, R. Ostrovsky et al., "Public key encryption with keyword search," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 506–522, Springer, Berlin, Germany, 2004.

[20] S. Sedghi, P. van Liesdonk, S. Nikova, H. Pieter, and W. Jonker, "Searching keywords with wildcards on encrypted data," in *Proceedings of the 7th Conference on Security and Cryptography for Networks (SCN '10), Amalfi, Italy, September 2010*, vol. 6280 of *Lecture Notes in Computer Science*, pp. 138–153, Springer, 2010.

[21] E.-J. Goh, "Secure indexes," IACR Cryptology ePrint Archive, https://eprint.iacr.org/2003/216.

[22] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in *Proceedings of the 2nd International Conference on Applied Cryptography and Network Security (ACNS '04)*, pp. 31–45, Springer, Yellow Mountain, China, 2004.

[23] M. Kuzu, M. S. Islam, and M. Kantarcioglu, "Efficient similarity search over encrypted data," in *Proceedings of the IEEE 28th International Conference on Data Engineering (ICDE '12)*, pp. 1156–1167, Washington, DC, USA, April 2012.

[24] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[25] E. K. Ryu and T. Takagi, "Efficient conjunctive keyword-searchable encryption," in *Proceedings of the IEEE 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW '07)*, vol. 1, pp. 409–414, Niagara Falls, Canada, May 2007.

[26] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Proceedings of the 4th Theory of Cryptography Conference (TCC '07), Amsterdam, The Netherlands, February 2007*, vol. 4392 of *Lecture Notes in Computer Science*, pp. 535–554, Springer, 2007.