Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 10, October, 2021: 2100 - 2105

Designing Key-Escrow Free Attribute-Based Multi-Keyword Search with Dynamic Policy Updates in Cloud Computing

Dr D Bujji Babu[1], K Jaya Krishna[2], P Suneetha[3], D V Ramya Sri[4], M Srikanth[5], R Sivannarayana[6]
[1]Professor & HOD, [2] Associate Professor, [3,4,5,6] MCA Scholars, Department of MCA
QIS College of Engineering and Technology (Autonomous), Ongole

Abstract:
Able to find and manage access to cloud-encrypted data using attribute-based searchable encryption (ABSE) approaches. The following issues plagued prior ABSE systems, which were supposed to protect data while yet making it accessible. There's a key-escrow problem here that might lead to misuse of the user's private information. If you only search for one phrase, you may get results that have little to do with your original inquiry, yet they're still vulnerable in real-world situations. Static policy updating is a time- and resource-intensive process that can be costly to implement. Using multiple keywords and dynamic policy changes in the cloud, we propose a novel technique in this paper using the newly developed ABSE method, described as a 'free attribute-escrow.' We'll utilise this to tackle all of the difficulties described above (KAMS-PU). KAMS-PU is safe from attacks on authority and keyword selection (CKA) in the context of a random oracle, according to the security research In addition, a performance analysis shows that KAMS-PU is both effective and feasible in real-world settings.

KeyWords: Issues with attribute-based encryption, security, and data privacy, as well as policy updates

1. **Introduction**:

Because of the numerous advantages it offers, individuals and organisations alike are increasingly flocking to cloud computing services and enterprises. With The transfer of data to a cloud service raises concerns about physical storage capacity issues. Outsourcing has had an impact. Data records, information about credit or debit cards, and passwords may contain sensitive information (such as images or medical records). Priority one should be given to ensuring the safety and security of the information contained inside. It's possible that a curious cloud server is trying to get ahold of sensitive data in order to make use of it. One way to secure the secrecy of the acquired data is to encrypt it before transporting it to the cloud. On the other hand, traditional encryption is still accessible, but it prevents the search for decrypted data [2]. It's necessary to implement searchable encryption (SE) individuals to do a search using the key phrases that are important to them over encrypted data The personal preferences and limits on who gets access to what data in a large organisation are important considerations Pre-existing and new users play a crucial role in user growth. It is possible to search for data in the cloud using traditional SE approaches such as symmetric searchable encryption (SSE) [4–6] and searchable asymmetric/public-key encryption (PSE) [7]–[9]. A new [10] technique has been proposed to make shared data searchable in addition to encryption searchable based on characteristics with access control (ABSE). When it comes to the safeguarding of the keys, ABSE looks to well-respected authoritative figures (TA). The person who owns the data has it encrypted and the access controls are established in such a manner that only the user may decode it. The traits are contrasted with the policy's requirements. Other systems [11–19] developed from ABSE have since been proposed. However, the following issues plague the bulk of existing ABSE strategies: As a starting point, evaluate all of your possibilities. That's why multi-keyword searches are beneficial. The significance of [points 17–19]. Second, the strategy appears to be working. One of the problems with key escrow is that authorities can gain access to a third party's Data if the third party is curious. This is because authorities can construct users' access secrets. The attacker will have simple access to authority after that power has been compromised, thus they'll be able to discover and utilise the master key then. Another problem is that they haven't thought through how they'll deal with dynamic policy changes. Once data has been distributed into the cloud, a static policy change will make it unreachable if the local copy is utilised. When the policy's newest version has to be downloaded and installed by the data owner. Decrypt all cloud data using a new encryption key and then transmit it to the cloud with the new policy. Despite this, the person in charge of the data must communicate and compute a lot. As a result, creating a strategy is critical for swiftly and dynamically altering the policy without encountering issues with key escrow. With our proposal, we'll be able to remove the necessity for an escrow service and instead provide a dynamic policy-update multi-keyword search.

2. Literature Survey

Numerous schemes with various features have been proposed since the discovery made by Song et al[3] of the first searchable encryption key, scheme. Searchable encryption (SE) is a cryptographic approach divided into three groups: searchable cryptography that may be found in a symmetrical to asymmetrical/public manner [4–6]. When employing a cloud storage service, attribute-based searchable encryption (ABSE) ensures data security by controlling access with fine-grained accuracy. ABSE may be divided into two types: general and specific. ABSE stands for Key Provisioning Absence and Basic Security Encryption Absence, respectively (CP-ABSE). Primary and Secondary Education, a user's attributes must be in accordance with regulations that control access to CPABSE In order to decode the data, you must have the decryption key. It is because of these characteristics that CP-ABSE is a superior way for devising secure search algorithms.

