Multi-Keyword Search with a Threshold for Cloud-Based Group Data Sharing

[1] Dr. K. Nagi Reddy *,
Professor & HoD
Department of Information Technology, Lords Institute of Engineering and Technology, Hyderabad, India,

[2] Mr.Mohd. Mateen Ahmed,
Assistant Professor,
Department of Information Technology, Lords Institute of Engineering and Technology, Hyderabad, India,

[3] Ms.B.Nagalakshmi
Assistant Professor
Department of Information Technology, Lords Institute of Engineering and Technology, Hyderabad, India,

[4]Mrs.AzeemUnissa
Assistant Professor
Department of Information Technology, Lords Institute of Engineering and Technology, Hyderabad, India,

Abstract— A common cryptographic basic, Searchable Encryption (SE), is used in the development of ciphertext retrieval systems that have a wide range of applications. Current SE methods, on the other hand, do not often provide threshold access control (i.e., data users must collaborate to perform search and decryption operations over encrypted cloud data), which is becoming more popular in the research community. Beginning with the most basic TMS scheme, we provide a Shamir's secret sharing technique for cloud-based group data sharing, which incorporates Threshold Multi-keyword Search (TMS) with Shamir's secret sharing technique to achieve threshold multi-keyword search, threshold decryption, and a small record ciphertext size. Then, we build on top of this fundamental TMS to provide threshold result verification as well as threshold traceability (referred to as enhanced TMS). Furthermore, the enhanced TMS has been upgraded to enable public result verification as well as dynamic operations using the public verifier and better hash tables, which are also included in the enhanced TMS package. Both the basic TMS and the improved TMS are semi-adaptively secure, and they can withstand the Chosen Keyword Attack, as shown by our rigorous security study (CKA). Our theoretical assessment and practical tests show that both systems have the potential to be beneficial.

Key Terms—

1. INTRODUCTION

People, companies, and governments all utilize cloud computing because it enables them to exchange data (documents, pictures, and other types of files) with specific/intended recipients in a

group environment, which saves time and money. Despite the fact that cloud-related security and privacy are two subjects that have been thoroughly researched in the literature, there are still problems that need to be handled in their fullness. For example, we may offshore extremely sensitive data in encrypted form exclusively in order to prevent certain cloud providers (and their workers) from gaining access to the data housed on these cloud servers. In reality, however, this restricts the users' ability to search through such encrypted cloud data because of the encryption. In order to combat this, there has been a resurgence of interest in developing incomplete but realistic Searchable Encryption (SE) methods. Such methods do not compromise data security or usability, and they have the potential to be utilized in a variety of contexts, including task suggestion in crowdsourcing [8, cloud-based healthcare services [9] as well as group data sharing [10]. As a result, a practical SE should have been capable of achieving features such as expressive search but also cost-effective storage in addition to offering high levels of security assurance [11, 12]. Nevertheless, one of the primary drawbacks of traditional SE systems is that they impose no constraints on access control in group-oriented applications, which is a major restriction in today's distributed computing environment (i.e., social network, wireless body area network, etc.). In other words, there is a danger of illegal access, which may jeopardize data confidentiality. Numerous studies have been conducted to integrate Ciphertext-Policy Attribute-based Keyword Search (CP-ABKS) with SE in order to provide keyword-based ciphertext retrieval while simultaneously offering an extremely fine degree of access control. Nonetheless, encryption and decryption have a relatively significant overhead when compared to other techniques1. Since previously mentioned [18, 19], CPABKS may not be the optimal cryptographic tool for lightweight deployments, as encryption and decryption operations are often conducted on computationally inefficient devices (e.g., Internet of Things (IoT) or Industrial Internet of Things (IIoT) devices). Another way to put it is that the four primary characteristics of a good SE system are security, expressive query, access control, and overall performance. Despite the vast number of published SE methods, the creation of realistic SE systems that additionally provide threshold access [20] remains an uncharted territory. When it comes to group-based data sharing applications [21] (such as electronic auctions or electronic voting), we may not be able to place our whole trust in a single individual. Rather than that, we may put our trust in a group of individuals who will have access to our sensitive information. One famous example is electronic voting, in which a pool of people is trusted to disclose the final result but are not permitted to reveal the identities of those who voted in the process. Furthermore, it is essential that this data be accessible even if certain members of the authorized group are hacked or otherwise unavailable. For threshold access in SE, the Threshold Public-Key Encryption (TPKE) technique [22, [23] may be used in conjunction with other methods. This method enables a group of authorized data users to collectively create a valid trapdoor & decrypt search results if a certain number of them meet the criteria set forth above. Some examples of adopting this method include those mentioned in [24] and [25], among others.

