ECPIR: Efficient and Controllable Privacy-Preserving Image Retrieval in Cloud-Assisted System

Mingyue Li[1], Yuntao Li(✉)[1], Ruizhong Du[1], Chunfu Jia[2], and Wei Shao[3]

[1] Hebei University, Baoding 071000, China
limingyue@hbu.edu.cn, ytaolee00@gmail.com, durz@hbu.edu.cn
[2] Nankai University, Tianjin 300350, China
cfjia@nankai.edu.cn
[3] Shandong Fundamental Research Center for Computer Science, Jinan 250000, China
shaow@sdas.org

Abstract. With the rise of cloud computing, large-scale image data from personal, medical, and enterprise archives are often outsourced to cloud servers for efficient storage and computation. To ensure privacy, sensitive images must be encrypted before uploading. Cloud service providers (CSP) offer Database as a Service (DBaaS), including secure image retrieval, for managing encrypted data. Many existing schemes aim to enable privacy-preserving image retrieval but face challenges such as low retrieval efficiency, high computational costs, and limited access control.

This paper presents an Efficient and Controllable Privacy-Preserving Image Retrieval (ECPIR) scheme for scenarios using cloud-based database services. We design a hierarchical graph index to organize image vectors in multi-level formats, improving retrieval efficiency. Additionally, we propose a lightweight polynomial-based access control strategy, FastPolyAccess, which uses Fast Fourier Transform (FFT) to enhance computational efficiency and manage access for large-scale user bases. Experimental results show that ECPIR offers superior retrieval performance and robust access control while ensuring privacy.

Keywords: privacy preserving · encrypted image retrieval · access control.

1 Introduction

Cloud computing has become a crucial infrastructure for efficiently storing, processing, and analyzing large datasets in scenarios like personal photo management, collaborative medical research, and enterprise data archives. As shown in Figure 1, these systems utilize cloud servers [3, 5] to provide the necessary computational power and storage for secure and efficient data retrieval in tasks such as facial recognition, medical diagnostics, and industrial monitoring. However,

challenges related to data privacy, retrieval efficiency, and access control persist. Privacy-preserving image retrieval has been a key focus, with early works such as [10] using order-preserving encryption and Min hash. Later, [17] introduced

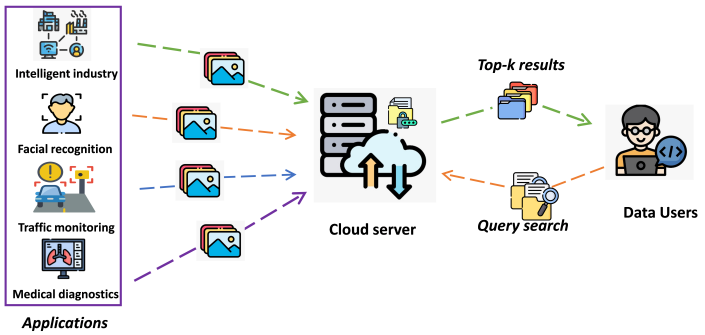


Fig. 1: Image retrieval scenarios in cloud-assisted system.

color layout and edge histogram descriptors for improved accuracy, while more recent methods [9, 15, 18] leveraged pre-trained CNNs for better performance, and [11] used fine-tuned CNNs with thumbnail-preserving encryption (TPE) to boost accuracy. For retrieval efficiency, approaches like [20] employed linear tables for indexing, but faced scalability issues, while [7] proposed tree-based indices with clustering, though their performance depended on clustering quality. [16] introduced the Twin Bloom Filter (TBF) [14], which obscured Bloom Filters but increased storage costs and exposed vulnerabilities through frequent searches. Access control mechanisms also evolved, with Ciphertext-Policy ABE (CP-ABE) used in [19], though it was limited to textual data. Fine-grained access control using accumulators was explored in [13], but faced performance and deployment challenges.

Motivated by these issues, we propose an efficient and controllable privacy-preserving image retrieval scheme for cloud-assisted systems, called ECPIR. Specifically, our contributions are as follows:

1. ***Efficient retrieval:*** We propose a hierarchical graph-based index, where each image is a node connected to its neighbors in a multi-level structure. A greedy search narrows the search space step by step, ensuring fast and accurate retrieval for large datasets.

