Wiley
International Journal of Intelligent Systems
Volume 2025, Article ID 2429577, 17 pages
https://doi.org/10.1155/int/2429577

WILEY

Research Article

Machine Learning Meets Encrypted Search: The Impact and Efficiency of OMKSA in Data Security

Zhongkai Wei,[1] **Ye Su**,[2] **Xi Zhang**,[3,4] **Haining Yang**,[1] **Jing Qin**,[1] **and Jixin Ma**[5]

[1]*School of Mathematics, Shandong University, Jinan, China*
[2]*School of Information Science and Engineering, Shandong Normal University, Jinan, China*
[3]*College of Computer and Cyber Security, Hebei Normal University, Shijiazhuang, China*
[4]*Hebei Provincial Key Laboratory of Network and Information Security, Hebei Normal University, Shijiazhuang, China*
[5]*School of Computing and Mathematical Sciences, University of Greenwich, London, UK*

Correspondence should be addressed to Haining Yang; hainingyang@sdu.edu.cn and Jing Qin; qinjing@sdu.edu.cn

Received 21 December 2023; Revised 8 June 2024; Accepted 23 December 2024

Academic Editor: Vasudevan Rajamohan

The convergence of machine learning and searchable encryption enhances the ability to protect the privacy and security of data and enhances the processing power of confidential data. To enable users to efficiently perform machine learning tasks on encrypted data domains, we delve into oblivious keyword search with authorization (OKSA). The OKSA scheme effectively maintains the privacy of the user's query keywords and prevents the cloud server from inferring ciphertext information through the searching process. However, limitations arise because the traditional OKSA approach does not support multi-keyword searches. If a data file is associated with multiple keywords, each keyword and corresponding data must be encrypted one by one, resulting in inefficiency. We introduce an innovative approach aimed at enhancing the efficiency of search processes while addressing the limitation of current encryption and search systems that handle only a single keyword. This method, known as the oblivious multiple keyword search with authorization (OMKSA), is designed for more effective keyword retrieval. One of our important innovations is that it uses the arithmetic techniques of bilinear pairs to generate new tokens and new search methods to optimize communication efficiency. Moreover, we present a detailed and rigorous demonstration of the security for our proposed protocol, aligned with the predefined security model. We conducted a comparative experiment to determine which of the two schemes, OKSA and OMKSA, is more efficient when querying multiple keywords. Based on our experimental results, our OMKSA is very efficient for data searchers. As the number of query keywords increases, the computational overhead of connected keyword searches remains stable. Finally, as we move into the 5G era, the potential applications of OMKSA are huge, with clear implications for areas such as machine learning and artificial intelligence. Our findings pave the way for further exploration and deployment of these frontier areas.

Keywords: authorization; machine learning; oblivious transfer; searchable encryption

1. Introduction

In contemporary times, the convergence of machine learning (ML) [1, 2] and searchable encryption (SE) [3–5] is an important innovation. This combination will promote the enhancement of data privacy protection and the utility of big data. Known for their ability to self-learn and adapt, ML algorithms play a key role in retrieving huge datasets, recognizing patterns, and predicting outcomes. However, since these algorithms require access to large amounts of data for performance optimization, the task of maintaining data privacy and security is becoming increasingly important. To solve this problem, we can use SE technology [6], which can perform a secure search on encrypted data, ensuring data confidentiality while preserving the availability of encrypted information. The convergence of ML and SE

enhances the ability to perform efficient secure data analysis on encrypted data to protect the privacy and security of the data.

However, there is an obvious problem in public key SE. The cloud server can infer the content and search keywords of outsourced data based on the frequency of retrieval, resulting in data privacy disclosure. However, most of the existing public-key SE schemes cannot solve the inside keyword guessing attack (IKGA) problem [7–10]. Specifically, an inside adversary is an honest but curious cloud server (and cloud server operator) that honestly executes search protocols but tries to infer user keywords and private information. Once the enemy obtains the keywords, it will threaten the security of the ciphertext data file, and it also means that the search preference, personal identity, and other information of the data receiver may be leaked. Therefore, solving the problem of internal keyword guessing attack is a very meaningful and important research topic. Several encryption schemes [11–15] have been designed to solve this problem. However, their search efficiency is relatively low. This highlights the need for a cryptographic search method that is resistant to internal keyword guessing attacks.

Ogata and Kurosawa proposed the oblivious keyword search (OKS) concept [16]. The OKS scheme protects the privacy of both the database and the client. On the one hand, it is to hide the user's query database target, and on the other hand, it is to ensure that the client gets any other information except the corresponding value of the query target. At the same time, the privacy of keywords is protected, and the cloud server cannot carry out keyword guessing attacks by matching trap information and keyword information. However, in this scheme, each data file can only be associated with one keyword, and in the encryption process, each ciphertext supports only one keyword encryption search. If a file has multiple keywords, the file must be encrypted multiple times. Obviously, most files have more than one keyword.

In addition, the existing OKS protocol has proven to be inefficient when applied to large databases. We need to consider a situation where a company intends to sell a portion of its encrypted data stored on cloud servers, and potential buyers face these restrictions because they want to keep the details of the specific data content of the intended purchase private. In this case, the sender may approve the set of keywords to be retrieved for the receiver and then restrict the receiver from querying the keywords in that set. Specifically, if X represents a set of authorization keywords, the recipient can only select the keyword $w \in X$. At the same time, while the sender allows the encrypted data associated with the keyword X to be retrieved from set X, they still do not know the specific keyword used, thus protecting the privacy of the receiver. In order to solve this problem, the oblivious keyword search with authorization (OKSA) [17] solution came into being. While OKSA protects the privacy of both recipients and senders, there is an obvious problem. This model binds a single keyword to a single data file, and if a file contains multiple keywords, then each keyword and file need to be encrypted individually, which is obviously inefficient. We introduced the oblivious multiple keyword search with authorization (OMKSA) scheme, as shown in Figure 1.

1.1. Our Contributions. This paper's primary advancements are detailed in the following ways:

1. We came up with a new concept called OMKSA. It is based on the OKSA scheme. In the OKSA scheme, each encrypted data file is bound to a separate keyword. If the file contains a large number of keywords, we must encrypt each keyword and file one by one; otherwise, the retrieval will fail due to inaccurate keyword extraction. Furthermore, in the real world, where a single file often contains multiple keywords, OKSA cannot meet the need for efficiency. In the OMKSA scheme, we improved the encryption algorithm to allow the data provider to assign several keywords to each encrypted file. And the scheme can resist the internal keyword guessing attack problem.
2. We designed OMKSA protocol. Compared with OKSA protocol, our algorithm has higher efficiency when searching multiple keywords. We have carefully designed our approach to disjunctive and conjunctive keyword search, which will meet the flexible needs of users and cloud storage providers. And it adds no additional communication or storage overhead compared to the OKSA scheme searching for the same number of keywords.
3. We have compiled OMKSA scheme and analyzed the efficiency of OMKSA scheme compared with OKSA scheme through comprehensive simulation experiment. The results show that this scheme can effectively improve the efficiency of multi-keyword search. In addition, in-depth security assessments show that OMKSA meets the needs of artificial intelligence (AI) applications in the 5G era.

1.2. Related Works. Here are some pertinent studies.

1.2.1. Public Key Encryption With Keyword Search (PEKS). As far as we know, PEKS [4] has undergone significant development and improvement since they were proposed by Boneh et al. The initial design was intended to address the limitations of symmetric searchable encryption (SSE) [18–21] by developing a unique mechanism. This mechanism allows the generation of searchable ciphertext using the user's public key and specific keywords. This innovation makes it not only feasible to search on encrypted data, but it can be very efficient. This innovation began the history of SE, which later researchers introduced many improvements and tweaks aimed at enhancing security and enhancing functionality. The advent of secure channel-free PEKS (SCF-PEKS) scheme [22–25] is a big development in preventing keyword guessing attacks on cloud servers. The untrustworthiness of cloud servers can compromise search confidentiality. Functionality enhancements were not left behind.

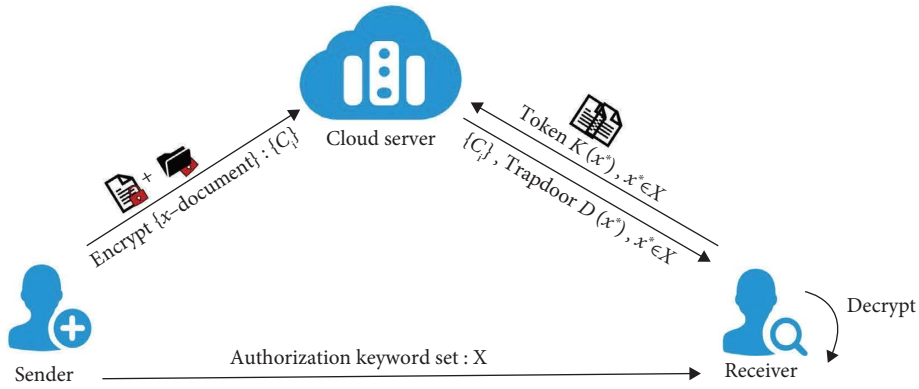

Figure 1: The framework of OMKSA.

The advancement in multi-keyword search capabilities is significantly enhanced by the integration of public-key encryption with conjunctive keyword search (PECKS) [26–30]. The introduction of public-key cryptography with temporary keyword searches (PETKS) [31] by Abdalla et al. is another important research result, as it not only significantly improves the usability of database systems but also enables the flexibility of keyword searches. The transformation protocol connects identity-based encryption (IBE) [32] to encrypted data to enable search under ciphertext, further enriching the field of cryptography. The attribute-based encryption (ABE) scheme [33–35] provides subtle control over data search and enhanced privacy protection in the cloud environment [36, 37].

1.2.2. Oblivious Transfer. The origins of inadvertent diversion (OT) [38] can be traced back to Rabin's seminal work. In the framework of the oblivious transfer protocol, the procedure is marked by an interaction between a sender, denoted as $\mathcal{S}$, and a receiver, indicated as $\mathcal{R}$. This interaction facilitates a transaction where the sender is unaware of which specific bit the receiver chooses, and the receiver is equally uninformed about the unselected bit. This approach to ensuring the privacy of both parties during transmission has led to the emergence of many OT variants [39–41], all of which contribute to enhanced security and functional diversity.

1.2.3. OKS. The design of OKS [16] is focused on ensuring rigorous privacy safeguards. It is meticulously designed to mask all informative traces, including access patterns, ensuring a user's query and retrieval activities remain concealed. The pioneering OKS protocol, elucidated by Ogata and Kurosawa, is anchored in the oblivious polynomial evaluation and the oblivious transfer methodologies. The protocol facilitates a user's engagement with a database in a manner that allows data acquisition without revealing any details about the search keywords or content to the data provider.

