International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2, 2025

Multi tenancy cloud data with a shared privacy preserving trusted keyword search

Mr G shekar, Asthami Sai Kiran, Muthe Harish, Padala Sai Rahul

[1]Assistant Professor, Department Of IT, Guru Nanak Institutions Technical Campus (Autonomous), India.

[2,3,4]B.Tech Students, Department Of IT, Guru Nanak Institutions Technical Campus (Autonomous), India

ABSTRACT

In today's cloud computing environment, secure and efficient data sharing is paramount. This project proposes a secure cloud-based data sharing and verification system using the Verifiable yet Accountable Keyword Searchable Encryption (VAKSE) scheme. The system architecture is divided into four modules involving the Cloud Service Provider (CSP), Verifier, Data Owner, and Client, each playing a key role in secure data handling and verification. The first module focuses on the Cloud Service Provider (CSP), which manages user login, stores data owner and client details, generates cryptographic keys, and maintains file information. The CSP is also responsible for handling data requests and adding clients to the system. The second module involves the Verifier, who also logs in securely and is tasked with validating cryptographic keys. The Verifier checks whether a key is original or tampered and ensures that only valid keys are forwarded to the intended data recipients, thereby maintaining data integrity and trust. The third module introduces the Data Owner, who registers and logs in to upload files securely. Upon uploading, a key is generated to securely share data with intended clients. This ensures that the control over data remains with the owner at all times. The fourth module pertains to the Client, who registers, logs in, queries for data, and uses a generated token linked to a private key for verification and decryption. This enables clients to verify the authenticity of the data and decrypt it securely. At the core of this system lies the VAKSE scheme, which enables keyword-searchable encryption with built-in verifiability and accountability. The system utilizes four primary cryptographic algorithms: Setup, KeyGen, Encap, and Decap. The Setup algorithm generates master keys, KeyGen assigns private keys based on identities, Encap encapsulates valid keys into ciphertexts, and Decap deterministically retrieves the encapsulated key using private keys. Additionally, a mechanism is in place to detect and reject invalid ciphertexts, enhancing the robustness of the system. By integrating MAC (Message Authentication Code) encryption and VAKSE, the system ensures confidentiality, authenticity, and verifiability of cloud-stored data, offering a comprehensive solution for secure and accountable cloud-based data sharing.

1-INTRODUCTION

CLOUD computing has had a profound impact on data management. It offers massive storage and computing resources, payment-on-demand, and flexible scalability. Motivated by these advantages, thousands of clients are opting for cloud services. One typical application area is healthcare, and some applications are Healthvana [1] and CDPHP [2]; both the platforms are the tenants of Amazon [3]. Healthvana stores patient reports and CDPHP stores doctor information. It is desirable for a patient to search both the datasets to find the most suitable doctor by matching the patient data with the doctor information. For example, HIV patients store their reports in Healthvana and seek for suitable doctors from CDPHP. However, such a search across tenancies is challenging. Each tenant is an independent data owner and must abide the privacy laws, such as HIPAA [4], which are enforced to protect individuals' medical data privacy. In addition, for their own interests, companies treat patient data as an asset and tend to maintain complete control over it. Data encryption is the best practice for maintaining data privacy. Each data owner encrypts their data before outsourcing it to the cloud. This guarantees the confidentiality of the data but greatly reduces their utility. A user must download an entire dataset in order to retrieve one piece of data. Considering data utility and privacy, Song et al. [5] introduced the primitives of symmetric searchable encryption (SSE). SSE is a keyword search technique that allows search over the cipher text without decryption. Goh et al. [6] proposed a secure index to improve search efficiency. Subsequently, Curtmola et al. [7] formalized the security definition of SSE and proposed two constructions that corresponded to non adaptive semantic security and adaptive semantic security.

In early research, most works on SSE focused on the honestbut-curious cloud service provider (CSP). In such a model, the search result is fully trusted and the CSP is assumed to honestly follow the protocol specification. Search results in practice may contain corrupted data due to underlying hardware/software failures. In addition, for self-interest, the CSP may deviate from the protocol specification. For example, to reduce computational costs, CSP may randomly choose data as a search result. To mitigate this problem, Chai and Gong [8] proposed verifiable

