EFFECTIVE DATA DE-DUPLICATION AND RECOVERY USING SEARCHABLE KEYWORD AND PUBLIC KEY ENCRYPTION

Syeda Samana Fatima
CSE Department,
Shadan women's College of Engineering and Technology,
Hyderabad, India.
fatimasyedasamana@gmail.com

Dr. P. Senthil Kumar
CSE Department,
Shadan women's College of Engineering and Technology,
Hyderabad, India.
psenthilkumarshadan@gmail.com

Dr. K. Palani
CSE Department,
Shadan women's College of Engineering and Technology,
Hyderabad,India.
principalswcet2020@gmail.com

ABSTRACT

Clients need for archived data is growing because of the current knowledge bursting, and wireless storage is now the recommended option for both people and businesses. By utilizing cloud computing, which enables users to transfer and backup their files, businesses can save cost on storage. Multiple users maintaining similar data results in a huge reduction in the amount of memory that cloud servers require. While repetitions in raw information can be easily erased, cloud services, which are typically semi-trusted, require data to be stored after encryption in order to safeguard user privacy. Our research addresses three primary problems: the safety of public key that can be searched indexes and matching links in the encrypted text are identical to make it possible for the safe deletion of the trapdoor; data recovery in encrypted text for different readers; and secure decoding. The cipher text chain table of the stored copy of the duplicate file has the data user's re-encryption key appended to it. The data user obtains the file by using the private key to decrypt the modified cipher text, which is produced by the cloud server through proxy re-encryption and uses the re-encryption key. Security study as well as experimental simulation analysis show that the suggested solution is safe and efficient.

1. INTRODUCTION

One of the main benefits of cloud computing technology is cloud storage, which makes it simple and quick for users to share and backup data. This can save customers money on storage costs and increase productivity. with the technology for cloud computing becoming more sophisticated. The industry is full of Cloud Service Providers (CSPs), including Baidu Cloud, Amazon Cloud, and other well-known CSPs. Users will upload and keep their private information in the cloud server's data storage center, which is run and maintained by the CSP. However, this frequently leads to security problems with cloud computing. Personal files, corporate contracts, user transaction records, environmental geographic data, and other sensitive data will be stored on the cloud server for businesses or individual users. Sensitive data leaks and user privacy breaches have happened, though, and there are even more CSPs due to the sale of user data for business profit. There should be more focus on the problem of data security in cloud storage. The world's users are producing an exponential amount of data, which is driving the rapid development of big data and cloud computing and driving up demand for cloud servers. The demand for enormous data storage will be effectively addressed by deduplicate data. While user data concerns the privacy of the user, uploading or storing it in plaintext to cloud servers may result in user privacy leakage. For plaintext data, the equality test can be met by direct comparison. Data encryption is an efficient way to safeguard user privacy. Encrypting info is an efficient way to safeguard user privacy. In real-world situations, various users encrypt files using various keys and random crypto parameters, which results in a distinct code generated from a single file. We are unable to identify file duplication by directly comparing cipher texts, and cloud servers will have to store many protected versions of the same file, placing a significant amount of storage burden on them. Therefore, in multi-user settings, it is urgently necessary to build a secure de-duplication mechanism for encrypted data with distinct keys. Convergent encryption [1] is currently frequently employed in the development of confidential information elimination of duplication systems; yet, it is not without risk, as evidenced by the following: breach of data, faking assaults, and chosen-plaintext attacks [2], [3], and [4]. Multiple files belonging to the same user will generate different encryption keys because converge encryption uses the hash value of the user's data file to generate the encryption key, which might lead to key management issues [5]. Information DE duplication and encryption are interdependent processes. The ciphertext produced by several users encrypting the same material with the same key will be identical. Direct cipher comparison allows for secure data de-duplication, but it also introduces a key management issue. It is possible to mitigate the key management issue by having separate users encrypt data with distinct keys, however achieving an equality test can be challenging. Therefore, the primary study directions of this work are how different users can encrypt data with the same key without contacting one another, producing the same cipher after encrypting the same data, and how users can retrieve