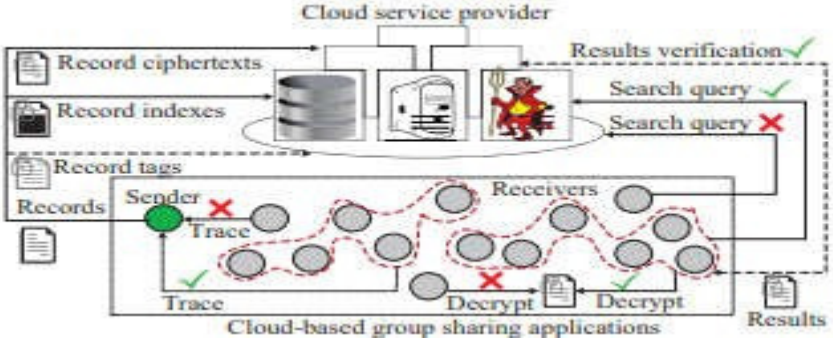

Fig 1: Feature of Basic TMS

If you use this method to enable multi-keyword search, it will often result in a large trapdoor size, and it will not be able to provide result verification [26] in a cloud computing environment that is semi-honest but suspicious, or threshold traceability [27] if there is a disagreement. A hacked or attacked cloud service [28, 29] may fabricate or tamper with results in order to benefit from different incentives, for example. Schemes such as those described in [30] and [31] tried to prevent the semi-honest-but-curious cloud server from providing inaccurate search results; however, owing to the usage of the Bloom filter, these solutions have large false-positive rates and should be avoided. Furthermore, individual traceability [32] and [33] in previous group signature systems enables each group member to disclose the actual signer's identity, but if each group member is given this capacity in certain applications, this may result in excessive misuse. It is possible to prevent such a restriction and to settle any potential conflicts by using threshold traceability. To the best of our knowledge, there is no current solution that addresses all of the constraints listed above in a single scheme, and this is the contribution we want to offer with this project. By using broadcast encryption [34, [35], we first design the fundamental Threshold Multikey Word Search (TMS) method in the context of a group-oriented data sharing architecture, which we next implement. Then, utilizing democratic group signature [27, 32], we modify the basic TMS scheme to create the improved TMS, which supports threshold result verification and threshold traceability while also enabling threshold traceability.

II LITERATURE REVIEW

A growing number of organizations are using cloud-based group information sharing, especially in collaborative and remote environments. It is still a research problem, however, to figure out how to accomplish flexible and secure data exchange among group-oriented participants. The following are two methods that are extremely closely linked to one another: Encryption that can be searched (SE). SE schemes, such as those proposed may be generally divided into two types: The scheme's symmetry dictates its name: Symmetric SE (SSE) and Asymmetric SE (ASE). With regard to group data sharing, SSE systems are not efficient. In fact, in order to provide secure access to data, the senders (the owners of the data) must exchange the secret keys with the recipients (the users of the data) through certain security channels, which causes a large level of information and key administration overhead. Due to the fact that ASE schemes don't need the presence of security channels, they may be implemented in a multi-user context, such as group-oriented settings. Additionally, in addition to letting the sender to choose whether or not to reveal their data, ASE systems should also offer strong security. The ability to enable multi-keyword search or Boolean search is essential for expressive questions, because a single keyword search may return a significant number of non-relevant results. Expressive queries need to be able to handle multi-keyword search or Boolean search in order to get relevant search results. Also, keep in mind that permission levels are implemented at the individual record level, which means that each individual message recipient may need fine-grained search permissions applied at the record level. Chetan Zheng et al. integrated CP-ABE with SE and came out with the CP-ABKS (coarse-grained keyword search) algorithm that achieves fine-grained keyword search. The solution suggested by Xu and colleagues combines identity-based encryption with ABE methods to provide a more secure, fine-grained group data sharing system that efficiently updates ciphertexts without the need of a delegated key. In such case, it would not be possible to search for keywords across encrypted cloud data using this strategy. To circumvent the restrictions in mobile crowdsourcing, the researchers from Miao et al. [14] created a finely-grained search method for multiple keywords and their usages that each job submitter may do in the cloud, in an effort to overcome this constraint. In recent years, many CP-ABKS techniques (13, 14) have been proposed, and the literature is full with them. An example of an honest but curious cloud server that performs search