2. ***Efficient and lightweight access control:*** *FastPolyAccess* integrates access control into secure kNN, using Fast Fourier Transform (FFT) to reduce polynomial multiplication complexity from $O(n^2)$ to $O(n \log n)$. Unauthorized users are excluded during similarity calculations, ensuring fast, secure access control for encrypted image retrieval.

1.1 FastPolyAccess

Traditional polynomial-based access control has a high computational cost with a time complexity of $O(n^2)$ during polynomial multiplication. We introduce *FastPolyAccess*, which uses Fast Fourier Transform (FFT) to reduce this complexity to $O(n \log n)$, making it more efficient for large-scale cloud-assisted environments.

In traditional polynomial access control, the polynomial is expressed as:

$$g_i(x) = N \cdot \prod_{\hat{c}_j \in \tilde{c}} (x - \hat{c}_j)^2 = N \cdot \left(\sum_{j=0}^{2\ell} c'_{i,j} x^j \right),$$

where $c'_{i,j}$ are the coefficients. The degree δ_i is less than 2ℓ, and for $\delta_i < j \leq 2\ell$, $c'_{i,j} = 0$. To determine access, we check if a user's role x_1 is a root of $g_i(x)$. If $g_i(x_1) = 0$, access is granted; otherwise, if $g_i(x_1) > N$, access is denied.

For instance, with a role set $\tilde{c} = \{3, 5, 9\}$ and image m_i, the polynomial is:

$$g_i(x) = N \cdot (x - 3)^2 (x - 5)^2.$$

For role $\{3\}$, $g_i(3) = 0$ grants access, while $g_i(9) > N$ denies it.

To optimize this, FastPolyAccess applies FFT to polynomial multiplication, following these steps:

Step 1: Polynomial Representation. Represent $g(x)$ in coefficient form.

Step 2: FFT Transformation. Transform the polynomial to the frequency domain:

$$P(\omega) = \mathrm{FFT}(P(x)).$$

Step 3: Point-wise Multiplication. Multiply the polynomials point-wise in the frequency domain:

$$G(\omega) = P_1(\omega) \cdot P_2(\omega) \cdot \cdots \cdot P_n(\omega).$$

Step 4: Inverse FFT. Apply inverse FFT to return to the time domain:

$$g(x) = \mathrm{IFFT}(G(\omega)).$$

This optimization reduces the time complexity from $O(n^2)$ to $O(n \log n)$, making FastPolyAccess highly efficient for large datasets and suitable for encrypted image retrieval with secure k-nearest neighbor (KNN) search.

1.2 FastPolyAccess based Secure k-Nearest Neighbor Method

We integrate FastPolyAccess with the Secure kNN algorithm to form a fast and lightweight access control mechanism. Specifically, the polynomial coefficients of authorized users in FastPolyAccess are encoded into the feature vectors of corresponding images. During retrieval, the cloud server evaluates the similarity scores, and only when the score exceeds a predefined threshold, the correct images are returned to the user, ensuring secure and efficient access control.

FKNN.GenKey: DO generates a key set $k_s = \{M_1, M_2, B\}$: M_1, M_2: Random reversible matrices $M_1, M_2 \in \mathbb{R}^{(d+\alpha+2\ell+2)\times(d+\alpha+2\ell+2)}$, ensuring encryption randomness and security. B: Binary vector $B \in \{0,1\}^{d+\alpha+2\ell+2}$, used to split feature vectors. The key set is securely shared with DO and DU.

FKNN.FeaEnc: For each image m_i, DO extracts the feature vector f_i, expands it:

$$f_i' = (f_i, \|f_i\|, \eta_1, \ldots, \eta_\alpha, c_{i,0}', \ldots, c_{i,2\ell}'), \tag{1}$$

where $\|f_i\|$ is the vector norm, η_j are random values, and $c_{i,j}'$ are coefficients for the polynomial $g(x)$. The vector f_i' is split into f_{ie1} and f_{ie2} using B, then encrypted:

$$f_i^* = \{M_1^\top f_{ie1}, M_2^\top f_{ie2}\}. \tag{2}$$

The encrypted dataset $\mathcal{I} = \{f_1^*, \ldots, f_n^*\}$ is uploaded to CS.