their data. This paper uses Proxy Re-encryption (PRE) for data recovery and Public Key Encryption with Keyword Search (PEKS) to match keywords with trapdoors in order to detect file duplicates [6]. The two key components of the scheme are data recovery and data de-duplication. The file cipher text, file tag, and re-encryption key must be uploaded by the data owner to the cloud server in order to perform data de-duplication. The file tag links to the file cipher text, and the re-encryption key is kept in the associated cipher. If the test returns True, the file has already been saved to the system. A owner can significantly lower a remote server's overhead for storage by simply uploading the file's re-encryption key to the appropriate crypto chain instead of the cipher text itself. If the test returns False, the data owner must upload the file tag, a cipher and re-encryption key because the file is not saved on the server. Workers do not need to save the file key for each file locally; only the user key is required for data recovery. To prevent key substitution, the user key can only be created locally, without the need for a key generation center. the KGC algorithm attacks, malevolent actions [7]. The user sends a request to the cloud server to retrieve the file, and the server re-encrypts and creates the modified text-crypto using the user's re-encryption key stored in the chain table. He receives the changed, which he can use his personal key to decipher.

2. OBJECTIVE

This study uses Proxy Re-encryption (PRE) for data security and public-key encryption with Phrase Search (PEKS) to match keywords and detect file duplication. When a file is uploaded by the owner, encryption is created using the user's data file's hash value; additional files containing the same content will not be stored.

2.1 SCOPE OF THE PROJECT

Through the use of re-encryption via proxy in conjunction with a common key accessible encryption, this project encourages us to build a safe data de-duplication and recovery system. In order to achieve secure data de-duplication in a multiple data upload and to efficiently conserve the storage capacity of the database, a secure data duplication reduction system based on global authentication with keyword search is created. Bypass re-encrypt is used in this paper to secure user data. The user simply needs to utilize his personal password to decipher and retrieve the files after the re-encryption.

3. EXISTING SYSTEM

- Subscribers may store and exchange info more easily with cloud computing, which lowers the cost of storing it.
- The dramatic decrease in internet server storage use is caused by duplicate data being stored numerous times by different people.
- The consumer's anonymity and sensitive knowledge leaks have surfaced, and there are even further CSP through the selling of user data to unique users.
- It data can eradicate imitate data directly, but cloud equipment are semi-trusted and typically require to waiting details after digital encryption to preserve user privacy.

EXISTING SYSTEM DISADVANTAGES

- Invalid Password registration.
- Unable to identify an intruder.
- It's unable to see facts, yet hackers can still assault documents.

3.1 PROPOSED SYSTEM

- The endeavor's suggested approach focuses on finding ways to recover data in cipher-text for various users and accomplish reliable eliminating duplication. It also assesses if the matched data has a unreadable to enable secure de-duplication and whether the public key encryption indexes are visible.
- The secret chain table of the stored copy of the copied file has the data the client's encryption re-en token appended to it.
- The data owner recovers the record by decrypting the altered unreadable-text using its private key, which the cloud server decrypts using the a second encryption key.

PROPOSED SYSTEM ADVANTAGES

- Enhanced enrolment.
- Higher efficiency can be achieved overall.
- The login id can only be created locally without introducing a the cloud service provider.

4. METHODOLOGIES

4.1. User Interface Design

We create the project's control panels in this module. Every person can securely log in using these windows. Users can only connect to the server by providing their login information and password in order to establish a connection. The user can log in straight to the server if they have previously left; otherwise, they need to enter their information, including their email address, login credentials, and customer. In order to maintain both download and upload rates, the server will create an account for each user. The user ID will be set to name. Typically, logging in allows access to a certain page.

4.2. Admin

The very initial part is this one. The admin module is located here. The administrator will store details after

logging in. Admin has the details of the data holder. The administrator will see all of the users' data. Management is aware of the assault.