Utilizing multiple owners and users as the basis, Sun et al. [11] developed a first CP-ABSE system in 2014, presenting a search approach that is both verifiable and file-level using conjunctive keywords permission provided by users. The system has no chance against CKA. It was later stated by Dong et al. in 2015 that the system was built with portability in mind for mobile phones and tablets.

In 2016, Li et al. [15] discovered that as the complexity of regulations pertaining to access increases, so does the price of ciphertext. Decryption keys were outsourced to save money on computational costs, and they are protected from plaintext-based assaults since they are provided by a system that was previously outsourced (CPA).

Similarly, Han et al. [16] in 2017 offered a scheme to maintain a constant rate of searchable ciphertext, encryption calculation cost, trapdoor duration and search algorithm. In 2017, Miao and colleagues proposed to build a system to deal with more practical aspects, which was adopted in

2018. Cui is also a pattern follower. Revocation of a user's permissions was recommended by et al. [18] in 2018 as a way to be effective. The encrypted index is kept on the Cloud server and is responsible for sending backdoors to users.

It was proposed in 2019 that Cao et al. [19] developed a fine-grained control over who gets access to search data as part of a plan to avoid search data leakage. The ideas [10]–[16] do not, however, provide a multi-keyword search (MKS). Permission to conduct MKS backdoor searches using a wide range of search phrases to produce more targeted results Similarly, the same body oversees all ABSE schemes [21]. the process of devising the top-secret code Because of this, key escrow could be implemented. both the computational burden on a single machine and the problem There will be a great deal of power in this situation. No other actions are taken. dynamically alter the policy in light of fresh data. In corporate contexts, it's critical to keep policies up to current on the fly since regulations might change quickly in real-world applications. Key escrow is addressed while dynamic policy changes are conducted as a collaborative effort under ABSE. scheme. As far as we know, ABSE does not exist.

3. System Model

Entities and security definitions for the system model are laid forth in this section.

The following is how the KAMS-PU system model includes all five parties:

Owner of the Data: The owner of the data is regarded as trustworthy. Prior to outsourcing the ciphertext form of the document and index to the cloud server, DO encrypts the documents F with an access policy (M,) and a keyword index I with an index of 1, 2's' and an index of 'n's'.

To decrypt the findings from the cloud server, Data User (DU) uses the secret key SK to create a trapdoor.

Cloud Server (CS): CS has a reputation for being honest, but also curious. It indicates that CS does everything perfectly, yet it's interested about the information that users provide. If it finds anything, CS uses the DU's trapdoor to search the encrypted index and returns it to the DU.

Trusted Authority (TA): TA is a well-trained authority in this architecture, although it is not completely trusted. There are two sets of public parameters generated: one is part of PK1 and the other is part of MSK1. In order to produce part of the secret key SK1 for every particular user, it talks with AA to generate part of MSK1.

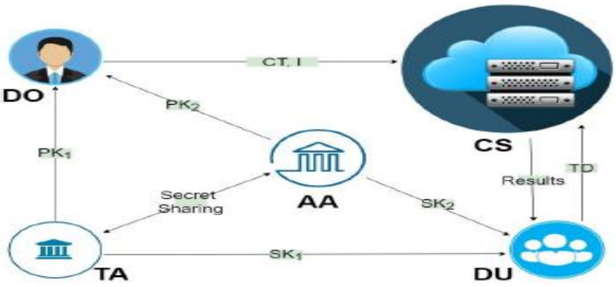

Fig. 1: The System Model of KAMS-PU

In this design, the Attribute Authority (AA) is also a semi-trusted authority. For every user t, an AA creates part of the public parameters PK2 and part of the master secret key MSK2, and it works along with TA to generate the corresponding portion of the secret key SK2.

The model's functioning is shown in Figure 1: To begin, TA and AA generate a set of open parameters as well as a master secret. MSK is a secret key that only the owner knows about. To index the papers, DO uses cloud storage to encrypt and index them. A trapdoor is generated by authorised users using their own search keywords so that they may obtain relevant results. As a result, the index is checked to see if the value given by CS matches the trap's entrance index entry If a match was found, this function returns 1; otherwise, it returns 0. Finally, those that meet the criteria for access may utilise the secret key to decode the findings.