Ogata and Kurosawa developed two separate protocols. The first relies on the intricacies of the one-more RSA inversion problem, while the second depends on the difficulties inherent in polynomial reconstruction. Nonetheless, a constraint is manifested in these initial models, and the search is confined to a singular keyword per query. Rhee et al. advanced the narrative by unveiling an oblivious conjunctive keyword search (OCKS) protocol [42]. Rooted in the RSA known target inversion complexity, this enhancement expanded the search capacity. Simultaneously, Freedman and colleagues tackled the issue of privacy-preserving database access by proposing an efficient framework [43]. This structure links keyword search activities with the oblivious evaluation of pseudorandom functions, creating a cohesive framework. Zhu and Bao further enriched this dialog, leveraging enhanced OPEs [44] to craft an OKS protocol, and validated its security credentials. Notwithstanding these advancements, an inefficiency permeates previous OKS protocols; they are confined to a one-keyword-to-one-document architecture. This limitation amplifies both storage requisites and search durations, particularly pronounced when a document is tagged with multiple keywords, necessitating the replication of the document for each associated keyword. Jiang and Liu confronted this challenge, proposing an adept OKS protocol characterized by sublinear time search efficiency and compact ciphertexts [45]. However, it is circumscribed to disjunctive keyword search and is not configured for conjunctive keyword search applications. In this context, OKS continues to evolve, enhancing privacy while facing inherent operational constraints. Since Ogata and Kurosawa first introduced the OKS concept, new protocols have been developed to emphasize the need for coordination between strict privacy protection and operational efficiency. The exploration and resolution of these challenges are not only intrinsic to OKS' progress but also a necessity in an era when data are ubiquitous and privacy is often under attack. Future research is expected to further address this complex problem, seeking innovative solutions for data privacy, search efficiency, and functional versatility in the OKS field. Zhang et al. introduced an advanced oblivious multiple keyword search (OMKS) service [46], designed to provide users and cloud storage providers with strong privacy protection while enabling efficient multi-keyword search. Jiang et al. introduced an innovative cryptographic primitive known as OKSA [17]. Specifically, OKSA streamlines the verification of search keywords, confirming their inclusion in the user's

authorized set prior to activating the OKS protocol. This feature enables user authorization and extends the level of privacy and security of traditional OKS.

1.2.4. The Integration of ML and SE. The combination of ML and SE is an emerging field that combines the efficient data processing of ML with the privacy protection principles of SE. In the era of big data, concerns about privacy are escalating, prompting people to seek solutions that can efficiently process data while protecting privacy. ML models can extract and process valuable information from massive datasets. However, when applied to sensitive data, a privacy-security conundrum emerges. SE has emerged as a viable solution that allows searching of encrypted data, thus ensuring data privacy [47]. One of the seminal works in this domain is CryptoNets [48], which explores the use of neural networks in encrypting data to make predictions about it while protecting data privacy. This method uses homomorphic encryption technology to convert data into encrypted form and realizes the calculation in ciphertext state. However, homomorphic encryption, while revolutionary, is computationally expensive. This leads to some solutions, such as personal data collection (PATE) [49]. It uses different private learning methods, aggregating the results of multiple models to protect privacy.

Kamil Khudhair et al. optimized image steganography [50–52] to support multiple scenes, high efficiency, and high security. This inspired us to improve the OKS scheme, combining it with deep learning to further achieve high efficiency and enable multi-scenario application. Raghunandan et al. used chaotic behavior [53] to encrypt data for greater security. How to easily access the encrypted data needs to be fully considered, and our scheme can achieve searchability in the encrypted state. Asmitha et al. perfected the problems of computer vision and deep learning in human authentication [54]. This further reinforces the need for ML and SE. ML needs to invoke a lot of private data, and our solution ensures the safe use of private data. Ramesh et al. introduced false modulus in the public key encryption process to improve the security of Rabin cryptosystem [55]. This is a very interesting study and encourages us to continue to investigate how to implement the security of SE on ML.

More recent advancements have broached efficient protocols like SecureML [56], which enhances scalability and efficiency, which improves the computational challenges inherent in privacy-protecting ML. Furthermore, innovations like oblivious neural network predictions (MiniONN) [57] have facilitated privacy-preserving by transforming existing neural network models into oblivious counterparts. With the convergence of ML and SE, the focus is shifting to ensuring data privacy without compromising the quality of ML predictions. The future trajectory may be to explore enhanced computing efficiency, robust security protocols, and common applications that span different data domains.

1.3. Organization. Section 2 presents essential foundational concepts, details the OMKSA algorithm, describes the security model, and discusses the underlying complexity assumptions. Section 3 introduces an efficient OMKSA protocol, while Section 4 offers a security proof. Section 5 focuses on conducting a thorough analysis and assessing the performance of the system. Additionally, Section 6 delves into applications in the field of ML. The paper concludes in Section 7.

2. Preliminaries

2.1. Notations. The symbols and terms utilized in this paper are efficiently compiled in Table 1.

2.2. Bilinear Pairing. $\mathbb{G}$, $\mathbb{G}_T$ are two multiplicative cyclic groups of order p, any generator of $\mathbb{G}$ is g, and bilinear mapping e: $\mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$ satisfies the following properties [58]:

1. Bilinearity: For the generator g of $\mathbb{G}$, the other generator h of $\mathbb{G}$, and the positive integers a, b, we can get the equation: $e(g^a, h^b) = e(g, h)^{ab}$.
2. Computability: For any element g, $h \in \mathbb{G}$, we can efficiently compute e, which is a probabilistic polynomial algorithm.
3. Non-degeneracy: There are two elements in $\mathbb{G}$, g and h, that satisfy the equation: $e(g, h) \neq 1$. 1 is the identity element of $\mathbb{G}_T$. 1 is the identity element of $\mathbb{G}_T$.

e is a symmetric bilinear pair mapping, and the corresponding symmetric double planet pair system is denoted by $\mathcal{PG} = (p, \mathbb{G}, \mathbb{G}_T, e)$.

2.3. Algorithm Definition. Our OMKSA solution interacts with three parties: the data sender, the cloud server, and the data receiver.

2.3.1. Setup. Commencing the procedure, the entity $\mathcal{S}$ provides a security parameter denoted by 1^λ and an integer value n, subsequently generating the master key pair (MPK, MSK) for cryptographic system configuration. Furthermore, in collaboration with each user, $\mathcal{S}$ establishes a distinct keyword set X, ensuring that the size of X does not exceed n^2.

2.3.2. Commit. $\mathcal{S}$ processes a message m_i, a designated keyword x_i, and the MPK to produce the corresponding encrypted data C_i. Notably, each message m_i is coupled with a distinct keyword array $\{x_{i_1}, x_{i_2}, \ldots, x_{i_n}\}$. Subsequently, $\mathcal{S}$ securely transmits the collection of ciphertexts $\{C_i\}$ to the cloud storage $\mathcal{C}$.

2.3.3. Transfer. It is assumed that $\mathcal{S}$ establishes a distinct keyword set X in negotiations with each receiver $\mathcal{R}$, where $X \subseteq \mathcal{KS}$.

$\mathcal{R} \longrightarrow \mathcal{C}$: In this step, $\mathcal{R}$ inputs the designated keyword set X, including specific keywords $\{x_{i_1}', x_{i_2}', \ldots, x_{i_n}'\}$ within X and MPK. This leads to the

Table 1: Notations.

Notation	Description
$\mathcal{S}$	A data sender
$\mathcal{C}$	The cloud server
$\mathcal{R}$	The receiver
X	Authorization keyword set
x^*	Search keyword
α	The master secret key
$\mathbf{K}(x^*)$	Token of *
$\mathbf{D}(x^*)$	Trapdoor cryptosystem of *
L	A collection of search results
ℓ	The actual number of keywords retrieved
D	The maximum number of keywords that can be retrieved simultaneously

creation of the token $\mathbf{K}(x_{i_1}^{'}, x_{i_2}^{'}, \ldots, x_{i_n}^{'})$ of keywords, along with the user's secret key sk and Γ which is proof of accountability. Then, the token and proof, represented as $(\mathbf{K}(x_{i_1}^{'}, x_{i_2}^{'}, \ldots, x_{i_n}^{'}), \Gamma)$, are sent to $\mathcal{C}$. The token $\mathbf{K}(x_{i_1}^{'}, x_{i_2}^{'}, \ldots, x_{i_n}^{'})$ is derived from sk, $(x_{i_1}^{'}, x_{i_2}^{'}, \ldots, x_{i_n}^{'}), X$, and MPK, with Γ aiding $\mathcal{C}$ in verifying the token's accountability, specifically ensuring it generates a trapdoor for a valid keyword in the set X.

$\mathcal{C} \longrightarrow \mathcal{R}$: $\mathcal{C}$ inputs the token $\mathbf{K}(x_{i_1}^{'}, x_{i_2}^{'}, \ldots, x_{i_n}^{'}), X$, and the MSK to affirm accountability, verifying that $|\mathbf{K}(x_{i_1}^{'}, x_{i_2}^{'}, \ldots, x_{i_n}^{'})| = 1$. It then generates a trapdoor D for delivery to $\mathcal{R}$.

$\mathcal{R}$: On the condition that the sequence $(x_{i_1}, x_{i_2}, \ldots, x_{i_n})$ aligns with $(x_{i_1}^{'}, x_{i_2}^{'}, \ldots, x_{i_n}^{'}), \mathcal{R}$, upon receiving inputs C_i, D, sk, will yield m_i; otherwise, it results in $\perp$.

2.3.4. Correctness. For an OMKSA to be deemed effective, it is essential that the recipient accurately acquires the specified message, provided all involved parties comply with the outlined protocol. Furthermore, the authenticity is validated when accountability checks confirm that the trapdoor created from the acquired token corresponds exclusively to specific keywords. It is also crucial that these keywords fall within the scope of the authorized keyword collection.

2.4. Security Notions. Based on the previous article [4, 16], our framework establishes three key security criteria: confidentiality for the receiver, indistinguishability, and accountability. The first aspect, user confidentiality, ensures that during the i – th transaction phase, the sender, denoted as $\mathcal{S}$, is unable to infer the search keywords from the user's token. The principle of indistinguishability acts as a protective measure, preventing a potentially adversarial receiver, labeled $\mathcal{R}$, from deciphering both the message and keyword embedded within the ciphertext. The accountability aspect primarily ensures that each trapdoor request by a user is distinctly linked to a specific keyword in the authorized keyword array.

Receiver privacy

For a receiver, discerning whether x is x_0 or x_1 becomes challenging when presented with $(\mathbf{K}(x), \Gamma)$ and the pair of keywords x_0, x_1.

Indistinguishability

When faced with a ciphertext C for (m, x) and two distinct message-keyword pairs (m_0, x_0) and (m_1, x_1), the receiver finds it difficult to determine if (m, x) matches (m_0, x_0) or (m_1, x_1).

Accountability

For scenarios where $(\mathbf{K}(X), X, sk)$ meets the condition $|X| > 1$, crafting $(\mathbf{K}(X), \Gamma)$ that successfully undergoes verification is a complex task.

In light of the specified requirements, our formulation of the security models is through a series of interactive games involving a challenger, denoted as $\mathcal{C}$, and an opponent, referred to as $\mathcal{A}$.

2.4.1. User Privacy

Setup.

$\mathcal{C}$ executes the **Setup** procedure, thereby generating the system parameter MPK which is then transmitted to the adversary $\mathcal{A}$.

Challenge.