4.3. Data Owner

This initiative's next component is this one. DO needs to log in and join for this part of the course. Content in text format can be stored by DO. Additionally, DO is able to send a key to the service-provider, who must then confirm it before sending the key to a legitimate user. Data is accessible to visitors.

4.4. CSP

The CSP has a logging in this module. There will be an inquiry from the user after access. Following registration, the information entered by the collector will request authorization from the server. Once approved by the server, the user can log in. service-Provider possesses the user's share keys. Additionally, cloud-provider can re-encrypt a file before sending it to the user. There is a version compiler and an unlock code with updates. The database keeps track of the login credentials for both data clients and its proprietors.

4.5. Data User

The information provided to the Patron can additionally signup with parameters in the aforementioned module. Login; the carbon capture and storage must provide authorization. Once approved, the sucker can log in. Users of the data are able to browse the data. A file can also be downloaded by a folder for searcher.

5.CLASSIFICATION OF ALGORITHMS MODEL

5.1 PROPOSED ALGORITHM

Advanced Encryption Standard

A homogeneous block ciphers scheme with a block/chunk size of 128 bits is the AES Encryption algorithm, sometimes referred to as the Rijndael algorithm. These discrete blocks are converted using keys consisting of 128, 192, and 256 bits. It combines these blocks to create the puzzle after encrypting them. Its foundation is an SP network, also referred to as a substitution-permutation circuit. It is made up of several interconnected processes.

AES is an essential part in protecting website server authentication from both the client and server ends when using encrypted browsing. This approach assists SSL/TLS encryption protocols to always browse with the highest level of confidentiality and secured by utilizing both the two types of cryptodata.This encryption algorithm the following advantages:

- A significantly better result is obtained when an entirely distinct key is used for each round.
- Bite replacement affects the data in a nonlinear manner, obscuring the connections between the cipher text and unencrypted.

These procedures lead to safe data interchange. For the decryption process, the identical procedure is carried out backwards.

Re-Encryption Algorithm

Using a cryptographic technique called Proxy Re-Encryption (PRE), a proxy entity can change cipher text encrypted with one key into cypher text encrypted with a different key without having to access the underlying plaintext. This method works especially well in situations where access control and data secrecy are crucial.

5.2 EXISTING SYSTEM

This paper's method just includes the user and the cloud server. The workflow is broken down into two steps: data recovery and data de-duplication. These phases comprise ten methods in all, which are explained below.

- **Config (k)** → parameters: Return a public parameter params ={q, g, e, Z, G1, G2, H1, H2, H3}, where G1 and G2 are two cyclic groups of prime order q, and g is a generator of G1. This is given a security parameter k. Z = e(g, g) and a bilinear map e: G1 × G1 → G2 are defined. Furthermore, three hash functions are defined in the following H2: {0, 1} ∗ → G1, H3: G2 → Z ∗ q, and H1: {0, 1} ∗ → Z ∗ q.
- **File Key (params, F)** → (SKF, PKF): The user executes this algorithm. After entering a file F, the user calculates the public key PKF = Z f = Z H1(F) and the private key SKF = f = H1 (F), producing a file key pair (PKF, SKF). User Key (params) -> (SKu, PKu): The user is the one running this algorithm. Once the user private key (SKu) is selected at random from the set Z ∗ q, the user computes the user public key (PKu = g a) and outputs a user key pair (PKu, SKu).

The user key pair is created independently without the assistance of KGC, and the key information is only retained by the user and not used in any way during server interactions, guaranteeing the privacy of the user key.

Test Tag(F) → TF: The user executes this algorithm. After entering the file F, test tag TF = H2(F) H1(F) is computed. To confirm that the matching file is already there on the cloud server, utilize the test tag.