4. Performance Analysis

This section looks at how KAMS performs. Both philosophically and experimentally, PU's are sound. In this section, we took a closer look at the KAMS-PU organization's theoretical performance. Set-up, key generation and cleaning calculation times are used to establish a search index and trapdoor algorithms for discovering items on the internet, and these times are used to measure the program's efficacy. Calculation costs were determined by comparing the tb bilinear pairing process to both exponential and linear procedures. [17] and [18] theoretical computation analysis is compared to KAMS- PU's To get started, you'll need (4+nu)te+2tb for KAMS-PU. And so forth, with a total of nu universal qualities to consider. Access to the vault requires 4+ns=te, where ns is the number of characteristics that will be created for user t when the vault is unlocked. Indicator creation has the following associated costs: T, where t is a triangular array of length three. It will cost (4 + d)te to build a trapdoor because of the query's key words. This will take 4TB of space to do a search. Due to the lesser degree of complexity in operations and the calculation cost of hash operations compared to tb, te, we don't include the multiplication computation cost in this case. In addition, we discovered that it takes 5+nlte+tb time to complete encryption, where Access structure and decryption have nl features, each algorithm of three petabytes each.

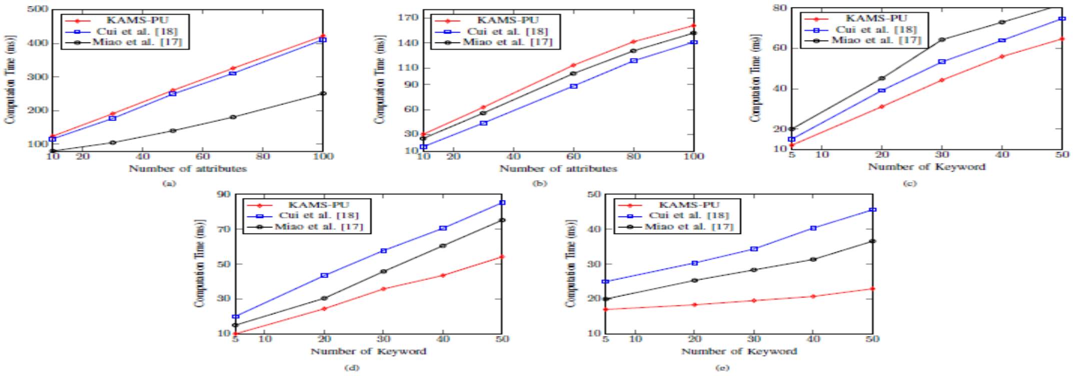

Fig. 2: **Computation cost: (a)** Computation cost for setup **(b)** Computation cost for key generation **(c)** Computation cost for index generation **(d)** Computation cost for trapdoor **(e)** Computation cost for search.

Using different algorithms and schemes, we compare the difference in computation time (measured in milliseconds). Setup and takedown durations are shown in Figure 2 (a) and (b), as well as a rise in when the number of characteristics increases. There are more features in our plan than in others due to the fact that these computer algorithms are in charge of calculating the costs of the two authorities. Figures 2 (c) and (d) show how long it takes to create an index and a trapdoor as the number of keywords increases. Figure 2 (e) shows how long the search takes in comparison to other activities with a large scope since there are so few of them. As can be seen in Fig. 3, the computation is broken down even further. The cost of altering the policy will be prohibitive. Type 1 procedures, as illustrated in Figure 3, have low computation costs for both the data owner and the service provider. The cloud is where the server is located. Type 3 policy adds new features to the current ones, increasing the computational effort compared to types 1 and 2. Only the pairing and exponent techniques are used in the experiment to determine policy changes. In comparison to a static policy update, KAMS-computation PU's burden is lowered since it uses previously decrypted information while changing the policy. When it comes to data security operations, KAMS-PU is the preferred and most cost-effective option.