The adversary $\mathcal{A}$ proposes two distinct keywords x_0, x_1 to the challenger $\mathcal{C}$. In response, $\mathcal{C}$ selects a random $\theta \in 0, 1$, sets x to x_θ, and produces $(\mathbf{K}(X), \Gamma)$.

Guess.

$\mathcal{A}$ attempts to guess by announcing θ' and succeeds if θ' matches θ.

The advantage of $\mathcal{A}$ is quantified as $\text{Adv} = |\Pr[\theta' = \theta] - 0.5|$.

Definition 1. The OMKSA framework is deemed to adhere to user privacy standards when it is infeasible for any adversary operating in probabilistic polynomial-time to secure a substantial edge in the stipulated user privacy contest.

2.4.2. Indistinguishability

Setup.

The challenger $\mathscr{C}$ executes the **Setup** protocol to produce the system parameter MPK and forwards it to adversary $\mathscr{A}$.

First Stage.

In this stage, $\mathscr{A}$ requests a trapdoor for keyword X, and $\mathscr{C}$ complies by providing trapdoor D.

Challenge.

$\mathscr{A}$ presents two equally long message-keyword pairs (m_0, x_0) and (m_1, x_1) to $\mathscr{C}$, ensuring that x_0, x_1 were not previously queried for trapdoors in the **First Stage**. $\mathscr{C}$ then issues a challenge ciphertext C^*, selecting θ randomly from 0,1.

Second Stage.

$\mathscr{A}$ continues to make trapdoor requests under the same constraints as the **Challenge**, with $\mathscr{C}$ providing responses similarly to the **First Stage**.

Guess.

$\mathscr{A}$ proposes a guess of θ' and is victorious if it matches θ.

The advantage of $\mathscr{A}$ is defined as $\text{Adv} = |\Pr[\theta' = \theta] - 0.5|$.

Definition 2. OMKSA achieves indistinguishability against chosen keyword attacks if no adversary, employing probabilistic polynomial time approaches, can significantly succeed in the described game.

2.4.3. Accountability. In OMKSA, the essence of accountability verification lies in confirming that each trapdoor produced is exclusively linked to a single permitted keyword. This process counteracts scenarios where an adversary, denoted as $\mathscr{A}$, might construct a legitimate keyword token $\mathbf{K}(X')$, with X' being a fraction of the endorsed keyword set X, maintaining that $1 < |X'| < |X| \leq n$, indicating $\mathscr{A}$'s awareness of X', X, and the secret key sk used in the token's formation.

Setup.

Executing the **Setup** phase, the challenger $\mathscr{C}$ creates the system's parameter MPK and subsequently dispatches this parameter to the adversary $\mathscr{A}$.

Challenge.

Here, $\mathscr{A}$ presents $(\mathbf{K}(X'), X, X', \text{sk})$ alongside a numeral 1 for the challenge, deriving $\mathbf{K}(X')$ from $X, X', \text{sk}, \text{MPK}$ with $|X'|$ exceeding 1.

Win.

The adversary $\mathscr{A}$ submits $(\mathbf{K}(X'), \Gamma)$ and is deemed victorious if this submission successfully clears the verification process.

The edge of $\mathscr{A}$ in this scenario is determined as Adv in the formulation of $(\mathbf{K}(X'), \Gamma)$.

Definition 3. Accountability within an OMKSA framework is established when no adversary, operating within polynomial time bounds, can achieve success in the previously outlined game with a notable margin.

2.5. Assumptions. The foundational security constructs of OKSA are built upon two complex challenges: the (f, n)-DHE Problem and the (f, q)-MSE-DDH Problem. While the former has been introduced in earlier works [59], here we provide only a brief overview, directing readers to these references for a thorough exploration of its complexities.

Definition 4. The challenge in the (f, n)-DHE Problem within a group $\mathbb{G}$ of prime order p, involving elements $h \in \mathbb{G}$ and $a \in \mathbb{Z}_p$, is to compute $(f(x), h^{f(a)})$ from the given sequence $h, h^a, \ldots, h^{a^n}$. In this context, $f(x)$ represents a polynomial within $\mathbb{Z}_p[x]$ whose degree is greater than n.

The (f, q)-MSE-DDH Problem is a refined variant of the MSE-DDH Problem, preserving the original problem's complexity. This problem stands as a particular case within the broader set of Diffie–Hellman exponent assumptions, as discussed in the previous article [60]. Detailed intractability analysis of this problem will be presented later. For more details, please refer to the paper [59].

3. OMKSA Protocol

In this segment, we introduce a protocol designed for OMKSA. This protocol seamlessly accommodates both disjunctive and conjunctive keyword searches, thanks to the integration of innovative token architectures and the implementation of sophisticated search algorithms as delineated in [59]. Moreover, our protocol endows the receiver with the capability to unobtrusively secure an authorized trapdoor by adaptively presenting a keyword token. A hallmark of this system is its efficiency, characterized by the constant size of communication overhead incurred between $\mathscr{S}$ and $\mathscr{R}$. $\mathscr{S}$ is enabled to construct a trapdoor nested within X without the capacity to ascertain the specific keyword in question. The OMKSA protocol engages three entities: a data sender $\mathscr{S}$, a cloud server $\mathscr{C}$, and a receiver $\mathscr{R}$, unfolding over three distinct phases: **Setup, Commit, and Transfer**.

3.1. Disjunctive Keyword Search. The scheme facilitates the execution of disjunctive keyword searches by innovatively creating tokens and applying sophisticated search techniques. Imagine a scenario where the data receiver, denoted as $\mathscr{R}$, aims to locate documents that include either keyword x_1 OR x_2. In such cases, the OMKSA scheme is implemented as follows.

3.1.1. Setup. Entity $\mathscr{S}$ begins by accepting a security parameter denoted as 1^λ, an integer value n, and the group system $\mathscr{PG} = (p, \mathbb{G}, \mathbb{G}_T, e)$. Following this, $\mathscr{S}$ selects generator elements g, h from $\mathbb{G}$, and chooses random numbers α, x from $\mathbb{Z}_p$. It proceeds to calculate g^α and $h_i = h^{\alpha^{i-1}}$ for

each i ranging from 1 to $n+1$ for $i = 1, 2, \ldots, n+1$. A individual hash function H mapping from $(\{0,1\}, \mathbb{G}_D)$ to a binary string of length ℓ is also selected. The ensuing steps involve the generation of the master key pair:

$$\text{MPK} = \left(\mathcal{PG}, H, g, g^{\alpha}, h_1, h_2, \ldots, h_{n+1}\right), \quad \text{MSK} = \alpha. \tag{1}$$

Entity $\mathcal{S}$ makes the MPK available to everyone and maintains the confidentiality of K.

3.1.2. Commit. Within the framework, the complete keyword space is represented by $\mathcal{KS}$, encompassing a total of n^2 elements. Each message is paired with corresponding keywords. For a given message m_i within the binary set $\{0,1\}^{\ell}$ and associated keywords x_{i_j} from $\mathcal{KS}$, the entity $\mathcal{S}$ selects a random value r_i from ${}_R\mathbb{Z}_p$, It then calculates β as the product of $\prod_{j=1}^{n}(\alpha + x_{i_j})$ and forms the encrypted message C_i in the following manner:

$$C_i = \left(c_{1i} = g^{r_i\beta}, c_{2i} = H\left(0, e(g,h)^{r_i}\right), c_{3i} = H\left(1, e(g,h)^{r_i}\right) \oplus m_i\right). \tag{2}$$

Subsequently, $\mathcal{S}$ securely transmits the collection of ciphertexts $\{C_i\}$ to the receiver $\mathcal{R}$.

3.1.3. Transfer. It is assumed that $\mathcal{S}$ establishes a distinct keyword set X for each receiver, with $X \subseteq \mathcal{KS}$ and the size of X is denoted as $|X| = k$ not exceeding n^2.

$\mathcal{R} \longrightarrow \mathcal{C}$: With the predefined keyword set X, and specific keywords $\left\{x_{i_1}, x_{i_2} \ldots x_{i_n}\right\} \in X$ alongside the MPK, the receiver $\mathcal{R}$ chooses a random number s from ${}_R\mathbb{Z}_p$ and let $sk = s$. Then, $\mathbf{K}_1(x_{i_1}), \mathbf{K}_2(x_{i_2})$, formulate the tokens, where x_{i_1} and x_{i_2} correspond to the chosen keywords to be searched by $\mathcal{R}$.

$$\begin{aligned} \gamma_1 &= \left(\alpha + x_{i_1}\right), \\ \gamma_2 &= \left(\alpha + x_{i_2}\right), \\ \tau_1 &= \prod_{x_{i_j} \in X, j \neq 1} \left(\alpha + x_{i_j}\right), \tau_2 = \prod_{x_{i_j} \in X, j \neq 2} \left(\alpha + x_{i_j}\right), \end{aligned} \tag{3}$$

and the proof Γ is

$$\begin{aligned} \mathbf{K}_1\left(x_{i_1}\right) &= h^{s\tau_1}, \\ \mathbf{K}_2\left(x_{i_2}\right) &= h^{s\tau_2}, \\ \Gamma_1^{(1)} &= h^{(\gamma_1/s)}, \\ \Gamma_2^{(1)} &= \Gamma_1^{(1)^{\alpha^{n-1}}}, \\ \Gamma^{(1)} &= \left(\Gamma_1{}^{(1)}, \Gamma_2^{(1)}\right), \\ \Gamma_1^{(2)} &= h^{(\gamma_2/s)}, \\ \Gamma_2^{(2)} &= \Gamma_1^{(2)^{\alpha^{n-1}}}, \\ \Gamma^{(2)} &= \left(\Gamma_1^{(2)}, \Gamma_2^{(2)}\right). \end{aligned} \tag{4}$$

Then $\mathcal{R}$ sends $(\mathbf{K}_1(x_{i_1}), \Gamma^1)$ and $(\mathbf{K}_2(x_{i_2}), \Gamma^2)$ to $\mathcal{C}$.

$\mathcal{C} \longrightarrow \mathcal{R}$: Given the tuple $(\mathbf{K}_1(x_{i_1}), X, \Gamma^1)$, $(\mathbf{K}_2(x_{i_2}), X, \Gamma^2)$, with the master public key MPK in hand, $\mathcal{C}$ proceeds to verify accountability using the subsequent equations,

$$\begin{aligned} e\left(\Gamma_2^{(1)}, h^{\alpha}\right) &= e\left(\Gamma_1^{(1)}, h^{\alpha^n}\right), \\ e\left(h, h^{\prod_{x_{i_j} \in X}\left(\alpha + x_{i_j}\right)}\right) &= e\left(\mathbf{K}_1\left(x_{i_1}\right), \Gamma_1^{(1)}\right), \end{aligned} \tag{5}$$

and

$$\begin{aligned} e\left(\Gamma_2^{(2)}, h^{\alpha}\right) &= e\left(\Gamma_1^{(2)}, h^{\alpha^n}\right), \\ e\left(h, h^{\prod_{x_{i_j} \in X}\left(\alpha + x_{i_j}\right)}\right) &= e\left(\mathbf{K}_2\left(x_{i_1}\right), \Gamma_1^{(2)}\right). \end{aligned} \tag{6}$$

When both equations are satisfied, it confirms that the tokens $\mathbf{K}_1, \mathbf{K}_2$ are associated with the authorized set of keywords, indicated as

$$\begin{aligned} \left|\mathbf{K}_1\left(x_{i_1}\right)\right| &= 1, \\ \left|\mathbf{K}_2\left(x_{i_2}\right)\right| &= 1. \end{aligned} \tag{7}$$

In cases where this condition is not met, the algorithm terminates. Upon having the MSK and the keyword set X, $\mathcal{C}$ then proceeds to generate the trapdoors D_1, D_2 as

$$\begin{aligned} D_1 &= \mathbf{K}_1\left(x_{i_1}\right)^{(1/\beta)}, \\ D_2 &= \mathbf{K}_2\left(x_{i_2}\right)^{(1/\beta)}. \end{aligned} \tag{8}$$

Then $\mathcal{C}$ returns the trapdoor D_1, D_2 to $\mathcal{R}$.