5.3 SYSTEM MODEL OF ARCHITECTURE

The entities engaged in this arrangement, as depicted in Fig.1, are CSP and data users, which are further explained below. Data user: To protect their privacy, the data user encrypts the file before uploading it to the CSP. Depending on their status at the moment of upload, users are classified as either data owner (DO) or data user (DU). The DO is required to upload the file tag, cipher text, and re-encryption key, whilst the DU is only required to submit the file's re-encryption key. The

altered cipher text is produced by the CSP using the matching re-encryption key in the cipher text chain table when the data user recovers the file. To recover the file, the user might use his private key to decrypt the changed cipher text. CSP: create a file tag index, store user-uploaded file tags and the cipher text. A matching procedure is carried out to ascertain whether a test tag uploaded by the user is a duplicate file. When dealing with non-duplicate files, the user must save the cipher text, file tags, and re-encryption key on the cloud server. Users just need to save their re-encryption keys for duplicate files in the associated cipher text chain table. Upon receiving a file recovery request from the user, the CSP transforms the cipher text and delivers it to the user using the matching re-encryption key found in the cipher text chain table.

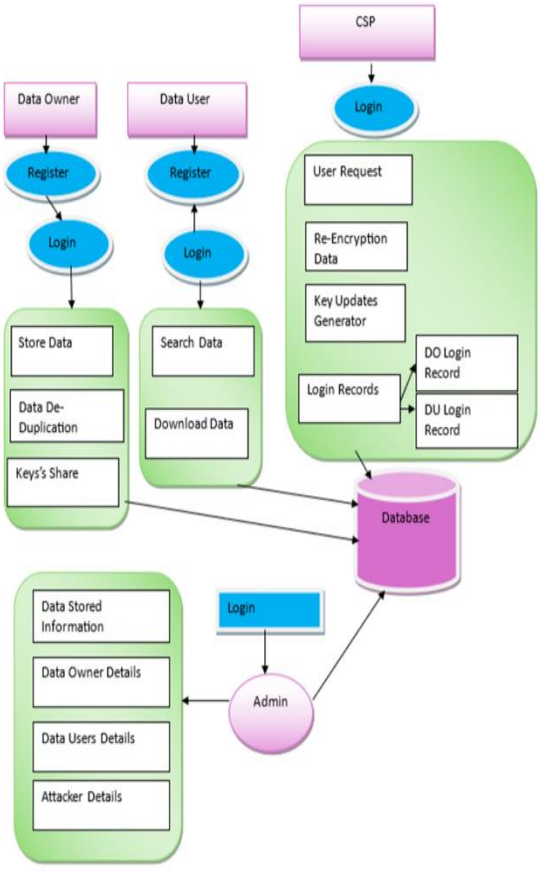


Fig 1: System Model Architecture

For this project, Alice has a data owner registration with login details. After logging in, data will be stored. After that, the data user will have to create two separate accounts. To log in, the user needs to have the required rights on the proxy server. The proxy server is operating well. The data user searched an Alice database after logging in. After searching, data will be encrypted again in a proxy server. The data user then makes a request to Alice. Alice sends a key to a proxy server. The proxy server provides the user with a key. People have the ability to download files. A user will be barred if they continuously hit the incorrect key. A user loses their ability to log in after being blocked. All of the data will be retained in this instance on the administrator's login.

6. REQUIREMENTS

- FRONT END : J2EE (JSP, SERVLET)
- BACK END : MY SQL 5.5
- OS : WINDOWS 7
- IDE : ECLIPSE

7. WORK PROCESS

Data recovery and secure data deduplication are possible with this programmer. This article primarily explains the program's workflow, which is mostly broken down into two stages: data recovery and data de-duplication. The initial stage: Deduplication of data Fig. 1 shows the data de-duplication phase's workflow. In order to create the file tag index, the data user creates file tags TagF based on the files F and uploads them to the cloud server. A test tag TF is created and provided to the CSP when a data user uploads a file. The CSP then use the matching relationship between TagF and TF to ascertain whether a file tag exists that matches the test tag the user uploaded. and provides the data consumer with the test result Test (TF, TagF). The user only needs to upload their file re-encryption key to the cipher text chain table of the corresponding file on the cloud server, if a matching file tag exists and returns true, indicating that there is already a duplicate file in the CSP. The user, acting as the data owner, must create the file cipher text, file tag, and re-encryption key for themselves and upload these to the cloud server if there isn't a matching file tag, return false, indicating that there are no duplicate files in CSP.