operations in an unethical way and distributes erroneous results is the CP-ABKS technique. Fu et al. proposed a new privacy-aware public auditing approach for cloud data sharing for groups in their study. While all of these approaches can provide access control beyond the needed threshold, none of them fully meet the needs of group-oriented data sharing applications. Shamir's secret sharing technique, together with Wang et al. and Kuchta and colleagues [25], provide the groundwork for the development of the threshold keyword search idea, which was later given by Qian et al. [20] on the basis of the identity-based threshold decryption concept. While both methods can't support searching for multiple keywords, and have larger backdoor sizes that grow in proportion to the amount of terms searched, each approach has its own separate drawbacks. Creative use of public key encryption (TPKE). TPKE algorithms enable data accessibility even if some members are not connected, as conventional public-key encryption methods do. To repeat, this cryptographic primitive includes three types of threshold broadcast encryption, as well as group signature threshold. Because of the absence of adaptive security, the use of traditional TPKE techniques causes larger ciphertext sizes. A secure TPKS with constant ciphertext length was proposed by Qin et al. and is based on adaptive broadcast encryption. The technique used in this example, on the other hand, has a huge private key size proportional to the group size, and therefore cannot identify the individual who submitted the message. With conventional group signature systems, the ability of the members of the group to hide their identities while still enabling the administration of the group to monitor the senders is maintained. A democratic group signature method was developed by Zheng in order to address the constraints of traditional group encryption schemes, such as the need of a trusted group manager and only enabling individual traceability.

III PROPOSED METHODOLOGY

To begin, we design the fundamental Threshold Multikeyword Search (TMS) method in the context of a group-oriented data sharing framework, which is accomplished via the use of broadcast encryption. Then, utilizing democratic group signature, we modify the basic TMS scheme to create the improved TMS, which supports threshold result verification and threshold traceability while also enabling threshold traceability. The following is a synopsis of the contributions made in this paper:

• Multi-keyword search with a threshold 2nd, there is no such thing as a formalized Instead of the usual situation in which each authorized data user is permitted to access encrypted data, our suggested SE schemes consider the group sharing scenario, which is in contrast to prior SE schemes that supported multi-keyword search. Assuming that a group is approved, By using their respective secret keys, as stated above, our basic or extended TMS systems allow each group member to build an unique trapdoor share based on a list of keywords by utilizing their respective secret keys. It also provides an opportunity for at least a threshold amount of group members to join together to create the final trapdoor, which ensures that the ciphertexts are accessible even if some group members are corrupted or otherwise unavailable. The Lagrange interpolation method is used to accomplish this. TMS's basic and enhanced versions, in contrast to previous SE systems, do not grow in proportion to the amount of keywords searched, as was the case with earlier SE systems. As a consequence, the techniques that have been proposed may be applied on devices that have limited resources. In addition, the record ciphertext size and threshold decryption overhead are also minimal in comparison to other algorithms. The record ciphertext size3 of our basic and extended TMS methods is not as large as that of previous threshold public encryption schemes, which was the case with previous threshold public encryption systems. When compared to previous threshold public encryption schemes, our basic and extended TMS methods are not as large as that of previous threshold public encryption systems. To put it another way, the size of the

group has no impact on the size of the ciphertext produced by our basic or enhanced TMS, respectively. Furthermore, previous schemes allow for the recovery of the decryption key by a single authorized data user, whereas our basic or enhanced TMS requires that a minimum number of group members generate their decryption shares and then integrate them in order to cooperatively recover the decryption key for each record. As a consequence, previous techniques entrust the whole of the decryption task to data consumers, causing a substantial rise in computation costs for data consumers that lack sufficient computer capabilities. By using an outsourced decryption technique, our enhanced TMS system is able to significantly reduce the amount of decryption overhead that must be performed.

• Verification of the threshold result. In our improved TMS scheme, rather than assuming an honest but suspect cloud server, as is the case in traditional SE systems, we examine a semi-honest but suspicious cloud server. This cloud server performs the requested search operations honestly the vast majority of the time, but it may occasionally return partial false search results as a result of financial incentives (e.g., saving storage space and computing resources) in our enhanced TMS scheme (e.g., saving storage space and computing resources). A further disadvantage of previous search engine systems was that they only allowed for the verification of the correctness of search results by a third party or a single user. A verification result obtained by one group member does not necessarily have the capacity to convince the other members of the group in a group sharing scenario. A homomorphic verifiable tag is added to each record as part of our enhanced TMS technique, which helps to assure the trustworthiness of search results by mandating that a minimum number of receivers be used to verify the dependability of the search results.