$\mathcal{R}$: Upon receiving C_i, D_1, D_2, sk, the entity $\mathcal{R}$ initiates the process of searching by

$$\begin{aligned} c_{2i}^{(1)} &= H\left(0, e\left(c_{1i}, D_1\right)^{(1/s\tau_1)}\right), \\ c_{2i}^{(2)} &= H\left(0, e\left(c_{1_i}, D_2\right)^{(1/s\tau_2)}\right). \end{aligned} \tag{9}$$

Should the preceding equation be satisfied, the receiver $\mathcal{R}$ then proceeds with the decryption process by

$$\begin{aligned} m_i^{(1)} &= c_{3i} \oplus H\left(1, e\left(c_{1i}, D_1\right)^{(1/s\tau_1)}\right), \\ m_i^{(2)} &= c_{3i} \oplus H\left(1, e\left(c_{1i}, D_2\right)^{(1/s\tau_2)}\right). \end{aligned} \tag{10}$$

$\mathcal{R}$ adds $(m_i^{(1)}, x_{i_1})$ to a list $L^{(1)}$ and adds $(m_i^{(2)}, x_{i_2})$ to a list $L^{(2)}$, and $\mathcal{R}$ has $L = L^{(1)} \cup L^{(2)}$ as final calculation.

3.1.4. Extension to D Keywords. The outlined OMKSA protocol, initially configured for disjunctive keyword search, is readily expandable to encompass D keywords. The data receiver $\mathcal{R}$ computes $P^{(\ell)}$ for each ℓ where $1 \leq \ell \leq t$, subsequently recovering $T^{(\ell)}$ and compiling the list $L^{(\ell)}$ for every keyword. The final step entails $\mathcal{R}$ aggregating the lists corresponding to all D keywords to yield the comprehensive disjunctive search outcome.

3.1.5. Correctness. With the (MPK, MSK) obtained from the **Setup**, and $(\mathbf{K_1}(x_{i_1}), \Gamma^{(1)})$ at hand, the verification of accountability's correctness hinges on the fulfillment of the following expression.

$$\begin{aligned}
e\left(\Gamma_2^{(1)}, h^{\alpha}\right) &= e\left(\Gamma_1^{(1)^{\alpha^{n-1}}}, h^{\alpha}\right) \\
&= e\left(\Gamma_1^{(1)}, h^{\alpha^n}\right), \\
e\left(h, h^{\prod_{x_{i_j}\in X}\left(\alpha+x_{i_j}\right)}\right) &= e\left(h, h^{\tau_1\gamma_1}\right) \\
&= e\left(h^{s\tau_1}, h^{(\gamma_1/s)}\right) \\
&= e\left(\mathbf{K}_1\left(x_{i_1}\right), \Gamma_1^{(1)}\right), \\
e\left(\Gamma_2^{(2)}, h^{\alpha}\right) &= e\left(\Gamma_1^{(2)^{\alpha^{n-1}}}, h^{\alpha}\right) \\
&= e\left(\Gamma_1^{(2)}, h^{\alpha^n}\right), \\
e\left(h, h^{\prod_{x_{i_j}\in X}\left(\alpha+x_{i_j}\right)}\right) &= e\left(h, h^{\tau_2\gamma_2}\right) \\
&= e\left(h^{s\tau_2}, h^{(\gamma_2/s)}\right) \\
&= e\left(\mathbf{K}_2\left(x_{i_2}\right), \Gamma_1^{(2)}\right).
\end{aligned} \tag{11}$$

With the ciphertext C_i obtained through the execution of the **Commit** algorithm, coupled with the trapdoors D_1 and D_2 generated via **Transfer**, and the user's sk, the process to validate the accuracy of the scheme's function involves

$$\begin{aligned}
H\left(0, e\left(c_{1i}, D_1\right)^{(1/s\tau_1)}\right) &= H\left(0, e\left(g^{r_i\beta}, h^{(s\tau_1/\beta)}\right)^{(1/s\tau_1)}\right) \\
&= H\left(0, e(g,h)^{r_i}\right) \\
&= c_{2i}^{(1)}, \\
c_{3i} \oplus H\left(1, e\left(c_{1i}, D_1\right)^{(1/s\tau_1)}\right) &= H\left(1, e(g,h)^{r_i}\right) \oplus m_i \oplus H\left(1, e\left(g^{r_i\beta}, h^{(s\tau_1/\beta)}\right)^{(1/s\tau_1)}\right) \\
&= H\left(1, e(g,h)^{r_i}\right) \oplus m_i \oplus H\left(1, e(g,h)^{r_i}\right) \\
&= m_i^{(1)}, \\
H\left(0, e\left(c_{1i}, D_2\right)^{(1/s\tau_2)}\right) &= H\left(0, e\left(g^{r_i\beta}, h^{(s\tau_2/\beta)}\right)^{(1/s\tau_2)}\right) \\
&= H\left(0, e(g,h)^{r_i}\right) \\
&= c_{2i}^{(2)}, \\
c_{3i} \oplus H\left(1, e\left(c_{1i}, D_2\right)^{(1/s\tau_2)}\right) &= H\left(1, e(g,h)^{r_i}\right) \oplus m_i \oplus H\left(1, e\left(g^{r_i\beta}, h^{(s\tau_2/\beta)}\right)^{(1/s\tau_2)}\right) \\
&= H\left(1, e(g,h)^{r_i}\right) \oplus m_i \oplus H\left(1, e(g,h)^{r_i}\right) \\
&= m_i^{(2)}.
\end{aligned} \tag{12}$$

3.2. Conjunctive Keyword Search. Imagine a scenario where the data receiver $\mathscr{R}$ is inclined to retrieve documents that encompass both keywords x_1 AND x_2. An initial instinct might be to adapt the protocol aligned with the earlier elaborated disjunctive keyword search; essentially, $\mathscr{R}$ would amass the list $T^{(1)}$ correlated with keyword x_1 and $T^{(2)}$ associated with x_2, culminating in the intersection $T = T^{(1)} \cap T^{(2)}$ to finalize the conjunctive search outcome. However, this approach is riddled with the pitfall of information leakage, as $\mathscr{R}$ inadvertently gains access to documents extraneous to the intersection $T^{(1)} \cap T^{(2)}$. In simple terms, if a file c_1 contains the keyword x_1, but does not contain the keyword x_2, then it belongs to set T_1, and if a file c_2 does not contain the keyword x_1, but contains the keyword x_2, it belongs to set T_2, and neither file should be searched by the data receiver $\mathscr{R}$. His searching scope includes these files which must contain two keywords x_1 and x_2. Therefore, the resulting set of files for the data recipient is required to be the intersection of T_1 and T_2. Otherwise, the search scope of the data receiver $\mathscr{R}$ will be expanded, causing the pitfall of information leakage.

To navigate around this challenge, an enhancement of the protocol is requisite, one that curtails $\mathscr{R}$ from accessing data that are not explicitly within the search range. The refinement of the protocol unfolds as delineated in the succeeding text.

3.2.1. Setup. Entity $\mathscr{S}$, takes as input a security parameter 1^λ, an integer n and the group tuple $\mathscr{PG} = (p, \mathbb{G}, \mathbb{G}_T, e)$ as Section 3.1.1. Then $\mathscr{S}$ selects generator elements g and h from $\mathbb{G}$, along with random values α, x from the set $\mathbb{Z}_p$. It then calculates $g^\alpha, h_i = h^{\alpha^{i-1}}$ for $i = 1, 2, \ldots, n+1$. H: $(\{0,1\}, \mathbb{G}_D) \longrightarrow \{0,1\}^\ell$ is also employed. The construction of the master key pair follows these steps.

$$\text{MPK} = \left(\mathscr{PG}, H, g, g^\alpha, h_1, h_2, \ldots, h_{n+1}\right), \quad \text{MSK} = \alpha. \tag{13}$$

Entity $\mathscr{S}$ makes the *MPK* publicly available while ensuring the confidentiality of MSK.

3.2.2. Commit. In the defined keyword space $\mathscr{KS}$, which encompasses n elements, each message is linked to specific keywords. For any given message m_i within the binary space $\{0,1\}^\ell$ and its associated set of keywords x_{i_j} from $\mathscr{KS}$, the system $\mathscr{S}$ selects a random r_i from ${}_R\mathbb{Z}_p$, and calculates $\beta = \prod_{j=1}^{n} (\alpha + w_{ij})$. C_i is

$$C_i = \left(c_{1i} = g^{r_i\beta}, c_{2i} = H\left(0, e\left(g,h\right)^{r_i}\right), c_{3i} = H\left(1, e\left(g,h\right)^{r_i}\right) \oplus m_i\right). \tag{14}$$

$\mathscr{S}$ securely transfers the set of ciphertexts $\{C_i\}$ to the recipient $\mathscr{C}$.

3.2.3. Transfer. It is assumed that $\mathscr{S}$ establishes a distinct set of keywords, X, for each receiver. This set X, a subset of $\mathscr{KS}$, is characterized by having a maximum size denoted as X, where $|X| = k$ and does not exceed n^2.

$\mathscr{R} \longrightarrow \mathscr{C}$: In the context of the authorized keyword collection X, encompassing specific keywords $\left\{x_{i_1}, x_{i_2}, \ldots x_{i_t}\right\}$ within X, and with access to MPK, the participant $\mathscr{R}$ selects a random s from ${}_R\mathbb{Z}_p$ to serve as $sk = s$ and proceeds to calculate the corresponding token $\mathbf{K}(x_{i_1}, x_{i_2}, \ldots x_{i_t})$,

$$\begin{aligned} \gamma &= \left(\alpha + x_{i_1}\right)\left(\alpha + x_{i_2}\right)(, \ldots,)\left(\alpha + x_{i_t}\right), \\ \tau &= \prod_{x_{i_j} \in X, j \notin \{1,2,\ldots,t\}} \left(\alpha + x_{i_t}\right), \end{aligned} \tag{15}$$

and the proof Γ as

$$\begin{aligned} \mathbf{K}\left(x_{i_1}, x_{i_2}, \ldots x_{i_t}\right) &= h^{s\prod_{x_{i_j} \in X, j \notin \{1,2,\ldots,t\}} \left(\alpha + x_{i_j}\right)}, \\ \Gamma_1 &= h^{(\gamma/s)}, \Gamma_2 = \Gamma_1^{\alpha^{n-1}}, \\ \Gamma &= (\Gamma_1, \Gamma_2). \end{aligned} \tag{16}$$

Then $\mathscr{R}$ sends $(\mathbf{K}(x_{i_1}, x_{i_2}, \ldots x_{i_t}), \Gamma)$ to $\mathscr{C}$.