Stage Two: Recovery of Data Fig. 1 shows the procedure for the data recovery step. When a data user sends a request to obtain the file, the CSP asks if the user's re-encryption key, RKF→u, is present in the file F's cipher text chain table. The cloud server re-encrypts the original cipher text C of the file when the user's re-encryption key is present in the cipher text chain table of the file F**,** creates and transmits to the user the converted cipher text C ′. The user can decode the file F using private key SKu after obtaining the modified cipher text. If the user's re-encryption key is absent from the file F's cipher text chain table, the user is unable to read the file.

8. SECURITY MODEL

This paper examines a malicious CSP server that is targeted by an insider attacker. This kind of attacker can access the file tag and cypher text of any file, but they will comply with the protocol's execution. Furthermore, the attacker has access to the file's test tag and re-encryption key. The attacker is not permitted to retrieve

the test tag for the target file. It is also forbidden for an attacker to receive the target file's re-encryption key from the user if they already know the attacker's private key. The formal definition of the scheme's security model for this kind of attacker is provided below.

Configure. The challenger creates the challenger user's key pair from User Key and obtains the system parameters params = {q, g, e, Z, G1, G2, H1, H2, H3} from Setup. The attacker A receives params and PKu from the challenger.

Phase 1. A answers queries as follows.

- File key queries: Input file F, then returns the key pair (PKF , SKF) of the file F. • File key queries: Input file F, then returns the file tag TagF of the file F.
- cipher text queries: Input file F, then returns the ciphertext CF of the file F.
- Re-encryption key queries: Input file F and user public key PKu, then returns the re-encryption key RKF→u.
- Test tag queries: Input file F, then returns the test tag TF of the file F.

Based on the pertinent algorithms, the attacker can fulfil different queries for other files in addition to the challenge file F∗ in the aforementioned inquiries. Test your mettle. The challenger computes the file tag TagF∗ and cypher text CF∗,chooses a challenge file F at random from the file space G2, and sends them back to A.

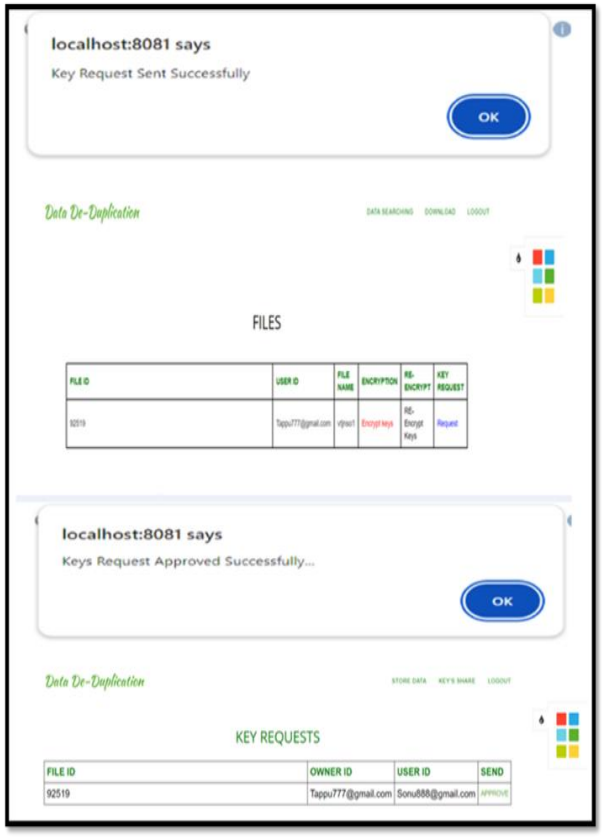
Phase 2. A can ask for queries related to any file in the same way as in phase1, but cannot ask for queries related to a challenge file F ∗ as follows: • File private key of challenge file F ∗ . • A can obtain the re-encryption key RKF∗→u from the challenge file F ∗ to the challenge user, but cannot ask for the re-encryption key from the challenge file to another user whose private key is known. • File test tag of challenge file F ∗ . Guess. A returns a file F ∈ G2. If F = F ∗ , then the attacker succeeds, otherwise the attacker fails. For any Probabilistic Polynomial-Time (PPT) attacker, a scheme is said to satisfy one-wayness under chosen file attack if the probability of the attacker succeeding in the above game is negligible.