FKNN.GenTrap: DU extracts the query feature vector f_q, expands it:

$$f_q' = (-2f_{q1}, \ldots, -2f_{qd}, \theta_1, \ldots, \theta_\alpha, 1, x', \ldots, x_{2\ell}'), \tag{3}$$

where θ_j are random values and x' is the DU's role identifier. The expanded vector f_q' is split into f_{qa}' and f_{qb}' using B, then encrypted:

$$f_q^* = \{M_1^{-1} f_{qa}'^\top, M_2^{-1} f_{qb}'^\top\}. \tag{4}$$

The trapdoor $TD = \{f_q^*\}$ is sent to CS.

FKNN.Match: CS computes the similarity between f_i^* and f_q^* in the encrypted domain:

$$\hat{f}_i^\top \hat{f}_q = (f_{ie1}^\top M_1)(M_1^{-1} f_{qa}^\top) + (f_{ie2}^\top M_2)(M_2^{-1} f_{qb}^\top). \tag{5}$$

The access control polynomial $g(x')$ is incorporated into the similarity score, and the computation of $g(x')$ leverages FastPolyAccess to significantly accelerate the process:

$$\hat{f}_i^\top \hat{f}_q = \|f_i - f_q\|^2 - \|f_q\|^2 + g(x'). \tag{6}$$

If $g(x') = 0$, the DU is authorized, and the score is valid. If $g(x') \neq 0$, the score is invalid due to a penalty from $g(x')$, ensuring unauthorized users are excluded.

Finally, CS filters invalid scores, sorts the valid ones, and retrieves the top-k encrypted results R for DU.

1.3 Hierarchical graph index

As shown in Figure 2, we propose a graph-based index for efficient retrieval in large-scale encrypted image datasets. The structure builds a multi-layer graph by adding nodes sequentially, each connected to its nearest neighbors. Image feature vectors, extracted by a pretrained CNN and encrypted using the FastPolyAccess-based kNN algorithm, are assigned to the nodes, with a greedy search identifying the closest neighbors from top to bottom.

Each node $Node_i = \{MaxL_i, f_i, Neighbor_i[0 \ldots MaxL_i], EImgID_i\}$ contains: 1) $MaxL_i$: maximum level, 2) f_i: image feature, 3) $Neighbor_i$: neighboring nodes, 4) $EImgID_i$: encrypted image ID.

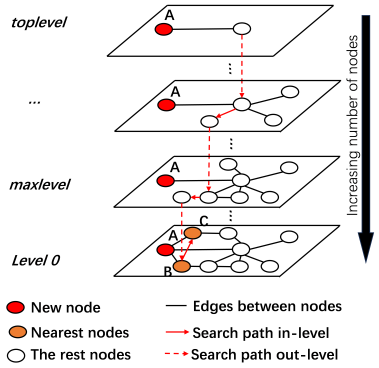


Fig. 2: Hierarchical graph index.

Figure 3 illustrates the index construction process. We mark the node to be inserted as A and set $maxLevel$ (level = 3) according to an exponential decay distribution function, where mL is the normalization factor. Node A is first inserted at $enterpoint$, and the $SearchNeighbor$ function is used from $toplevel$ to $maxlevel$ to find the closest node for each layer. Node A is then inserted in each layer by greedily searching for the M most similar nodes. Finally, at $level0$, node A retrieves the two closest nodes B and C, and is inserted near them. The specific search strategy is a variant of greedy graph search. After all nodes are inserted, the index is encrypted using the FastPolyAccess-based kNN algorithm and uploaded to the cloud server (CS).

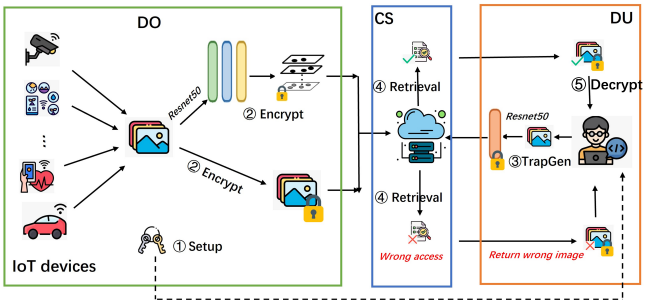


Fig. 3: System framework of ECPIR.