SSE, where the search result includes not only retrieved documents but also proof of the correctness and completeness of the search. The correctness of the search means that the returned search result matches the query. completeness of the search means that the retrieved data has not been tampered with. In addition, Chen et al. [9] proposed an authenticated Merkle hash tree to verify the search result. Although significant progress has been made by the existing constructions [8], [9], the verifiable property comes at the high cost of extra storage and computation. There is still room for solutions that are more practical. Recently, with increasing demand for users (e.g., physicians), previous SSE constructions, providing a client either full access to the data or no access, expose their short term. It is desirable to design a fine-grained access control mechanism to enable data owners to selectively grant grant clients access to their data. To achieve this goal, Han et al. [10] proposed to apply attribute-based encryption [11] to solve this problem and provided a general solution in the context of public-key keyword search scenarios. With this design, only a keyword search request that matches the predefined access structure can retrieve the target document. The above searchable encryption schemes rely on public key encryption, which is inefficient compared with symmetric encryption. Moreover, none of them are suitable for use with dynamic dynamic access structures since the access structure is associated with either a key or cipher text. Any change in the access structure may result in all of the ciphertext or keys being renewed. Furthermore, all the mentioned works failed to allow a client to search the data from multiple data owners, where each data owner encrypts their data with a unique key. The existing SSE schemes only support a client to search over a single data owner [5], [12], [13]. However, in our scenario, a client needs to search for data outsourced by multiple independent data owners. For example, to identify a medical treatment for a cancer patient, a physician may need to analyze medical data from thousands of contributors (e.g., patients). An intuitive solution for this scenario is to deploy existing schemes [5], [12], [13] for each data owner, where each data owner manages their outsourced data independently. For each service request, the physician generates a specialized (authorized) request for each owner's data and sends it to the CSP. This results in a high volume of requests for a single query. Another approach is to adopt the recent Multi-Writer Encrypted Database [14], which allows multiple data owners to store data and allows clients to search across data owners.

2-LITERATURE SURVEY

TITLE: Attribute-based expressive and ranked keyword search over encrypted documents in cloud computing
AUTHOR: Q. Huang, G. Yan, and Q. Wei,
YEAR: 2023
In recent years, several new notions of security have begun receiving consideration for public-key cryptosystems, beyond the standard of security against adaptive chosen ciphertext attack (CCA2). Among these are security against randomness reset attacks, in which the randomness used in encryption is forcibly set to some previous value, and against constant secret-key leakage attacks, wherein the constant factor of a secret key's bits is leaked. In terms of formal security definitions, cast as attack games between a challenger and an adversary, a joint combination of these attacks means that the adversary has access to additional encryption queries under a randomness of his own choosing along with secret-key leakage queries. This implies that both the encryption and decryption processes of a cryptosystem are being tampered under this security notion. In this paper, we attempt to address this problem of a joint combination of randomness and secret-key leakage attacks through two cryptosystems that incorporate hash proof system and randomness extractor primitives. The first cryptosystem relies on the random oracle model and is secure against a class of adversaries, called non-reversing adversaries. We remove the random oracle oracle assumption and the non-reversing adversary requirement in our second cryptosystem, which is a standard model that relies on a proposed primitive called lossy functions. These functions allow up to *M* lossy branches in the collection to substantially lose information, allowing the cryptosystem to use this loss of information for several encryption and challenge queries. For each cryptosystem, we present detailed security proofs using the game-hopping procedure. In addition, we present a concrete instantiation of lossy functions in the end of the paper—which relies on the DDH assumption.
TITLE: Secure keyword search and data sharing mechanism for cloud computing,
AUTHOR: C. Ge, W. Susilo, Z. Liu, J. Xia, P. Szalachowski, and L. Fang,.
YEAR: 2023
The emergence of cloud infrastructure has significantly reduced the costs of hardware and software resources in computing infrastructure. To ensure security, the data is usually encrypted before it's outsourced to the cloud. Unlike searching and sharing the plain data, it is challenging to search and share the data after encryption. Nevertheless, it is a critical task for the cloud service provider as the users expect the cloud to conduct a quick search and return the result without losing data confidentiality. To overcome these problems, we propose a ciphertext-policy attribute-based mechanism with

keyword search and data sharing (CPAB-KSDS) for encrypted cloud data. The proposed solution not only supports attribute-based keyword search but also enables attribute-based data sharing at the same time, which is in contrast to the existing solutions that only support either one of two features. Additionally, the keyword in our scheme can be updated during the sharing phase without interacting with the PKG. In this article, we describe the notion of CPAB-KSDS as well as its security model. Besides, we propose a concrete scheme and prove that it is against chosen ciphertext attack and chosen keyword attack secure in the random oracle model. Finally, the proposed construction is demonstrated practical and efficient in the performance and property comparison.