$\mathscr{C} \longrightarrow \mathscr{R}$: With the set $(\mathbf{K}(x_{i_1}, x_{i_2}, \ldots x_{i_t}), X, \Gamma)$ and the MPK at hand, $\mathscr{S}$ undertakes the verification of accountability through the application of specific equations,

$$\begin{aligned} e\left(\Gamma_2, h^\alpha\right) &= e\left(\Gamma_1, h^{\alpha^n}\right), \\ e\left(h, h^{\prod_{x_{i_j} \in X} \left(\alpha + x_{i_j}\right)}\right) &= e\left(\mathbf{K}\left(x_{i_1}, x_{i_2}, \ldots x_{i_t}\right), \Gamma_1\right). \end{aligned} \tag{17}$$

When the conditions of both equations are satisfied, $\mathscr{R}$ confirms that the provided keyword token corresponds to a singular keyword trapdoor, indicated by $|\mathbf{K}(x_{i_1}, x_{i_2}, \ldots x_{i_t})| = 1$. In the event of a failure to meet these conditions, the operation is terminated. Upon possessing MSK and the keyword set X, entity $\mathscr{C}$ proceeds to generate the specific trapdoor D.

$$T = \mathbf{K}\left(x_{i_1}, x_{i_2}, \ldots x_{i_t}\right)^{(1/\beta)}. \tag{18}$$

Subsequently, $\mathscr{C}$ hands over the trapdoor D to $\mathscr{R}$.

$\mathscr{C}$: Upon this transfer, with C_i, the trapdoor D, and the secret key sk in possession, $\mathscr{R}$ commences the search operation

$$c_{2i} = H\left(0, e\left(c_{1i}, D\right)^{(1/s\tau)}\right). \tag{19}$$

Upon verification, $\mathscr{R}$ proceeds with the next steps of the decryption process

$$m_i = c_{3i} \oplus H\left(1, e\left(c_{1i}, D\right)^{(1/s\tau)}\right). \tag{20}$$

3.2.4. Correctness. With the (MPK, MSK) obtained through the execution of the **Setup** algorithm, and equipped with the token or proof combination $(\mathbf{K}(x_{i_1}, x_{i_2}, \ldots x_{i_t}), \Gamma)$, the

validation of accountability's accuracy is determined through a series of specific equations.

$$\begin{aligned} e(\Gamma_2, h^{\alpha}) &= e\left(\Gamma_1^{\alpha^{n-1}}, h^{\alpha}\right) \\ &= e\left(\Gamma_1, h^{\alpha^n}\right), \\ e\left(h, h^{\prod_{x_{i_j}\in X}\left(\alpha+x_{i_j}\right)}\right) &= e\left(h^{s\tau}, h^{(\gamma/s)}\right) \\ &= e\left(\mathbf{K}\left(x_{i_1}, x_{i_2}, \ldots, x_{i_t}\right), \Gamma_1\right). \end{aligned} \tag{21}$$

Upon receiving the ciphertext C_i generated through the **Commit** algorithm, along with the trapdoor D created via the **Transfer**, and the user's sk, one can authenticate the effectiveness of the search and decryption processes

$$\begin{aligned} H\left(0, e\left(c_{1i}, D\right)^{(1/s\tau)}\right) &= H\left(0, e\left(g^{r_i\beta}, h^{(s\tau/\beta)}\right)^{(1/s\tau)}\right) \\ &= H\left(0, e(g,h)^{r_i}\right) \\ &= c_{2i}, \\ c_{3i} \oplus H\left(1, e\left(c_{1i}, D\right)^{(1/s\tau)}\right) &= H\left(1, e(g,h)^{r_i}\right) \oplus m_i \oplus H\left(1, e\left(g^{r_i\beta}, h^{(s\tau/\beta)}\right)^{(1/s\tau)}\right) \\ &= H\left(1, e(g,h)^{r_i}\right) \oplus m_i \oplus H\left(1, e(g,h)^{r_i}\right) \\ &= m_i. \end{aligned} \tag{22}$$

4. Security Analysis

Our analysis focuses on the security aspects of the OMKSA protocol, in accordance with the framework outlined in Section 2.3. This security reduction draws its foundation from the complex problems specified in Section 2.4. A thorough and systematic proofing process is outlined in the subsequent discussion.

4.1. Receiver Privacy

Theorem 1. *In the framework, the assurance of absolute keyword privacy for the token, as it pertains to the receiver, is achieved within the context of the privacy game.*

Proof 1. Consider X as the set of authorized keywords, and the pair $(\mathbf{K}(X), \Gamma)$ derived from the condition $x = x_0$. The resultant formulation of the keyword token and its corresponding proof is as follows:

$$\begin{aligned} \mathbf{K}(X) &= h^{s\prod_{x_j\in X, x_j\neq x_0}\left(\alpha+x_j\right)}, \\ \Gamma &= \left(\Gamma_1 = h^{\left(\alpha+x_0/s\right)}, \Gamma_2 = h^{\left(\alpha^{n-1}\left(\alpha+x_0\right)/s\right)}\right). \end{aligned} \tag{23}$$

In the case of a unique keyword x_1, and with s' chosen from $\mathbb{Z}_p$, we adopt the implicit assignment $s' = s \cdot (\alpha + x_1)/(\alpha + x_0)$. This results in an equivalence of keyword tokens, namely, $\mathbf{K}(x_0)$ equals $\mathbf{K}(x_1)$, a fact that is substantiated through the following verification:

$$\begin{aligned} \mathbf{K}(x_0) &= h^{s\prod_{x_j\in X, x_j\neq x_0}\left(\alpha+x_j\right)} \\ &= h^{s'\prod_{x_j\in X, x_j\neq x_1}\left(\alpha+x_j\right)} \\ &= \mathbf{K}(x_1). \end{aligned} \tag{24}$$

Let us consider Γ' to be composed of (Γ_1', Γ_2'). In this scenario, the accountability proofs are found to be congruent, signifying Γ_1 is equal to Γ_1' and Γ_2 is equal to Γ_2'. This equivalency can be substantiated through the following validation:

$$\begin{aligned} \Gamma_1 &= h^{\left(\alpha+x_0/s\right)} \\ &= h^{\left(\alpha+x_1/s'\right)} \\ &= \Gamma_1', \\ \Gamma_2 &= h^{\left(\alpha^{n-1}\left(\alpha+x_0\right)/s\right)} \\ &= h^{\left(\alpha^{n-1}\left(\alpha+x_1\right)/s'\right)} \\ &= \Gamma_2'. \end{aligned} \tag{25}$$

The equivalence $(\mathbf{K}(x_0), \Gamma) = (\mathbf{K}(x_1), \Gamma')$ holds true. Given that s is selected at random from $\mathbb{Z}_p$, it follows that s' possesses uniform randomness within $\mathbb{Z}_p$ as well. Consequently, the distributions of $(\mathbf{K}(X), \Gamma)$ corresponding to both x_0 and x_1 are indistinguishable, offering no leverage to an adversary $\mathcal{A}$ in predicting the keyword encapsulated in $\mathbf{K}(X)$. This argument forms the basis for the conclusion of Theorem 1. □

4.2. Indistinguishability

Theorem 2. *It is structured to ensure semantic security and indistinguishability, as per the criteria of the Indistinguishability game, under the assumption that solving the (f,q)-MSE-DDH Problem is an arduous task, as delineated in [60].*

Proof 2. Imagine an adversary, referred to as $\mathscr{A}$, capable of undermining this indistinguishability. Under this hypothesis, we can devise an algorithm, named $\mathscr{B}$, specifically engineered to tackle the (f,q)-MSE-DDH Problem. Given a scenario with a (f,q)-MSE-DDH Problem instance and a variable Z in $\mathbb{G}_T$, the mission for $\mathscr{B}$ is to ascertain if Z is equal to $e(g_0,h_0)^{rq(\alpha)}$ or if it is a randomly chosen element within $\mathbb{G}_T$. The strategic interplay between $\mathscr{B}$ and $\mathscr{A}$ is elaborated in the following explanation. □

4.2.1. Setup. The foundational premise posits a universal keyword space, represented as $\mathscr{KS} = \{x_1, x_2, \ldots, x_n\}$. $\mathscr{B}$ selects a specific keyword x_θ from $\mathscr{KS}$, with the associated message symbolized as m_θ. The algorithm tacitly establishes the polynomials.

$$f(\alpha) = \alpha + x_\theta,$$
$$q(\alpha) = \prod_{\substack{x_j \in \mathscr{KS} \\ x_j \neq x_\theta}} \left(\alpha + x_j\right) \tag{26}$$

Additionally, the algorithm lets $g = g_0, h = h_0^{f(\alpha)q(\alpha)}$ and it gets $h_i = h_0^{\alpha^{i-1} f(\alpha)q(\alpha)}$. Following this, $\mathscr{B}$ forwards MPK to adversary $\mathscr{A}$, where

$$\text{MPK} = \left(g_0, g_0^{\alpha}, h_1, h_2, \ldots, h_{n+1}, \mathscr{PG}\right). \tag{27}$$

4.2.2. H-Query. The entity $\mathscr{B}$ operates a hash list designated as $L(a_i, X_i, h^i)$, starting in an empty state. When a query for $H(a_i, X_i)$ is received and if the tuple (a_i, X_i) already exists within L, then $\mathscr{B}$ provides the associated h^i back to $\mathscr{A}$. In cases where (a_i, X_i) is not present in L, $\mathscr{B}$ determines the hash value h^i using the following approach.

$$h^i = H(a_i, X_i) = \begin{cases} b_0^i, & \text{if } a_i = 0, \\ b_1^i, & \text{if } a_i = 1. \end{cases} \tag{28}$$

Selections for b_0^i, b_1^i are made randomly from the binary set $\{0,1\}^\ell$. Subsequently, $\mathscr{B}$ incorporates (a_i, X_i, h^i) into the list and furnishes h^i back to $\mathscr{A}$.

4.2.3. Phase 1. In this phase, $\mathscr{A}$ selects a set of keywords X, confined within the bounds of $\mathscr{KS}$, and limited to a maximum size of $|X|$ not more than n. For a trapdoor request corresponding to any keyword x_i from $X, \mathscr{A}$ opts for a random $s \in \mathbb{Z}_p$ as $sk = s$, and communicates (x_i, s) to $\mathscr{B}$.

Should x_i matches x_θ, the process is terminated.