9. RESULTS & DISCUSSION

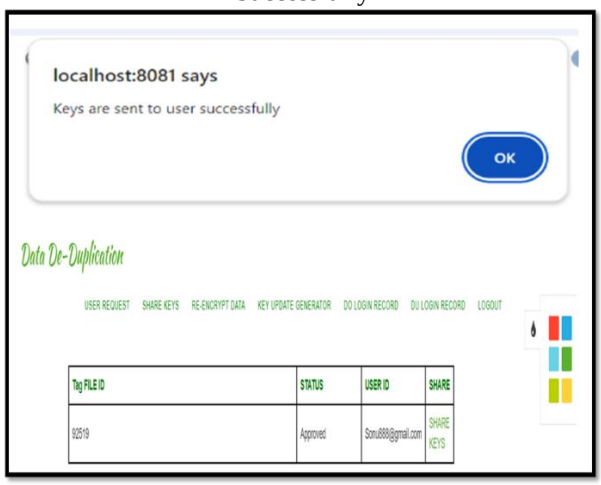
Snapshot 1 : Data Owner Registration and Login

Snapshot 2 : Uploading Data

Snapshot 3 : Searching Data

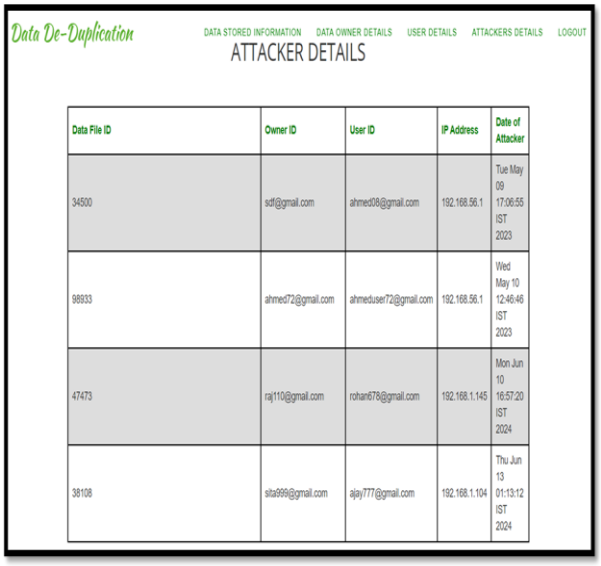


Snapshot 4 : Key Request Sent and Approved Successfully



Snapshots 5 : Approving Key Requests

Snapshots 6 : Downloading Data File



Snapshots 7 : Attacker Details

10. CONCLUSION

In the case of cloud storage, secure data de-duplication is very valuable and can help cloud storage systems use their capacity more efficiently. In this research, a secure data de-duplication and recovery technique based on PEKS is built by achieving file equality test in cipher text state using the matching relationship between keyword and trapdoor of searchable encryption and achieving data recovery using proxy re-encryption. This approach aims to minimize the computational overhead of the equality test algorithm, which is necessary to de-duplicate a single file and may require the execution of many algorithms depending on the size of the database. The findings of the experimental simulation demonstrate the good performance of the strategy presented in this work in a system that stores data in the cloud. As of right now,

researchers have made significant progress in understanding secure data de-duplication and have used it in real-world settings.