2 Implementation

As shown in Fig. 3, the system model involves three entities: Data Owners (DO), Data Users (DU), and Cloud Servers (CS). DOs initialize the system by gener-

ating and securely transmitting keys to registered DUs, extract feature vectors from images using a pre-trained CNN, establish encrypted index structures, and upload the encrypted database to the cloud. DUs, when querying for similar images, extract query image feature vectors, encrypt them using public keys from DOs to generate a trapdoor, and upload the trapdoor to CS, which retrieves similar images based on a similarity score and returns the encrypted result. CS stores the encrypted index and database, retrieves similar images for DUs based on the trapdoor, and returns the correct or incorrect encrypted image based on a threshold score.

Specifically, as depicted in Fig. 3, ECPIR scheme consists of five algorithms: *Setup*, *Encrypt*, *TrapGen*, *Retrieve* and *Decrypt*. Next, we provide a detailed introduction to the constructions of ECPIR.

1. Setup $(1^\lambda) \rightarrow \{k_f, k_m, pk, sk\}$: DO generates the image encryption key k_m via AES and uses the *FKNN.GenKey* algorithm to create the vector encryption key k_f:

$$\{k_m, k_f = \{\eta, M, M^{-1}\}\}$$

2. Encrypt $(M, k_m, k_f) \rightarrow \{C, \tilde{I}\}$: DO encrypts each image m_i in M using k_m, producing ciphertext c_i:

$$c_i = E(m_i; k_m)$$

Then, DO uses k_f and the $FKNN.FeaEnc$ algorithm to encrypt the index nodes, resulting in $\tilde{I}$.

3. TrapGen $(m_q, k_f) \rightarrow Q$: The DU extracts feature vectors from the query image m_q using the same CNN, then applies $FKNN.GenTrap$ to generate the query Q.

4. Retrieval $(C, Q, \tilde{I}) \rightarrow R$: CS calculates similarity using $FKNN.Match$, finds the top-k similar images, and sends the result R to DU. Unauthorized users receive inaccurate results.

5. Decrypt $(R, k_m) \rightarrow M_R$: DU decrypts the results using k_m to retrieve plaintext images. Unauthorized access is rejected.

3 security analysis

3.1 Image security

Numerous previous experiments have demonstrated the effectiveness of AES in ensuring image security; therefore, we will not elaborate on this further. As long as the key for image encryption is not leaked, the privacy of the image can be well safeguarded.

3.2 Index and Trapdoor Privacy

Theorem 1. *ECPIR can guarantee the confidentiality of indexes and trapdoors under ciphertext-known attack and background-known attack.*

Proof. To generate the index, the image's expanded feature vector v_{ie} is split using the binary vector B, and encrypted with matrices M_1 and M_2, which the

adversary cannot solve for without knowing B, M_1, or M_2. In both Known Ciphertext and Known Background Attacks, the plaintext remains secure because solving for M_1 and M_2 would require an exponentially costly process, with at least $O(n^{d+1})$ possible ciphertext combinations, making it infeasible for the adversary. □

3.3 Query Unlinkability

We prove that the queries generated in ECPIR are unlinkable by proving the following theorem.

Theorem 2. *ECPIR can realize the query unlinkability among different query requests.*

Proof. A feature vector f_q, which is extracted from the image m_q, is encrypted as

$$\tilde{f}_q = M^{-1} \cdot \left(\gamma \cdot \hat{f}_q^{\top} + \epsilon_q^{\top} \right).$$

Since ϵ_q serves as an unknown random noise vector and varies with each query, the generated query $\tilde{f}_q$ will not remain identical. Additionally, random values (i.e., $\lambda_q, \beta_1, \ldots, \beta_{d-1}$) are introduced during the extension of f_q to $\hat{f}_q$, further ensuring that queries derived from the same image are distinct. Consequently, it becomes computationally infeasible for the cloud server to ascertain whether two queries originate from the same image.

In conclusion, ECPIR can realize the query unlinkability among different query requests. □

3.4 Security of k-Nearest Neighbor Search with FastPolyAccess

FastPolyAccess is integrated with a secure k-nearest neighbor (kNN) search mechanism. In this context, the cloud server computes the similarity between the encrypted feature vectors, incorporating the access control polynomial into the similarity score computation:

$$\hat{f}_i^{\top} \hat{f}_q = \|f_i - f_q\|^2 - \|f_q\|^2 + g(x').$$

The polynomial $g(x')$ is used to penalize the similarity score for unauthorized users. If $g(x') \neq 0$, the score is invalid, and the query is rejected. This mechanism ensures that only authorized users are able to retrieve the correct k-nearest neighbors, as they are the only ones able to pass the access control check.