TITLE: Omnes pro uno: Practical multi-writer encrypted database

AUTHOR: J. Wang and S. S. Chow,

YEAR: 2022

Multi-writer encrypted databases allow a reader to search over data contributed by multiple writers securely. Public-key searchable encryption (PKSE) appears to be the right primitive. However, its search latency is not welcomed in practice for having public-key operations linear in the entire database. In contrast, symmetric searchable encryption (SSE) realizes sublinear search, but it is inherently not multi-writer. This paper aims for the best of both SSE and PKSE, i.e., sublinear search and multiple writers, by formalizing hybrid searchable encryption (HSE), with some seemingly conflicting yet desirable features, requiring new insights to achieve. Our first contribution is a history-based security definition with new flavors of leakage concerning updates and writer corruptions, which are absent in the only known multi-writer notion of PKSE since it is vacuously secure against writers. HSE, built on top of dynamic SSE (DSSE), should satisfy the de facto standard of forward privacy. Its multi-writer support, again, makes the known approach (of secret state maintenance) fails. HSE should also feature efficient controllable search – each search can be confined to a different writer subset, while the search token size remains constant. For these, we devise a new partial rebuild technique and two new building blocks (of independent interests) – ID-coupling key-aggregate encryption and (optimal) epoch-based forward-private DSSE.Our evaluation over real-world datasets shows that HSE, surpassing prior arts by orders of magnitude, is concretely efficient for popular multi-writer database applications.

TITLE: Multi-authority fine-grained access control with accountability and its application in cloud.

AUTHOR: M. W. Zhang, W. X. Song, and J. X. Zhang

YEAR: 2022

Attribute-based encryption (ABE) is one of critical primitives for the application of fine-grained access control. To reduce the trust assumption on the attribute authority and in the meanwhile enhancing the privacy of users and the security of the encryption scheme, the notion of multi-authority ABE with an anonymous key issuing protocol has been proposed. In an ABE scheme, it allows to encrypt data for a set of users satisfying some specified attribute policy and any leakage of a decryption key cannot be associated to a user. As a result, a misbehaving user could abuse the property of access anonymity by sharing its key other unauthorized users. On the other hand, the previous work mainly focus on the key-policy ABE, which cannot support cipher text-policy access control. In this paper, we propose a privacy-aware multi-authority cipher text-policy ABE scheme with accountability, which hides the attribute information in the ciphertext and allows to trace the dishonest user identity who shares the decryption key. The efficiency analysis demonstrates that the new scheme is efficient, and the computational overhead in the tracing algorithm is only proportional to the length of the identity. Finally, we also show how to apply it in cloud computing to achieve accountable fine-grained access control system.

TITLE: "A general transformation from KP ABE to searchable encryption

AUTHOR: K. Lee

YEAR: 2021

Users are inclined to share sensitive data in a remote server if no strong security mechanism is in place. Searchable encryption satisfies the need of users to execute a search encrypted data. Previous searchable encryption methods such as "public key encryption with keyword search (PEKS)" restricted the data access to certain users, because only the assigned users were able to search the encrypted data. In this paper we will discuss the relation between Attribute Based Encryption (ABE) and searchable encryption and define a weak anonymity of the ABE scheme, named "attribute privacy". With this weak anonymity, we propose a general transformation from ABE to Attribute Based Encryption with Keyword Search (ABEKS) and a concrete attribute private key-policy ABE (KP-ABE) scheme. We present an ABEKS scheme based on this KP-ABE scheme and permit multi-users to execute a flexible search on the remote encrypted data.

TITLE: "Verifiable dynamic sym metric searchable encryption: Optimality and forward security.

AUTHOR: R. Bost, P.-A. Fouque, and D. Pointcheval

YEAR: 2023

Symmetric Searchable Encryption (SSE) is a very efficient and practical way for data owners to out-

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2, 2025

source storage of a database to a server while providing privacy guarantees. Such SSE schemes enable clients to encrypt their database while still performing queries for retrieving documents matching some keyword. This functionality is interesting to secure cloud storage, and efficient schemes have been de- signed in the past. However, security against malicious servers has been overlooked in most previous constructions and these only addressed security against honest-but-curious servers. In this paper, we study and design the first efficient SSE schemes provably secure against malicious servers. First, we give lower bounds on the complexity of such verifiable SSE schemes. Then, we construct generic solutions matching these bounds using efficient verifiable data structures. Finally, we modify an existing SSE scheme that also provides forward secrecy of search queries, and make it prov- ably secure against active adversaries, without increasing the computational complexity of the original scheme

3-METHODOLOGY

2.1 GENERAL:

In the above schemes, the client is trusted. In reality, dishonest users may attempt to access data without authorization. Even worse, some users may give away some of their original or transformed keys such that nobody can tell who has distributed these keys. The first problem is called unauthorized access. The second problem is called key abuse. The first problem can be prevented by fine-grained access control, and the second problem can be discouraged by user accountability.