Although this project undertakes extensive research based on future work, it currently has many flaws. For example, the current paper scheme only enables equality tests at the file level. The de-duplication rate is the primary factor to be considered in the future. This article will identify two files as distinct files when the data user has two entries with only little variations, hence lowering the deletion rate.

REFERENCES

[1] J. R. Douceur, A. Adya, W. J. Bolosky, P. Simon, and M. Theimer, ‘‘Reclaiming space from duplicate files in a serverless distributed file system,’’ in Proc. 22nd Int. Conf. Distrib. Comput. Syst., Vienna, Austria, 2002, pp. 617–624.

[2] A. Agarwala, P. Singh, and P. K. Atrey, ‘‘DICE: A dual integrity convergent encryption protocol for client side secure data deduplication,’’ in Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC), Banff, AB, Canada, Oct. 2017, pp. 2176–2181.

[3] P. Anderson and L. Zhang, ‘‘Fast and secure laptop backups with encrypted deduplication,’’ in Proc. 24th Large Installation Syst. Admin. Conf., San jose, CA, USA, 2010, pp. 1–12.

[4] D. Harnik, B. Pinkas, and A. Shulman-Peleg, ‘‘Side channels in cloud services: Deduplication in cloud storage,’’ IEEE Security Privacy, vol. 8, no. 6, pp. 40–47, Nov./Dec. 2010.

[5] J. Li, X. Chen, M. Li, J. Li, P. P. C. Lee, and W. Lou, ‘‘Secure deduplication with efficient and reliable convergent key management,’’ IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 6, pp. 1615–1625, Jun. 2014.

[6] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, ‘‘Public key encryption with keyword search,’’ in Proc. Int. Conf. Theory Appl. Cryptograph. Techn., C. Cachin and J. Camenisch, Eds. Interlaken, Switzerland, 2004, pp. 506–522.

[7] M. H. Au, J. Chen, J. K. Liu, Y. Mu, D. S. Wong, and G. Yang, ‘‘Malicious KGC attacks in certificateless cryptography,’’ in Proc. 2nd ACM Symp. Inf., Comput. Commun. Secur., New York, NY, USA, 2007, pp. 302–311.

[8] M. Bellare, S. Keelveedhi, and T. Ristenpart, ‘‘Message-locked encryption and secure deduplication,’’ in Proc. 32nd Annu. Int. Conf. Theory Appl. Cryptograph. Techn., T. Johansson and P. Q. Nguyen, Eds. Athens, Greece, 2013, pp. 296–312.

[9] S. Keelveedhi, M. Bellare, and T. Ristenpart, ‘‘DupLESS: Server-aided encryption for deduplicated storage,’’ in Proc. 22th USENIX Secur. Symp., S. T. King, Ed. Washington, DC, USA, 2013, pp. 179–194.

[10] M. Abadi, D. Boneh, I. R. A. Mironov, and G. Segev, ‘‘Message-locked encryption for lock-dependent messages,’’ in Proc. 33rd Annu. Cryptol. Conf., Santa barbara, CA, USA, 2013, pp. 374–391.

[11] J. Liu, N. Asokan, and B. Pinkas, ‘‘Secure deduplication of encrypted data without additional independent servers,’’ in Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur., Denver, CO, USA, Oct. 2015, pp. 874–885.

[12] P. Puzio, R. Molva, M. Önen, and S. Loureiro, ‘‘PerfectDedup: Secure data deduplication,’’ in Proc. 10th Int. Workshop, 4th Int. Workshop, Vienna, Austria, 2015, pp. 150–166.

[13] J. Li, C. Qin, P. P. C. Lee, and J. Li, ‘‘Rekeying for encrypted deduplication storage,’’ in Proc. 46th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw., Toulouse, France, Jun. 2016, pp. 618–629.

[14] M. Li, C. Qin, J. Li, and P. P. C. Lee, ‘‘CDStore: Toward reliable, secure, and cost-efficient cloud storage via convergent dispersal,’’ IEEE Internet Comput., vol. 20, no. 3, pp. 45–53, May 2016.