Access Control Integration: The integration of the access control polynomial directly into the kNN computation ensures that even if an adversary can compute the similarity score between feature vectors, they will be unable to retrieve valid results unless they pass the access control check.

Denial of Service for Unauthorized Users: If the polynomial evaluation $g(x')$ results in a non-zero value, the similarity score is invalidated, preventing unauthorized users from obtaining access to the protected resources.

Theorem 1. Let $\mathcal{A}$ be an adversary who does not know the role identifiers $\hat{c}_j \in \tilde{c}$, the polynomial coefficients $c'_{i,j}$, or the encryption matrices M_1 and M_2. The probability that $\mathcal{A}$ can successfully guess a role identifier x_1 such that $g_i(x_1) = 0$ and gain unauthorized access to the system is negligible, provided that the encryption scheme and polynomial construction are secure.

Proof. The adversary cannot compute the roots of the polynomial without knowledge of the coefficients $c'_{i,j}$ or the role set $\tilde{c}$, and the encryption scheme ensures that these values remain hidden. Additionally, even if the adversary queries the access control polynomial at different points, they cannot learn any useful information about the polynomial's roots or the encrypted data. Therefore, the probability of unauthorized access is negligible. □

4 Experiment

4.1 Experimental Setup

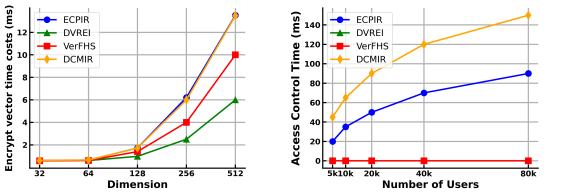
We conducted experiments using the Caltech256 [4] dataset, which consists of 256 categories, each with at least 80 images. Retrieval accuracy is measured using Precision at top-k (P@k) [1,6]. For comparison, we evaluate ECPIR against several state-of-the-art methods: DCMIR [12], which uses ResNet50 with polynomial access control, VerFHS [2], employing secure KNN for cloud-assisted IoT, and DVREI [8], which uses hierarchical index trees with secure KNN for encrypted image features.

4.2 Efficiency

Since ECPIR integrates access control and vector encryption into a lightweight framework, enhanced by a hierarchical graph indexing structure, we will evaluate the scheme's efficiency by comparing the time costs of vector encryption, access control, and retrieval. Note that all four methods use AES for image encryption, so this aspect is not included in the comparison.

1. Encryption and access control time cost: As shown in Figure 4a and 4b, DVREI and VerFHS, which lack access control, have lower encryption times than ECPIR and DCMIR, which include access control. However, ECPIR and DCMIR experience only a slight time increase (under 1ms) for 128 dimensions. While their encryption times are similar, ECPIR outperforms DCMIR in access control verification, thanks to FastPolyAccess ($O(n \log n)$) compared to traditional polynomial access control ($O(n^2)$), making ECPIR more efficient for multi-user scenarios with lightweight, high-performance access control.

2.Retrieval time cost: Figure 4c and 4d compare retrieval time and index storage costs across different schemes as data size increases. DVREI and DCMIR use tree-based index structures, achieving higher retrieval efficiency than the linear index of VerFHS. However, as data grows, tree height and internal nodes increase, leading to higher retrieval times and significant storage overhead. In contrast, VerFHS minimizes storage costs with a linear structure but suffers from the lowest retrieval efficiency due to sequential scanning.



(a) Feature encryption time (b) Access control time

(c) Retrieval time cost (d) Index construct time cost

Fig. 4: Experimental analysis

5 Conclusion and Future Work

In this paper, we design an efficient and controllable privacy-preserving image retrieval in cloud-assisted system(ECPIR) scheme that significantly accelerates image retrieval speed and supports lightweight access control. We create a hierarchical graph index structure for achieving efficient retrieval of large-scale high-dimensional vectors. Additionally, we integrated FFT technology and Secure KNN method to construct a controllable access framework. In future work, how will focus on dynamic verification systems and information concealment techniques, which will prevent authorized malicious users from disclosing privacy for profit.