In contrast, if x_i differs from $x_\theta, \mathscr{B}$ delivers $T = h_0^{sq_i(\alpha)f(\alpha)}$ to $\mathscr{A}$, with $q_i(\alpha)$ defined as $(q(\alpha))/(\alpha + x_i)$. Subsequent steps involve verifying the trapdoor.

$$\begin{aligned} T &= \mathbf{K}(x_i)^{\left(1/\Pi_{x_j \in X}\left(\alpha + x_j\right)\right)} \\ &= h^{\left(s\prod_{x_j \in X, j \neq i}\left(\alpha + x_j\right)/\prod_{x_j \in X}\left(\alpha + x_j\right)\right)} \\ &= h^{\left(sf(\alpha)q(\alpha)/\alpha + x_i\right)} \\ &= h_0^{sq_i(\alpha)f(\alpha)}. \end{aligned} \tag{29}$$

It becomes apparent that the expression $h_0^{q_i(\alpha)f(\alpha)}$ can be computed from elements $h_0^{f(\alpha)}, h_0^{\alpha f(\alpha)}, h_0^{\alpha^{n-2} f(\alpha)}$ as provided in the instance.

4.2.4. Challenge. When $\mathscr{A}$ presents the pairs (m_0, x_0) and (m_1, x_1) as a challenge to $\mathscr{B}$, and assuming no prior trapdoor queries have been made for x_0 or x_1, the protocol proceeds as follows.

If x_θ is neither x_0 nor x_1, the process halts.

However, if x_θ matches either x_0 or x_1, $\mathscr{B}$ verifies the presence of $(0, Z)$ and $(1, Z)$ within the list L. If found, $\mathscr{B}$ retrieves and labels the corresponding hash values as b_0^* and b_1^*. In the absence of these entries, $\mathscr{B}$ randomly generates b_0^* and b_1^* from the binary set $\{0,1\}^\ell$ and proceeds to set them accordingly.

$$H(0, Z) = b_0^*, H(1, Z) = b_1^*. \tag{30}$$

Subsequently, $\mathscr{B}$ incorporates the pairs $(0, Z, b_0^*)$ and $(1, Z, b_1^*)$ into the compilation L. Following this, $\mathscr{B}$ furnishes $\mathscr{A}$ with the designated ciphertext for the challenge.

$$C^* = \left(c_1 = g_0^r, c_2 = b_0^*, c_3 = b_1^* \oplus m_\theta\right). \tag{31}$$

Verification of the condition $Z = e(g_0, h_0)^{rq(\alpha)}$ is feasible by assigning the value of r as $r_i f(\alpha)$ in an implicit manner.

$$\begin{aligned} c_1 &= g^{r_i(\alpha + x_\theta)} \\ &= g_0^{r_i f(\alpha)} \\ &= g_0^r, \\ c_2 &= H\left(0, e(g,h)^{r_i}\right) \\ &= H\left(0, e\left(g_0, h_0\right)^{rq(\alpha)}\right) \\ &= H(0, Z) \\ &= b_0^*, \\ c_3 &= H\left(1, e(g,h)^{r_i}\right) \oplus m_\theta \\ &= H\left(1, e\left(g_0, h_0\right)^{rq(\alpha)}\right) \oplus m_\theta \\ &= H(1, Z) \oplus m_\theta \\ &= b_1^* \oplus m_\theta. \end{aligned} \tag{32}$$

Hence, the ciphertext C* constitutes an appropriate challenge.

In the scenario where Z represents a random variable within $\mathbb{G}_T$, from the perspective of $\mathscr{A}$, C^* emerges as a random ciphertext.

4.2.5. Phase 2. $\mathscr{A}$ seeks further trapdoor queries for x_i, adhering to the constraint $x_i \neq x_0, x_1$. In response, $\mathscr{B}$ acts in accordance with the guidelines of Phase 1.

4.2.6. Guess. The adversary $\mathscr{A}$ generates a prediction, denoted as θ', aiming to ascertain the value of θ. With this step, our simulation narrative reaches its completion. Next, we transition to an analysis phase, probing the proficiency of $\mathscr{B}$ in unraveling the intricate hard problem. Let us assume q_T represents the total volume of trapdoor queries and n symbolizes the magnitude of the keyword space. In light of the preceding simulation dynamics, the probability of $\mathscr{B}$ persevering without an abortive outcome is delineated in the following sections.

$$\begin{aligned}\Pr[\text{abort}] &= \left(1-\frac{1}{n}\right)\left(1-\frac{1}{n-1}\right)\cdots\left(1-\frac{1}{n-q_T+1}\right)\\ &= \frac{n-q_T}{n}.\end{aligned} \tag{33}$$

Given that queries for x_0 and x_1 were not made in either Phase 1 or Phase 2, the quantity q_T is bounded by $n-2$. Consequently, the probability $\Pr[\text{abort}]$ is no less than $2/n$. Assuming $\mathscr{A}$'s success in compromising the security game is at least ϵ, the reduction in advantage can be represented as

$$\begin{aligned}\epsilon_{\text{reduction}} &= \Pr\left[\theta' = \theta \,\middle|\, Z = e(g_0, h_0)^{rq(\alpha)}\right] - \Pr\left[\theta' = \theta \,\middle|\, Z \text{ is an arbitrary element}\right]\\ &= \epsilon.\end{aligned} \tag{34}$$

From this, the advantage of $\mathscr{B}$ in resolving the $(f,q)-\text{MSE}-\text{DDH}$ Problem is calculated to be at least $\epsilon_{\mathscr{B}} = \Pr[\text{abort}]\epsilon_{\text{reduction}} = 2\epsilon/n$, thus finalizing the proof of Theorem 2.

4.3. Accountability

Theorem 3. *The framework's competence in ensuring accountability is validated within the accountability game, contingent on the intractability of the $(f,n)-$DHE Problem [17].*

Proof 3. Let us hypothesize the existence of an adversary $\mathscr{A}$ who is capable of compromising the accountability security. In this case, an algorithm $\mathscr{B}$ is formulated to address the $(f,n)-$DHE Problem. Confronted with a $(f,n)-$DHE challenge, $\mathscr{B}$ engages with $\mathscr{A}$ in the following manner. □

4.3.1. Setup. Algorithm $\mathscr{B}$ assigns $\alpha = a$, leading to the derivation of $h_1 = h, h_2 = h^a, \ldots, h_{n+1} = h^{a^n}$, as per the $(f,n)-$DHE scenario. Subsequently, $\mathscr{B}$ selects H akin to the actual scheme. Then $\mathscr{B}$ circulates the MPK for $\mathscr{A}$,

$$\text{MPK} = (g, g^a, h_1, h_2, \ldots, h_{n+1}, H, \mathscr{PG}). \tag{35}$$

4.3.2. Challenge. $\mathscr{A}$ selects two unique keyword sets, X and X', ensuring $|X'| > 1$ and $|X| \leq n$. A secret key sk is then established by randomly choosing a value s from $\mathbb{Z}_p$, with $sk = s$. Following this, $\mathscr{A}$ proposes $(\mathbf{K}(X'), X, X', sk)$ along with a challenge indication set to 1. Here, the token $\mathbf{K}$ is characterized as

$$\mathbf{K}(X') = h^{s\prod_{x_j \in X - X'}(a+x_j)}. \tag{36}$$

4.3.3. Win. The adversary $\mathscr{A}$ selects two keyword sets, X and X', conforming to the constraints $|X'| > 1$ and $|X| \leq n$. Upon choosing a random secret key s from $\mathbb{Z}_p$, denoted as $sk = s$, $\mathscr{A}$ then formulates and submits $(\mathbf{K}(X'), X, X', sk)$ with a challenge indicator set to 1. The token within this setup is characterized as

$$\Gamma = \left(\Gamma_1 = h^{(1/s)\prod_{x_j \in X'}(a+x_j)}, \Gamma_2 = \Gamma_1^{a^{n-1}} = h^{(1/s)a^{n-1}\prod_{x_j \in X'}(a+x_j)}\right). \tag{37}$$

Consequently, both the token and its associated proof are capable of successfully undergoing the matching process as follows:

$$\begin{aligned}e(\Gamma_2, h^a) &= e\left(\Gamma_1, h^{a^n}\right),\\ e\left(h, h^{\prod_{x_i \in X}(a+x_i)}\right) &= e(\mathbf{K}(X'), \Gamma_1).\end{aligned} \tag{38}$$

Let $f(x) = 1/sx^{n-1}\prod_{x_j \in X'}(x + x_j)$; then $\Gamma = h^{f(a)}$. Given that the function $f(x)$ is classified as a polynomial with a degree, $\deg f(x)$, surpassing n, it follows that $\mathscr{B}$ furnishes the pair $(f(x), \Gamma)$ as the resolution of the (f,n)-DHE Challenge. The conclusion of this proof of Theorem 3 leads to the deduction that the magnitude of X' is singular, $|X'| = 1$, and correspondingly, the token $\mathbf{K}(X')$ also possesses a singular count, $|\mathbf{K}(X')| = 1$.

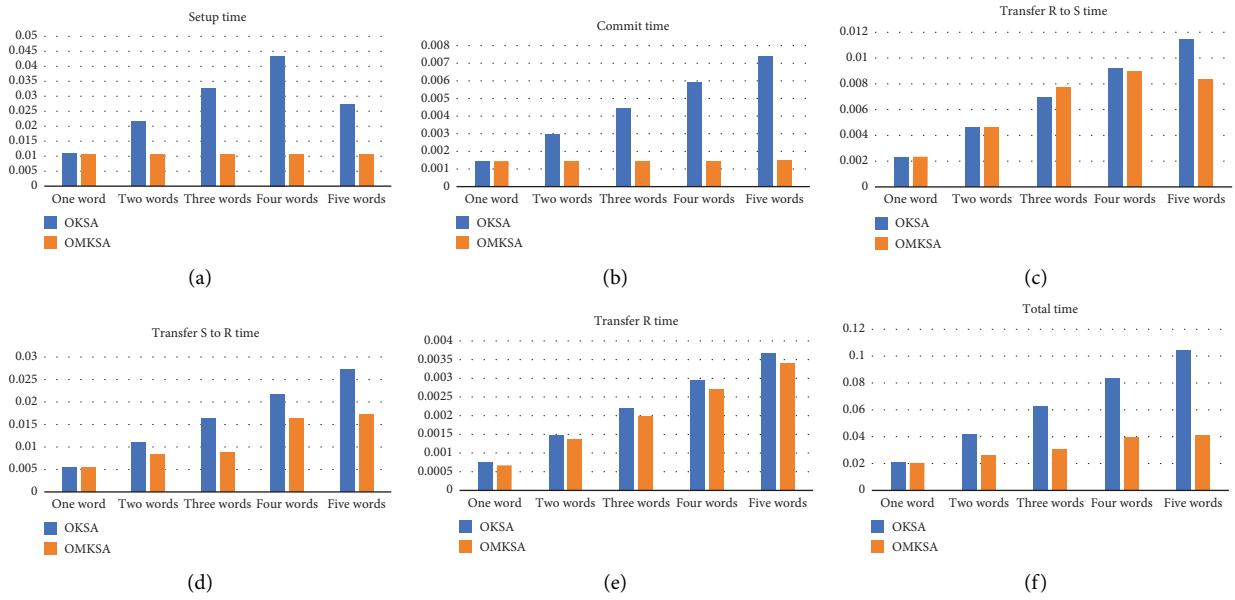

FIGURE 2: Comparison of search time between OKSA and OMKSA. (a) Setup time. (b) Commit time. (c) Transfer R to S time. (d) Transfer S to R time. (e) Transfer R time. (f) Total time.