• Traceability of the threshold. Instead of relying on the group manager, our improved TMS system does not need one and guarantees threshold traceability rather than individual traceability, in contrast to traditional group signature schemes. Previous individual tracing methods allowed any group member to track the signer's identity, which resulted in a high level of misuse in the first place. To address this issue, we have developed an improved TMS that provides threshold traceability. We have developed a new TMS system in which each group member can sign a record on behalf of the specified group by producing its matching signature as well as certain auxiliary information, and this authorized group can work together to determine who signed the record in the event of a disagreement. As an added bonus, we demonstrate that both the basic and improved TMS systems are semi-adaptively safe and capable of surviving a kind of dictionary assault known as the Chosen Keyword Assault. In order to prevent the semi-honest but curious cloud server from successfully manufacturing valid record tags, the improved TMS will be implemented. Finally, we compare the performance of our proposed methods to that of two other competing systems. which we obtained from a publicly available dataset.

IV PROPOSED SYSTEM MODEL

According to the system paradigm (see Fig. 2) of our proposed TMS systems, four entities are involved: the sender, receivers, Cloud Server (CS), and Trusted Authority (TA). The sender and receivers are the most important elements (omitted in Fig. 2). Enhanced TMS system models provide two additional capabilities (e.g., traceability and outcomes verification, as indicated by the red dashed line) that are not included in the basic TMS system model. These additional functionalities are provided by the material included inside the red dashed box. The following is a description of the role played by each entity:

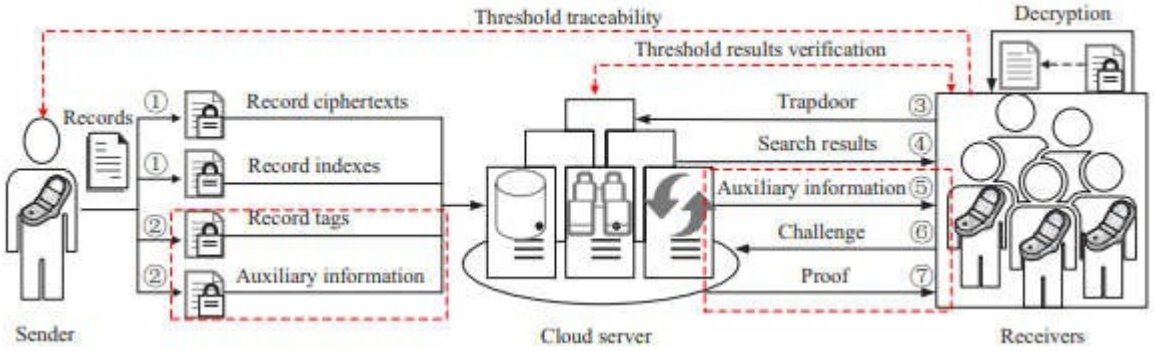

Fig 2: System Model

• *Sender*. Although the sender, who is each and every team member, will share his data with selected receivers (i.e., a subset of team members), he insists that only a small number of receivers be permitted to conduct search queries and decode ciphertexts cooperatively. A record ciphertext and an index are generated by the sender in basic TMS (step 1), while auxiliary information such as record tags and auxiliary information are generated by the sender in enhanced TMS (step 2). (step 2).

• *Receivers*. When using a single receiver, multiple receivers (rather than a single receiver) can conduct search queries by generating the trapdoor (step 3), as shown in a list of keywords, and decrypting the returned search results in both basic and advanced TMS schemes, whereas a single receiver can only conduct search queries. Additional benefits include the ability to monitor a threshold number of recipients' actual identities in the case of an alleged dispute and the ability to verify the correctness of search results when the challenged information (step 6) is sent to Customer Service in the updated TMS.

• *Cloud server*. It is possible for CS to offer on-demand storage and compute services when it has adequate resources. It is CS's regular but important responsibilities to match the indexes with the trapdoor and then provide the required search results (step 4) to the receivers, which is an essential service. The CS will also provide additional information (step 5) and proof information (step 7) to the receivers in order for them to be able to monitor the sender's real identity and verify the correctness of search results in enhanced TMS, using the information provided by the CS. As a side benefit, with improved TMS, CS may be used to provide intermediate results on behalf of receivers, allowing for a significant decrease in the enormous decryption overhead which is associated with basic TMS.