5. Conclusion

In this post, we introduced a new ABSE approach we called It is now able to do multiple keyword searches without the need of key escrow, and dynamic encryption policy changes are now possible (KAMS-PU). Multi-keyword search tool KAMS-PU works by establishing a secure index that allows the user to search for documents with a wide range of pertinent keywords. In addition, the method allows you to restrict access with fine-grained security to prevent unauthorised entry. Both of these problems are addressed by the KAMS-PU key escrow system, which generates the secret key with the assistance of two authorities. Due to the dynamic updates provided by KAMS-PU the old policy may be replaced with a new one without having to download it from the web. KAMS-PU is secure against CKA attack and malicious authority assaults in the paradigm of a random oracle, according to the security evaluation results under the DBDH assumption. A mix of theoretical and experimental research has demonstrated that our strategy is workable and viable. We want to add enhancements to KAMS-PU in the future to make user revocation operate better.

References

[1] A. Kalapatapu and M. Sarkar, “Cloud computing: An overview,” *Cloud Computing: methodology, systems, and applications*, pp. 3–29, 2012.

[2] E.-J. Goh *et al.*, “Secure indexes.” *IACR Cryptology ePrint Archive*, vol. 2003, p. 216, 2003.

[3] D. X. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in *Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000*. IEEE, 2000, pp. 44–55.

[4] Q. Wang, M. He, M. Du, S. S. Chow, R. W. Lai, and Q. Zou, “Searchable encryption over feature-rich data,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 3, pp. 496–510, 2016.

[5] B. Wang and X. Fan, “Search ranges efficiently and compatibly as keywords over encrypted data,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 6, pp. 1027–1040, 2016.

[6] J. Zhu, Q. Li, C. Wang, X. Yuan, Q. Wang, and K. Ren, “Enabling generic, verifiable, and secure data search in cloud services,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 8, pp. 1721–1735, 2018.

[7] Z. Wan and R. H. Deng, "Vpsearch: Achieving verifiability for privacypreserving multi-keyword search over encrypted cloud data," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 6, pp. 1083–1095, 2016.

[8] W. Zhang, Y. Lin, and G. Qi, "Catch you if you misbehave: Ranked keyword search results verification in cloud computing," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 74–86, 2018.

[9] J. Li, R. Ma, and H. Guan, "Tees: An efficient search scheme over encrypted data on mobile cloud," *IEEE Transactions on Cloud Computing*, vol. 5, no. 1, pp. 126–139, 2017.

[10] Q. Zheng, S. Xu, and G. Ateniese, "Vabks: verifiable attribute-based keyword search over outsourced encrypted data," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 522–530.

[11] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, "Protecting your right: Attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 226–234.

[12] S. Li and M.-Z. Xu, "Attribute-based public encryption with keyword search," *Chin. J. Comput.*, vol. 37, no. 5, pp. 1017–1024, 2014.

[13] Q. Dong, Z. Guan, and Z. Chen, "Attribute-based keyword search efficiency enhancement via an online/offline approach," in *2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2015, pp. 298–305.

[14] K. Liang and W. Susilo, "Searchable attribute-based mechanism with efficient data sharing for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1981–1992, 2015.

[15] J. Li, X. Lin, Y. Zhang, and J. Han, "Ksf-oabe: outsourced attributebased encryption with keyword search function for cloud storage," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 715–725, 2016.

[16] J. Han, Y. Yang, J. K. Liu, J. Li, K. Liang, and J. Shen, "Expressive attribute-based keyword search with constant-size ciphertext," *Soft Computing*, vol. 22, no. 15, pp. 5163–5177, 2018.

[17] Y. Miao, J. Ma, X. Liu, X. Li, Z. Liu, and H. Li, "Practical attributebased multi-keyword search scheme in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3008–3018, 2017.

[18] J. Cui, H. Zhou, H. Zhong, and Y. Xu, "Akser: Attribute-based keyword search with efficient revocation in cloud computing," *Information Sciences*, vol. 423, pp. 343–352, 2018.

[19] Q. Cao, Y. Li, Z. Wu, Y. Miao, and J. Liu, "Privacy-preserving conjunctive keyword search on encrypted data with enhanced fine-grained access control," *World Wide Web*, pp. 1–31, 2019.

[20] K. Yang, X. Jia, and K. Ren, "Secure and verifiable policy update outsourcing for big data access control in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3461–3470, 2014.

[21] U. Varri, S. Pasupuleti, and K. Kadambari, "A scoping review of searchable encryption schemes in cloud computing: taxonomy, methods, and recent developments," *The Journal of Supercomputing*, pp. 1–30, 2019.

[22] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, "Charm: a framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013.