5. Performance Evaluation

The experiment was conducted on the Ubuntu-22.04.3-desktop-amd64 operating system, with Linux hardware specifications: an 11th Gen Intel (R) Core i5-1135G7 CPU @2.40 GHz and 40.0 G of RAM. All program algorithms were implemented using the 8.3.1 20190311 (Red Hat 8.3.1-3) (GCC) compiler. The experimental data were derived from a dataset of $n = 10$, generated by random algorithms. We executed both the OKSA and OMKSA protocols on these data. OKSA is designed by Jang, which can only query keywords one by one. OMKSA is based on OMKSA made by us and can query multiple keywords at the same time. When we want to compare queries for the same keywords, our scheme is more efficient than the OKSA scheme.

To illustrate the effectiveness of actual search terms, we varied the number of query keywords D from 1 to 5. The following figure compares the two scenarios and depicts an increase in search time with the augmentation of keyword quantity. The items from left to right are setup time2a, commit time2b, transfer $\mathscr{R}$ to $\mathscr{C}$ time2c, transfer $\mathscr{C}$ to $\mathscr{R}$ time2d, transfer $\mathscr{R}$ time2e, and total time2f. We found that OMKSA is designed to query multiple keywords simultaneously, and the OMKSA protocol runtime does not increase with the number of queried words in the setup and commit algorithms. However, the OKSA scheme needs to query keywords one by one, and the query time will double as the number of query keywords increases.

As shown in Figure 2, at $t = 1$, the total time consumed by both protocols is almost the same; however, for $t \geq 2$, the total time consumption of OMKSA is notably less than that of OKSA. Furthermore, the increase in duration for OMKSA is minimal, indicating its aptness for retrieval under multi-keyword scenarios.

Since cloud servers are not trusted, replay attacks are possible. We can further solve the problem of cloud server replay attacks in the future to further ensure the security of cloud storage. In addition, the proposed scheme works well in the laboratory environment, but the experimental data are limited, and it has not been applied in the real scene, which may cause unknown problems. It is necessary to further improve the scheme in the process of application.

6. Application in ML

The applications of OMKSA within the realm of ML are interesting. Users are able to execute ML tasks on encrypted data domains without revealing the contents of training data or the queries. Integrating this with ML poses both challenges and significant practical value. For instance, data in sectors like healthcare and finance, even when encrypted, may still require ML operations. In addition, users might not want to expose the specifics of their queries or the parameters and structure of their ML models.

6.1. Applied on the Medical Platform. The convergence of AI and SE holds expansive prospects for application within the healthcare platform. This realm of possibility not only underscores the importance of leveraging technological advances but also signals a transformative shift in the way healthcare data are managed and accessed.

To expand on this, we must first understand the tremendous role that AI plays in healthcare. AI systems have the potential to revolutionize the medical field by offering personalized medicine, predictive analytics, and enhanced patient care. From analyzing medical images for early disease detection to predicting patient outcomes based on data trends, AI is pushing the boundaries of what is achievable in

medicine. Also, AI is now poised to bridge the patient–doctor gap with automatic diagnostic tools that can operate round the clock. On the other hand, the significance of SE in protecting patient data cannot be overstated. The plethora of data generated in healthcare, from patient records to vital signs, necessitates a stringent data privacy policy. SE offers the solution by not only encrypting these data but also allowing them to be searched without being decrypted. It provides an avenue for secure querying of databases while safeguarding sensitive patient information.

Now, let us consider the combined force of AI and SE. When these technologies are integrated, we can pave the way for a reliable, encrypted database where AI can freely and securely extract insights. The use of SE can ensure the privacy of the data while AI can efficiently analyze the data to facilitate improved healthcare. This fusion of technologies can thus pave the way for a new era of medical treatment and patient management. For instance, consider a healthcare portal where users employ distinct keywords to classify their health issues. The company operating this medical platform employs Public Key SE to securely store user information and uploads encrypted data onto a cloud server. To broaden their business scope, the firm opts to delve deeper into research involving its user base, such as identifying common illnesses like "Hypertension," "Periarthritis of Shoulder," "Cervical Spondylosis," "Osteoporosis," and "Diabetes." To achieve this objective, the organization delegates the task to an expert external agency, which is tasked with managing these activities while rigorously safeguarding the confidentiality of user health information. Figure 3 illustrates the framework, with the main steps outlined below.

1. Data protection and encryption process: The company meticulously gathers user data, utilizing Public Key SE techniques to securely encrypt the information prior to its upload.
2. Task transfer: The company establishes the parameters by collaborating with the third party, utilizing OMKSA for secure transformation of the encrypted data.
3. Task computation: The external agency conducts analytical operations and then relays the findings back to the company.

6.2. Data Preprocessing and Feature Selection. In contemporary ML applications, data preprocessing is not just a prerequisite step, but it also profoundly influences the overall model performance and accuracy. To achieve optimal training results, it might be necessary to sift through vast encrypted data repositories to select critical data subsets for comprehensive analysis. This step often entails choosing and categorizing a myriad of features. With the introduction of the OMKSA technique, researchers can now employ multi-keyword searches to select specific data or features from encrypted datasets. Crucially, this process ensures that there is no disclosure regarding which particular keywords the researchers are keen on, thereby preserving search privacy.

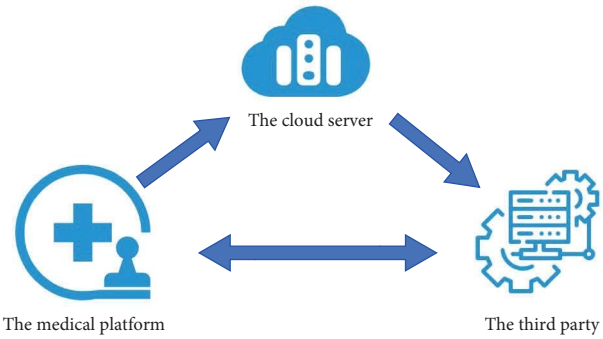

Figure 3: An application of OMKSA in machine learning.

6.3. Distributed ML. Distributed ML and federated learning have become essential strategies for multi-organizational collaborative ventures. In such settings, various entities or organizations aim to collaboratively develop a robust, shared ML model. However, due to concerns spanning data privacy, legal regulations, and intellectual property rights, the direct sharing of raw data becomes infeasible. The advent of OMKSA provides a novel solution to this conundrum. Using OMKSA, one organization can securely authorize another entity to search specific segments of its encrypted data. This means multiple entities can collaborate and train models without directly sharing any tangible data.

7. Conclusion

In this study, we introduced the advanced concept of OMKSA, which improves the traditional OKSA scheme to accommodate multi-keyword search. While traditional OKSA limits each encrypted data file to a single keyword association, OMKSA gives data providers the flexibility to extract multiple keywords for each encrypted dataset. Our analysis and experiments confirm that the OMKSA protocol is both practical and easy to implement for querying encrypted data in a cloud environment. A significant innovation of our approach lies in its integration of aggregation algorithms to generate the trapdoor. Consequently, the sender sent a fixed-size trapdoor to the receiver, enhancing the efficiency of communication.

Additionally, a comprehensive security analysis of the proposed protocol has been conducted, adhering to recognized security models. Notably, when benchmarked against existing OKSA protocols, our OMKSA stands out as exceptionally efficient for data seekers. Moreover, the computational overhead for conjunctive keyword searches remains stable, regardless of an increase in the volume of query keywords. Finally, for the replay attack problem, we may use the password accumulator tool in the future to further solve the problem of cloud server replay attack to ensure the security of cloud storage. Conclusively, as we transition into the 5G era, the potential applications for OMKSA are vast, with pronounced implications for domains like ML and AI. Our findings pave the way for further exploration and deployment in these cutting-edge fields.

Data Availability Statement

The data used to support the findings of this study are available from the corresponding authors upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Funding

This work was supported by the National Natural Science Foundation of China (Nos. 62402290, U24A20244, 62302280, and 62072276), Natural Science Foundation of Shandong Province (No. ZR2023QF133), and Youth Innovation Team of Shandong Higher Education Institutions (No. 2024KJN051).

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 62402290, U24A20244, 62302280, and 62072276), Natural Science Foundation of Shandong Province (No. ZR2023QF133), and Youth Innovation Team of Shandong Higher Education Institutions (No. 2024KJN051).

References

[1] H. Jia, M. S. Aldeen, C. Zhao, S. Jing, and Z. Chen, "Flexible Privacy-Preserving Machine Learning: When Searchable Encryption Meets Homomorphic Encryption," *International Journal of Intelligent Systems* 37, no. 11 (2022): 9173–9191, https://doi.org/10.1002/int.22985.

[2] R. Shouval, J. A. Fein, B. N. Savani, M. Mohty, and A. Nagler, "Machine Learning and Artificial Intelligence in Haematology," *British Journal of Haematology* 192, no. 2 (2020): 239–250, https://doi.org/10.1111/bjh.16915.

[3] D. X. Song, D. A. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," in *Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000* (2000), 44–55.

[4] D. Boneh, G. Di Crescenzo, R. M. Ostrovsky, and G. Persiano, "Public Key Encryption With Keyword Search," *Lecture Notes in Computer Science* (2004): 506–522, https://doi.org/10.1007/978-3-540-24676-3_30.

[5] J. Liu, B. Zhao, J. Qin, X. Zhang, and J. Ma, "Multi-Keyword Ranked Searchable Encryption With the Wildcard Keyword for Data Sharing in Cloud Computing," *The Computer Journal* 66, no. 1 (2021): 184–196, https://doi.org/10.1093/comjnl/bxab153.

[6] X. Zhang, B. Zhao, J. Qin, W. Hou, Y. Su, and H. Yang, "Practical Wildcard Searchable Encryption With Tree-Based Index," *International Journal of Intelligent Systems* 36, no. 12 (2021): 7475–7499, https://doi.org/10.1002/int.22595.

[7] J. W. Byun, H. S. Rhee, H. A. Park, and D. H. Lee, "Off-Line Keyword Guessing Attacks on Recent Keyword Search Schemes Over Encrypted Data," *Lecture Notes in Computer Science* (2006): 75–83, https://doi.org/10.1007/11844662_6.

[8] I. R. Jeong, J. O. Kwon, D. Hong, and D. H. Lee, "Constructing PEKS Schemes Secure Against Keyword Guessing Attacks Is Possible?" *Computer Communications* 32, no. 2 (2009): 394–396, https://doi.org/10.1016/j.comcom.2008.11.018.

[9] M. Abdalla, M. Bellare, D. Catalano, et al., "Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions," *Journal of Cryptology* 21, no. 3 (2008): 350–391, https://doi.org/10.1007/s00145-007-9006-6.

[10] W. C. Yau, C. W. Phan, S. H. Heng, and B. M. Goi, "Keyword Guessing Attacks on Secure Searchable Public Key Encryption Schemes With a Designated Tester," *International Journal of Computer Mathematics* 90, no. 12 (2013): 2581–2587, https://doi.org/10.1080/00207160.2013.778985.