• *Trusted authority*. The system and the master key are created in TA, and the key is then shared with members of the group. Although TA isn't needed to be accessible if no new or leaving group members routinely join or leave the original group, it should be emphasized that TA is available if new or departing group members do join or leave the original group.

V Modules Implementation

Data Owner Module

This module assists the owner with registering such data as well as including login information. This module assists the owner in uploading his or her file while encrypting it using the CP-ABE method. This guarantees that the files are secured from being accessed by unauthorized users. In order to maximize efficiency, the data owner wishes to outsource his collection of papers F =f1, f2; :::; fn to a cloud server in encrypted form while still retaining the capacity to search through the documents for efficient usage. In our approach, the data owner first constructs a safe searchable tree index I from document collection F, after which he or she creates an encrypted document collection C for collection F. Following that, the data owner companies outsource the encrypted

collection C and the wildcard access policy I to a cloud server and securely distributes key information for trapdoor construction and document decryption to authorized data users in a controlled environment. Additionally, the data owner is responsible for maintaining his papers stored on a cloud-based server.

Data User Module

This module contains the login information for users who have registered. This module is designed to assist the client in searching for information in a file utilizing the multiple wildcard access policy idea and retrieving an accurate result list depending on the user query. Before entering the activation code, the user will need to choose the appropriate file and register his or her information. He or she will then get an activation code through email. After that, the user may download the Zip file and extract the contents of that file. Data users are those who have been granted permission to view the documents of the data owner. The authorized user may construct a trapdoor TD in accordance with search control methods in order to retrieve k encrypted documents from a cloud server using t wildcard query keyword combinations. The data user may then decrypt the documents using the secret key that has been provided with them.

Cloud Server Module:

Aims to assist the server in encrypting documents using the CP-BE algorithm and converting the encrypted documents to Zip files with activation codes, which are subsequently sent to the user for download. The cloud server holds the encrypted document collection C as well as the encrypted searchable wildcard access policy I for the data owner. When the cloud server receives the trapdoor TD from the data user, it searches for the matching collection of encrypted documents using the wildcard access policy I, and then returns the collection of encrypted documents. Furthermore, following receipt of the update information from the data owner, the server is responsible for updating the index I and document collection C in accordance with the information received. Several studies on safe cloud data search have made use of the phrase "honest but inquisitive" to describe the cloud server used in the suggested system.

VI CONCLUSION

As a result of the observation that there really is no scheme available that provides flexible access control in a group oriented data sharing setting, the development of a basic TMS scheme and an enhanced TMS scheme was motivated. Both schemes achieve versatile features such as threshold encryption, threshold result verification, and threshold traceability through the use of threshold decryption. Through the use of cutting edge methods, the improved TMS scheme, in particular, may be further upgraded to offer public result verification and dynamic operations, among other things (rather than reinventing the wheel). Following that, it is shown that both basic TMS and improved TMS systems may provide semi-adaptive security while also resisting CKA. Furthermore, it is shown that the improved TMS method provides unforgeability of record tags. Our suggested methods are efficient in reality, as shown by our evaluation on a publicly accessible NSF dataset, which was made available to us. In spite of the fact that our suggested methods are capable of producing small record ciphertext sizes, when the record indexes (and record tags) are taken into account, these schemes still have a somewhat significant computation and storage cost. For this reason, as part of our future work for this article, we will concentrate on developing a lightweight encryption method that does not compromise the other characteristics of the system.

VII REFERENCES

[1] B. Cui, Z. Liu, and L. Wang, “Key-aggregate searchable encryption (kase) for group data sharing via cloud storage,” IEEE Transactions on computers, vol. 65, no. 8, pp. 2374–2385, 2016.

[2] Y. Miao, J. Ma, X. Liu, X. Li, Q. Jiang, and J. Zhang, “Attributebased keyword search over hierarchical data in cloud computing,” IEEE Transactions on Services Computing, pp. 1–14, 2017.