[15] X. Tang, L. Zhou, W. Shan, and D. Liu, ‘‘Threshold re-encryption based secure deduplication method for cloud data with resistance against side channel attack,’’ J. Commun., vol. 41, no. 6, p. 14, 2020.

[16] W. Gao, H. Xian, and R. Cheng, ‘‘A Cloud data deduplication method based on double-layered encryption and key sharing,’’ Chin. J. Comput., vol. 44, no. 11, pp. 2203–2215, 2021.

[17] G. Kan, C. Jin, H. Zhu, Y. Xu, and N. Liu, ‘‘An identity-based proxy re-encryption for data deduplication in cloud,’’ J. Syst. Archit., vol. 121, Dec. 2021, Art. no. 102332.

[18] H. Yuan, X. Chen, J. Li, T. Jiang, J. Wang, and R. H. Deng, ‘‘Secure cloud data deduplication with efficient re-encryption,’’ IEEE Trans. Services Comput., vol. 15, no. 1, pp. 442–456, Jan. 2022.

[19] M. Blaze, G. Bleumer, and M. Strauss, ‘‘Divertible protocols and atomic proxy cryptography,’’ in Proc. Int. Conf. Theory Appl. Cryptograph. Techn., Espoo, Finland, 1998, pp. 127–144.

[20] Y. Lu and J. Li, ‘‘A pairing-free certificate-based proxy re-encryption scheme for secure data sharing in public.

[21] K. O. Agyekum, Q. Xia, E. Sifah, J. Gao, H. Xia, X. Du, and M. Guizani, ‘‘A secured proxy-based data sharing module in IoT environments using blockchain,’’ Sensors, vol. 19, no. 5, p. 1235, Mar. 2019. [22] Q. Wang, W. Li, and Z. Qin, ‘‘Proxy re-encryption in access control framework of information-centric networks,’’ IEEE Access, vol. 7, pp. 48417–48429, 2019.

[23] H. Hong and Z. Sun, ‘‘Sharing your privileges securely: A key-insulated attribute-based proxy re-encryption scheme for IoT,’’ World Wide Web, vol. 21, no. 3, pp. 595–607, 2018.

[24] X. Liu, J. Yan, S. Shan, and R. Wu, ‘‘A blockchain-assisted electronic medical records by using proxy re-

encryption and multisignature,'' Secur. Commun. Netw., vol. 2022, Feb. 2022, Art. no. 6737942.
[25] L. Fang, W. Susilo, C. Ge, and J. Wang, ''Public key encryption with keyword search secure against keyword guessing attacks without random Oracle,'' Inf. Sci., vol. 238, pp. 221–241, Jul. 2013.
[26] Y. Lu, J. Li, and Y. Zhang, ''Secure channel free certificate-based searchable encryption withstanding outside and inside keyword guessing attacks,'' IEEE Trans. Services Comput., vol. 14, no. 6, pp. 2041–2054, Nov. 2021.
[27] B. Qin, H. Cui, X. Zheng, and D. Zheng, ''Improved security model for public-key authenticated encryption with keyword search,'' in Proc. 15th Int. Conf., Q. Huang and Y. Yu, Eds. Guangzhou, China, 2021, pp. 19–38.
[28] W. Zhang, B. Qin, X. Dong, and A. Tian, ''Public-key encryption with bidirectional keyword search and its application to encrypted emails,'' Comput. Standards Interfaces, vol. 78, Oct. 2021, Art. no. 103542.
[29] B. Chen, L. Wu, S. Zeadally, and D. He, ''Dual-server public-key authenticated encryption with keyword search,'' IEEE Trans. Cloud Comput., vol. 10, no. 1, pp. 322–333, Jan. 2022.
[30] Y. Lu and J. Li, ''Lightweight public key authenticated encryption with keyword search against adaptively-chosen-targets adversaries for mobile devices,'' IEEE Trans. Mobile Comput., vol. 21, no. 12, pp. 4397–4409, Dec. 2022.

.