[11] D. Boneh, E. Kushilevitz, R. M. Ostrovsky, and W. E. Skeith, "Public Key Encryption That Allows PIR Queries," *Lecture Notes in Computer Science* (2007): 50–67, https://doi.org/10.1007/978-3-540-74143-5_4.

[12] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "Dual-Server Public-Key Encryption With Keyword Search for Secure Cloud Storage," *IEEE Transactions on Information Forensics and Security* 11, no. 4 (2016): 789–798, https://doi.org/10.1109/tifs.2015.2510822.

[13] S. Zhi-Yi, "On Security Against the Server in Designated Tester Public Key Encryption with Keyword Search," *Information Processing Letters* (2015).

[14] Q. Tang and L. Chen, "Public-Key Encryption With Registered Keyword Search" (2009).

[15] Q. Huang and H. Li, "An Efficient Public-Key Searchable Encryption Scheme Secure Against Inside Keyword Guessing Attacks," *Information Sciences* 403–404 (2017): 1–14, https://doi.org/10.1016/j.ins.2017.03.038.

[16] W. Ogata and K. Kurosawa, "Oblivious Keyword Search," *Journal of Complexity* 20, no. 2-3 (2004): 356–371, https://doi.org/10.1016/j.jco.2003.08.023.

[17] P. Jiang, X. Wang, J. Lai, F. Guo, and R. Chen, "Oblivious Keyword Search With Authorization," *Lecture Notes in Computer Science* (2016): 173–190, https://doi.org/10.1007/978-3-319-47422-9_10.

[18] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M. C. Roşu, and M. Steiner, "Highly-Scalable Searchable Symmetric Encryption With Support for Boolean Queries," *Lecture Notes in Computer Science* (2013): 353–373, https://doi.org/10.1007/978-3-642-40041-4_20.

[19] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," *Journal of Computer Security* 19, no. 5 (2011): 895–934, https://doi.org/10.3233/jcs-2011-0426.

[20] S. Jarecki, C. Jutla, H. Krawczyk, M. Rosu, and M. Steiner, "Outsourced Symmetric Private Information Retrieval," *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security-CCS' 13* (2013): 875–888, https://doi.org/10.1145/2508859.2516730.

[21] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic Searchable Symmetric Encryption" (2012).

[22] H. S. Rhee, W. Susilo, and H. J. Kim, "Secure Searchable Public Key Encryption Scheme Against Keyword Guessing Attacks," *IEICE Electronics Express* 6, no. 5 (2009): 237–243, https://doi.org/10.1587/elex.6.237.

[23] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Improved Searchable Public Key Encryption With Designated Tester," *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security* (2009): 376–379, https://doi.org/10.1145/1533057.1533108.

[24] L. Fang, W. Susilo, C. Ge, and J. Wang, "Public Key Encryption With Keyword Search Secure Against Keyword Guessing Attacks Without Random Oracle," *Information*

Sciences 238 (2013): 221–241, https://doi.org/10.1016/j.ins.2013.03.008.

[25] M. Noroozi and Z. Eslami, "Public Key Authenticated Encryption With Keyword Search: Revisited," *IET Information Security* 13, no. 4 (2019): 336–342, https://doi.org/10.1049/iet-ifs.2018.5315.

[26] L. Ballard, S. Kamara, and F. Monrose, "Achieving Efficient Conjunctive Keyword Searches Over Encrypted Data," *Lecture Notes in Computer Science* (2005): 414–426, https://doi.org/10.1007/11602897_35.

[27] D. Boneh and B. Waters, "Conjunctive, Subset, and Range Queries on Encrypted Data," *IACR Cryptology ePrint Archive* 2006 (2007): 287.

[28] D. J. Park, K. Kim, and P. J. Lee, "Public Key Encryption With Conjunctive Field Keyword Search" (2004).

[29] E. K. Ryu and T. Takagi, "Efficient Conjunctive Keyword-Searchable Encryption," in *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, 1 (2007), 409–414, https://doi.org/10.1109/ainaw.2007.166.

[30] S. Sedghi, P. van Liesdonk, S. Nikova, P. H. Hartel, and W. Jonker, "Searching Keywords With Wildcards on Encrypted Data," *Lecture Notes in Computer Science* (2010): 138–153, https://doi.org/10.1007/978-3-642-15317-4_10.

[31] M. Abdalla, M. Bellare, D. Catalano, et al., "Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions," *Journal of Cryptology* 21, no. 3 (2008): 350–391, https://doi.org/10.1007/s00145-007-9006-6.

[32] J. Camenisch, M. Kohlweiss, A. Rial, and C. Sheedy, "Blind and Anonymous Identity-Based Encryption and Authorised Private Searches on Public Key Encrypted Data," *Lecture Notes in Computer Science* (2009): 196–214, https://doi.org/10.1007/978-3-642-00468-1_12.

[33] J. Shi, J. Lai, Y. Li, R. H. Deng, and J. Weng, "Authorized Keyword Search on Encrypted Data," *Lecture Notes in Computer Science* (2014): 419–435, https://doi.org/10.1007/978-3-319-11203-9_24.

[34] 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* (2014), 226–234, https://doi.org/10.1109/infocom.2014.6847943.

[35] 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* (2014), 522–530.

[36] J. Li, J. Li, X. Chen, Z. Liu, and C. Jia, "Privacy-Preserving Data Utilization in Hybrid Clouds," *Future Generation Computer Systems* 30 (2014): 98–106, https://doi.org/10.1016/j.future.2013.06.011.

[37] J. Li, D. Lin, A. C. Squicciarini, J. Li, and C. Jia, "Towards Privacy-Preserving Storage and Retrieval in Multiple Clouds," *IEEE Transactions on Cloud Computing* 5, no. 3 (2017): 499–509, https://doi.org/10.1109/tcc.2015.2485214.

[38] M. O. Rabin, "How to Exchange Secrets with Oblivious Transfer," *IACR Cryptology ePrint Archive* 2005 (2005): 187.

[39] C. K. Chu and W. G. Tzeng, "Efficient K-Out-of-N Oblivious Transfer Schemes With Adaptive and Non-Adaptive Queries," *Lecture Notes in Computer Science* (2005): 172–183, https://doi.org/10.1007/978-3-540-30580-4_12.

[40] K. Kurosawa and R. Nojima, "Simple Adaptive Oblivious Transfer Without Random Oracle," *IACR Cryptology ePrint Archive* 2009 (2009): 327.

[41] J. Camenisch, M. Dubovitskaya, and G. Neven, "Oblivious Transfer With Access Control," *IACR Cryptology ePrint Archive* 2009 (2009): 529.

[42] H. S. Rhee, J. W. Byun, D. H. Lee, and J. Lim, "Oblivious Conjunctive Keyword Search" (2005).

[43] M. J. Freedman, Y. Ishai, B. Pinkas, and O. Reingold, "Keyword Search and Oblivious Pseudorandom Functions," *Lecture Notes in Computer Science* (2005): 303–324, https://doi.org/10.1007/978-3-540-30576-7_17.

[44] H. Zhu and F. Bao, "Oblivious Keyword Search Protocols in the Public Database Model," in *2007 IEEE International Conference on Communications* (2007), 1336–1341.

[45] Z. Jiang and L. Liu, "Secure Cloud Storage Service With an Efficient DOKS Protocol," in *2013 IEEE International Conference on Services Computing* (2013), 208–215, https://doi.org/10.1109/scc.2013.37.

[46] R. Zhang, R. Xue, L. Liu, and L. Zheng, "Oblivious Multi-Keyword Search for Secure Cloud Storage Service," in *2017 IEEE International Conference on Web Services (ICWS)* (Honolulu, HI, 2017), 269–276.

[47] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine Learning Classification over Encrypted Data," *IACR Cryptology ePrint Archive* 2014 (2015): 331.

[48] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. E. Lauter, M. Naehrig, and J. R. Wernsing, "CryptoNets: Applying Neural Networks to Encrypted Data With High Throughput and Accuracy" (2016).

[49] N. Papernot, M. Abadi, Ú Erlingsson, I. J. Goodfellow, and K. Talwar, "Semi-Supervised Knowledge Transfer for Deep Learning From Private Training Data," *ArXiv* (2016).

[50] S. Kamil Khudhair, M. Sahu, R. R. K, and A. K. Sahu, "Secure Reversible Data Hiding Using Block-Wise Histogram Shifting," *Electronics* 12, no. 5 (2023): 1222, https://doi.org/10.3390/electronics12051222.

[51] A. K. Sahu and M. Sahu, "Digital Image Steganography Techniques in Spatial Domain-A Study," *International Journal of Pharmacy and Technology* 8, no. 4 (2017): 5205–5217.

[52] G. Swain and A. K. Sahu, "A Novel Multi Stego-Image Based Data Hiding Method for Gray Scale Image," *Pertanika Journal of Science and Technology* 27, no. 2 (2019): 753–768.

[53] K. R. Raghunandan, R. Dodmane, K. Bhavya, N. S. K. Rao, and A. K. Sahu, "Chaotic-Map Based Encryption for 3D Point and 3D Mesh Fog Data in Edge Computing," *IEEE Access* 11 (2023): 3545–3554, https://doi.org/10.1109/access.2022.3232461.

[54] P. Asmitha, C. Rupa, S. Nikitha, J. Hemalatha, and A. K. Sahu, "Improved Multiview Biometric Object Detection for Anti Spoofing Frauds," *Multimedia Tools and Applications* 83, no. 33 (2024): 80161–80177, https://doi.org/10.1007/s11042-024-18458-8.

[55] R. K. Ramesh, R. Dodmane, S. Shetty, G. Aithal, M. Sahu, and A. K. Sahu, "A Novel and Secure Fake-Modulus Based Rabin-3 Cryptosystem," *Cryptography* 7, no. 3 (2023): 44, https://doi.org/10.3390/cryptography7030044.

[56] P. Mohassel and Y. Zhang, "SecureML: A System for Scalable Privacy-Preserving Machine Learning," in *2017 IEEE Symposium on Security and Privacy (SP)* (2017), 19–38, https://doi.org/10.1109/sp.2017.12.

[57] J. Liu, M. Juuti, Y. Lu, and N. Asokan, "Oblivious Neural Network Predictions via MiniONN Transformations," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017).

[58] S. D. Galbraith, K. G. Paterson, and N. P. Smart, "Pairings for Cryptographers," *Discrete Applied Mathematics* 156, no. 16 (2008): 3113–3121, https://doi.org/10.1016/j.dam.2007.12.010.

[59] F. Guo, Y. Mu, W. Susilo, and V. Varadharajan, "Membership Encryption and Its Applications," *Lecture Notes in Computer Science* (2013): 219–234, https://doi.org/10.1007/978-3-642-39059-3_15.

[60] D. Boneh, X. Boyen, and E. J. Goh, "Hierarchical Identity Based Encryption with Constant Size Ciphertext," *Lecture Notes in Computer Science* (2005): 440–456, https://doi.org/10.1007/11426639_26.