[3] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. IEEE Symposium on Security and Privacy (S&P'00). IEEE, 2000, pp. 44–55.
[4] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Proc. International conference on the theory and applications of cryptographic techniques (EUROCRYPT'04). Springer, 2004, pp. 506–522.
[5] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "Dual-server publickey encryption with keyword search for secure cloud storage," IEEE transactions on information forensics and security, vol. 11, no. 4, pp. 789–798, 2016.
[6] X. Zhang, Y. Tang, H. Wang, C. Xu, Y. Miao, and H. Cheng, "Latticebased proxy-oriented identity-based encryption with keyword search for cloud storage," Information Sciences, vol. PP, pp. 1–15, 2019.
[7] R. Chen, Y. Mu, G. Yang, F. Guo, X. Huang, X. Wang, and Y. Wang, "Server-aided public key encryption with keyword search," IEEE Transactions on Information Forensics and Security, vol. 11, no. 12, pp. 2833– 2842, 2016.
[8] J. Shu, K. Yang, X. Jia, X. Liu, C. Wang, and R. Deng, "Proxy-free privacy-preserving task matching with efficient revocation in crowdsourcing," IEEE Transactions on Dependable and Secure Computing, pp. 1–14, 2018.
[9] R. Zhang, R. Xue, and L. Liu, "Searchable encryption for healthcare clouds: a survey," IEEE Transactions on Services Computing, vol. 11, no. 6, pp. 978–996, 2018.
[10] J. Shen, T. Zhou, D. He, Y. Zhang, X. Sun, and Y. Xiang, "Block designbased key agreement for group data sharing in cloud computing," IEEE Transactions on Dependable and Secure Computing, vol. PP, pp. 1–15, 2017.
[11] R. Chen, Y. Mu, G. Yang, and F. Guo, "Bl-mle: block-level messagelocked encryption for secure large file deduplication," IEEE Transactions on Information Forensics and Security, vol. 10, no. 12, pp. 2643–2652, 2015.
[12] Y. Zhang, C. Xu, H. Li, K. Yang, J. Zhou, and X. Lin, "Healthdep: An efficient and secure deduplication scheme for cloud-assisted ehealth systems," IEEE Transactions on Industrial Informatics, vol. 14, no. 9, pp. 4101–4112, 2018.
[13] Y. Miao, X. Liu, K.-K. R. Choo, R. H. Deng, J. Li, H. Li, and J. Ma, "Privacy-preserving attribute-based keyword search in shared multi-owner setting," IEEE Transactions on Dependable and Secure Computing, vol. PP, pp. 1–15, 2019.
[14] 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, 2018.
[15] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attributebased encryption," in Proc. IEEE symposium on security and privacy (S&P'07). IEEE, 2007, pp. 321–334.
[16] Z. Wan, J. Liu, and R. H. Deng, "Hasbe: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing," IEEE transactions on information forensics and security, vol. 7, no. 2, pp. 743–754, 2012.
[17] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," IEEE Transactions on information forensics and security, vol. 8, no. 8, pp. 1343–1354, 2013.
[18] P. Xu, S. He, W. Wang, W. Susilo, and H. Jin, "Lightweight searchable public-key encryption for cloud-assisted wireless sensor networks," IEEE Transactions on Industrial Informatics, vol. 14, no. 8, pp. 3712– 3723, 2018.
[19] K. N. Alharbi, X. Lin, and J. Shao, "A privacy-preserving data-sharing framework for smart grid," IEEE Internet of Things Journal, vol. 4, no. 2, pp. 555–562, 2017.

[20] J. Baek and Y. Zheng, "Identity-based threshold decryption," in Proc. International Workshop on Public Key Cryptography (PKC). Springer, 2004, pp. 262–276.
[21] C. Esposito, M. Ficco, A. Castiglione, F. Palmieri, and A. De Santis, "Distributed group key management for event notification confidentiality among sensors," IEEE Transactions on Dependable and Secure Computing, 2018.
[22] D. Boneh, X. Boyen, and S. Halevi, "Chosen ciphertext secure public key threshold encryption without random oracles," in Proc. Cryptographers' Track at the RSA Conference (CT-RSA'06). Springer, 2006, pp. 226–243.
[23] C. Delerablee and D. Pointcheval, "Dynamic threshold public-key en- ´ cryption," in Proc. International Cryptology Conference on Advances in Cryptology (CRYPTO'08). Springer, 2008, pp. 317–334.
[24] P. Wang, H. Wang, and J. Pieprzyk, "Threshold privacy preserving keyword searches," in Proc. International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'19). Springer, 2008, pp. 646–658.