2.2 METHODOLOGIES

2.2.1MODULES NAME:

1. User interface design
2. Csp
3. Verifier
4. Data owner
5.Client

REQUIREMENTS ENGINEERING

3.1 GENERAL

We have conducted experiments on our collected dataset and extensive results have demonstrated that our model outperforms all other existing models. In the future, we will investigate more tasks under this framework, such as event summarization and event attribute mining in social media.

3.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

- PROCESSOR : DUAL CORE 2 DUOS.
- RAM : 2GB DD RAM
- HARD DISK : 250 GB

3.3 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

SOFTWARE REQUIREMENTS

- FRONT END : J2EE (JSP, SERVLET)
- BACK END : MY SQL 5.5
- OPERATING SYSTEM : WINDOWS 10
- IDE : ECLIPSE

4-IMPLEMENTATION

This chapter is about the software language and the tools used in the development of the project. The platform used here is JAVA. The Primary languages are JAVA,J2EE and J2ME. In this project J2EE is chosen for implementation.

FEATURES OF JAVA

THE JAVA FRAMEWORK

Java is a programming language originally developed by James Gosling at Microsystems and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is general-purpose, concurrent, class-based, and object-oriented, and is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere".

Java is considered by many as one of the most influential programming languages of the 20th century, and is widely used from application software to web applications the java framework is a new platform independent that simplifies application development internet. Java technology's versatility, efficiency, platform portability, and security make it the ideal technology for network computing. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

OBJECTIVES OF JAVA

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2, 2025

To see places of Java in Action in our daily life, explore java.com.

WHY SOFTWARE DEVELOPERS CHOOSE JAVA

Java has been tested, refined, extended, and proven by a dedicated community. And numbering more than 6.5 million developers, it's the largest and most active on the planet. With its versatility, efficiency, and portability, Java has become invaluable to developers by enabling them to:

- Write software on one platform and run it on virtually any other platform
- Create programs to run within a Web browser and Web services
- Develop server-side applications for online forums, stores, polls, HTML forms processing, and more
- Combine applications or services using the Java language to create highly customized applications or services
- Write powerful and efficient applications for mobile phones, remote processors, low-cost consumer products, and practically any other device with a digital heartbeat

SOME WAYS SOFTWARE DEVELOPERS LEARN JAVA

Today, many colleges and universities offer courses in programming for the Java platform. In addition, developers can also enhance their Java programming skills by reading Sun's java.sun.com Web site, subscribing to Java technology-focused newsletters, using the Java Tutorial and the New to Java Programming Center, and signing up for Web, virtual, or instructor-led courses.

5-TESTING

UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures : interfacing systems or procedures must be invoked.

SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

PERFORMANCE TEST

The Performance test ensures that the output be produced within the time limits,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

6-CONCLUSION

Herein, we propose a privacy-preserving, efficient, verifiable, accountable, and parallel solution for the keyword search problem in a multitenant cloud environment. To achieve this, we devised a privacy-preserving inverted index to enable a verifiable ciphertext search. Each entry contains encrypted keyword and document identity pairs and the compressed MAC for all corresponding documents. Then, we designed a fine-grained access control mechanism through keyword-based token generation. Moreover, we embedded the user identity into the token to achieve user accountability. All those components were built into the VAKSE scheme. To further improve search efficiency, we introduced the PVAKSE, in which the inverted index was partitioned into small segments that could be searched synchronously.

REFERENCES

[1] (2022). Healthvana. [Online]. Available: https://healthvana.com
[2] (2022). CDPHP. [Online]. Available: https://www.cdphp.com
[3] (2022). Customer Success Stories. [Online]. Available: https://aws. amazon.com/solutions/case-studies/
[4] (2022). HiPAA. [Online]. Available: http://www.cms.hhs.gov/ HIPAAGenInfo
[5] D. Xiaoding Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted

data," in Proc. IEEE Symp. Secur. Privacy. (S&P), May 2000, pp. 44–55.
[6] E.-J. Goh, "Secure indexes," Cryptol. ePrint Arch., Oct. 2003.
[7] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," J. Comput. Secur., vol. 19, no. 5, pp. 895–934, 2011.
[8] Q. Chai and G. Gong, "Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers," in Proc. IEEE Int. Conf. Commun. (ICC), Jun. 2012, pp. 917–922.
[9] C. Chen et al., "An efficient privacy-preserving ranked keyword search method," IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 4, pp. 951–963, Apr. 2016.
[10] F. Han, J. Qin, H. Zhao, and J. Hu, "A general transformation from KP ABE to searchable encryption," Future Gener. Comput. Syst., vol. 30, pp. 107–115, Jan. 2014.
[11] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryp tion for fine-grained access control of encrypted data," in Proc. 13th ACM Conf. Comput. Commun. Secur., Oct. 2006, pp. 89–98.
[12] R. Brinkman, L. Feng, J. Doumen, P. H. Hartel, and W. Jonker, "Efficient tree search in encrypted data," Inf. Syst. Secur., vol. 13, no. 3, pp. 14–21, May 2004.
[13] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan, "CryptDB: Protecting confidentiality with encrypted query process ing," in Proc. 23rd ACM Symp. Oper. Syst. Principles (SOSP), 2011, pp. 85–100.
[14] J. Wang and S. S. Chow, "Omnes pro uno: Practical multi-writer encrypted database," in Proc. 31st USENIX Secur. Symp. (USENIX Security), 2022, pp. 2371–2388.
[15] J. Li, K. Ren, B. Zhu, and Z. Wan, "Privacy-aware attribute-based encryption with user accountability," in Proc. Int. Conf. Inf. Secur. Berlin, Germany: Springer, 2009, pp. 347–362. [16] J. Li, X. Chen, S. S. M. Chow, Q. Huang, D. S. Wong, and Z. Liu, "Multi-authority fine-grained access control with accountability and its application in cloud," J. Netw. Comput. Appl., vol. 112, pp. 89–96, Jun. 2018.
[17] A. Soleimanian and S. Khazaei, "Publicly verifiable searchable sym metric encryption based on efficient cryptographic components," Des., Codes Cryptogr., vol. 87, no. 1, pp. 123–147, 2019.
[18] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy preserving multi-keyword ranked search over encrypted cloud data," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 1, pp. 222–233, Jan. 2013.
[19] J. Wang et al., "Efficient verifiable fuzzy keyword search over encrypted data in cloud computing," Comput. Sci. Inf. Syst., vol. 10, no. 2, pp. 667–684, 2013.
[20] W. Sun et al., "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in Proc. 8th ACM SIGSAC Symp. Inf., Comput. Commun. Secur., 2013, pp. 71–82.
[21] Z. Fu, L. Xia, X. Sun, A. X. Liu, and G. Xie, "Semantic-aware searching over encrypted data for cloud computing," IEEE Trans. Inf. Forensics Security, vol. 13, no. 9, pp. 2359–2371, Sep. 2018.
[22] S. Kamara and C. Papamanthou, "Parallel and dynamic searchable symmetric encryption," in Proc. Int. Conf. Financial Cryptography Data Secur. Berlin, Germany: Springer, 2013, pp. 258–274.
[23] W. Sun, X. Liu, W. Lou, Y. T. Hou, and H. Li, "Catch you if you lie to me: Efficient verifiable conjunctive keyword search over large dynamic encrypted cloud data," in Proc. IEEE Conf. Comput. Commun. (INFOCOM), Apr. 2015, pp. 2110–2118..
[24] R. Bost, P.-A. Fouque, and D. Pointcheval, "Verifiable dynamic sym metric searchable encryption: Optimality and forward security," Cryptol. ePrint Arch., Jan. 2016.
[25] K. Kurosawa and Y. Ohtaki, "UC-secure searchable symmetric encryp tion," in Proc. Int. Conf. Financial Cryptogr. Data Secur. Berlin, Germany: Springer, 2012, pp. 285–298.
[26] J. Zhu, Q. Li, C. Wang, X. Yuan, Q. Wang, and K. Ren, "Enabling generic, verifiable, and secure data search in cloud services," IEEE Trans. Parallel Distrib. Syst., vol. 29, no. 8, pp. 1721–1735, Aug. 2018.
[27] M. Hinek, S. Jiang, R. Safavi-Naini, and S. Shahandashti, "Attribute based encryption with key cloning protection," Tech. Rep. 2008/478, 2008.
[28] S. Yu, K. Ren, W. Lou, and J. Li, "Defending against key abuse attacks in KP-ABE enabled broadcast systems," in Proc. Int. Conf. Secur. Privacy Commun. Syst. Berlin, Germany: Springer, 2009, pp. 311–329.
[29] Q. Zheng, S. Xu, and G. Ateniese, "VABKS: Verifiable attribute-based keyword search over outsourced encrypted data," in Proc. IEEE Conf. Comput. Commun., Apr. 2014, pp. 522–530.
[30] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Proc. Int. Conf. Theory Appl. Cryptograph. Techn. Berlin, Germany: Springer, 2004, pp. 506–522.