References

1. N. Cao, C. Wang, M. Li, K. Ren, and W. Lou. Privacy-preserving multi-keyword ranked search over encrypted cloud data. *IEEE Transactions on parallel and distributed systems*, 25(1):222–233, 2013.
2. H. Chen, X. Lv, W. Zheng, and D. Lin. Verfhs: Verifiable image retrieval on forward privacy in blockchain-enabled iot. *IEEE Internet of Things Journal*, 10(19):17465–17478, 2023.
3. M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou. The faiss library. *arXiv preprint arXiv:2401.08281*, 2024.
4. G. Griffin, A. Holub, P. Perona, et al. Caltech-256 object category dataset. Technical report, Technical Report 7694, California Institute of Technology Pasadena, 2007.

5. Y. Jing, D. Liu, D. Kislyuk, A. Zhai, J. Xu, J. Donahue, and S. Tavel. Visual search at pinterest. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1889–1898, 2015.
6. X. Li, Q. Xue, and M. C. Chuah. Casheirs: Cloud assisted scalable hierarchical encrypted based image retrieval system. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE, 2017.
7. Y. Li, J. Ma, Y. Miao, H. Li, Q. Yan, Y. Wang, X. Liu, and K.-K. R. Choo. Dvrei: Dynamic verifiable retrieval over encrypted images. *IEEE Transactions on Computers*, 71(8):1755–1769, 2021.
8. Y. Li, J. Ma, Y. Miao, H. Li, Q. Yan, Y. Wang, X. Liu, and K.-K. R. Choo. Dvrei: Dynamic verifiable retrieval over encrypted images. *IEEE Transactions on Computers*, 71(8):1755–1769, 2022.
9. Q. Liu, X. Nie, X. Liu, T. Peng, and J. Wu. Verifiable ranked search over dynamic encrypted data in cloud computing. in 2017 ieee/acm 25th international symposium on quality of service (iwqos). 1–6, 2017.
10. W. Lu, A. Swaminathan, A. L. Varna, and M. Wu. Enabling search over encrypted multimedia databases. In *Media Forensics and Security*, volume 7254, pages 404–414. SPIE, 2009.
11. Y. Ma, X. Chai, Z. Gan, and Y. Zhang. Privacy-preserving tpe-based jpeg image retrieval in cloud-assisted internet of things. *IEEE Internet of Things Journal*, 11(3):4842–4856, 2024.
12. C. Mao, Z. Shen, K. Chen, Y. Liu, Q. Meng, and F. Wang. Dcirm: Dynamic and controllable image retrieval scheme in multi-owner multi-user settings. *IEEE Transactions on Services Computing*, 17(4):1435–1448, 2024.
13. M. S. Nair and M. Rajasree. Fine-grained search and access control in multi-user searchable encryption without shared keys. *Journal of Information Security and Applications*, 41:124–133, 2018.
14. Q. Tong, X. Li, Y. Miao, X. Liu, J. Weng, and R. H. Deng. Privacy-preserving boolean range query with temporal access control in mobile computing. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):5159–5172, 2023.
15. Q. Tong, Y. Miao, L. Chen, J. Weng, X. Liu, K.-K. R. Choo, and R. H. Deng. Vfirm: Verifiable fine-grained encrypted image retrieval in multi-owner multi-user settings. *IEEE Transactions on Services Computing*, 15(6):3606–3619, 2021.
16. Q. Tong, Y. Miao, J. Weng, X. Liu, K.-K. R. Choo, and R. H. Deng. Verifiable fuzzy multi-keyword search over encrypted data with adaptive security. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):5386–5399, 2023.
17. Z. Xia, N. N. Xiong, A. V. Vasilakos, and X. Sun. Epcbir: An efficient and privacy-preserving content-based image retrieval scheme in cloud computing. *Information Sciences*, 387:195–204, 2017.
18. J. Xu, F. Li, K. Chen, F. Zhou, J. Choi, and J. Shin. Dynamic chameleon authentication tree for verifiable data streaming in 5g networks. *IEEE Access*, 5:26448–26459, 2017.
19. 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*, pages 522–530. IEEE, 2014.
20. Y. Zhu, Z. Huang, and T. Takagi. Secure and controllable k-nn query over encrypted cloud data with key confidentiality. *Journal of Parallel and Distributed Computing*, 89:1–12, 2016.