This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3550198

Oblivious Keyword Search with Authorization and Verification for IoT Devices in Untrusted Cloud Environments

Zhongkai Wei, Bo Zhao, Haining Yang, Jing Qin, Jixin Ma

***Abstract*—With the rapid advancement of Internet of Things (IoT) technology, large volumes of data are exchanged among users via cloud servers. However, in an untrusted cloud server environment, the risk of data tampering is significant. For instance, a cloud server may fail to update its records promptly after receiving updated data from a data sender. Consequently, when the data receiver retrieves the relevant information, the cloud server may return outdated data, leading to security issues in data utilization. To address this problem, we propose a scheme that facilitates efficient verification in untrustworthy cloud environments. Our research approach is to utilize cryptographic accumulators within the oblivious searchable encryption model to achieve efficient verification. The data sender first uses a cryptographic accumulator to calculate the cumulative value of all messages to be uploaded, which are publicly accessible. In addition, the accumulator generates witness values for messages authorized to the data recipient. Before retrieving data, the data receiver can leverage the cryptographic accumulator to verify the timeliness of incoming messages, ensuring that the data is current and free from tampering. Furthermore, the data sender retains the flexibility to dynamically update the data stored in the cloud and efficiently refresh both the encrypted accumulator and its corresponding witness value. This paper presents a rigorous security proof and a comparative experiment was carried out, supported by both analytical evaluations and experimental results, which collectively confirm the practical applicability of the proposed scheme in the context of the Internet of Things (IoT).**

***Index Terms*—Internet of Things (IoT), Cloud Servers, Searchable Encryption, Oblivious Keyword Search, Authentication, Cryptographic Accumulator.**

I. Introduction

THE Internet of Things (IoT) is a global network of interconnected devices that bridges the gap between the physical world and the virtual world of electronics [1]. With the advent of the 5G network, IoT has made significant strides across a variety of sectors, including smart cities, smart homes, and healthcare [2], [3]. With the rapid advances of Internet of things [4]–[6] technology. Concurrently, the rapid growth of cloud computing has provided cloud servers with large storage capacities, high scalability, and flexible availability, making them well-suited to meet the complex demands and high fragmentation inherent in IoT systems [7], [8].

Cloud storage technology provides users with a cost-effective and convenient way to store data, attracting them to upload large amounts of data to the cloud. However, due to the increasing prevalence of cyberattacks, data breaches have become a frequent occurrence, posing significant risks to the security of both business and consumer data [9], [10]. In 2024, for example, the cloud data giant Snowflake experienced a series of data theft incidents, during which cybercriminals managed to steal and expose hundreds of millions of records. The number of compromised records continues to rise, underscoring the urgent need to strengthen data protection mechanisms for cloud storage.

The Cloud Security Alliance has emphasized that, on the cloud, a file that is not encrypted is effectively considered compromised. As a result, encrypting data before uploading it to the cloud has become a standard practice to safeguard data privacy. However, encryption introduces a challenge: it impedes users' ability to search for and efficiently retrieve encrypted data. This trade-off between security and usability has led to a growing demand for encryption techniques that allow for secure data search. Developing such techniques has become a key focus of research and innovation, attracting significant attention from both academia and industry.

Searchable encryption addresses the challenge of searching encrypted data stored on cloud servers. It can be categorized into two primary types based on the encryption techniques used: Searchable Symmetric Encryption (SSE) [11] and Public Key Searchable Encryption (PKSE) [12]. Research in searchable encryption has largely focused on enhancing security, particularly in terms of resisting keyword guessing attacks [13]. In the context of untrusted cloud servers, the security requirements for searchable encryption can be summarized as follows: 1) When the sender uploads ciphertext to the cloud server, the server should not be able to deduce the plaintext content; 2) When the receiver searches the cloud server, the server should not learn the specific query being made; 3) Even though the server does not know the query, it must still be able to return the correct information without knowing what is being returned.

In this regard, Ogata et al. proposed the Oblivious Keyword Search (OKS) scheme [14], which allows users to search for data associated with specific keywords without revealing the content of the query. This method ensures that searchable

This work is supported by the National Natural Science Foundation of China (No.62402290, No.U24A20244, No.62072276) and the Youth Innovation Team of Shandong Higher Education Institutions (No.2024KJN051). (Corresponding author: Haining Yang and Jing Qin.)

Zhongkai Wei is with the School of Mathematics, Shandong University, Jinan 250100, China (e-mail: zhongkaicode@163.com)

Bo Zhao is with the School of Mathematics, Shandong University, Jinan 250100, China (e-mail: zhaobomath@163.com)

Haining Yang is with the School of Mathematics, Shandong University, Jinan 250100, China (e-mail: hainingyang@sdu.edu.cn)

Jing Qin is with the School of Mathematics, Shandong University, Jinan 250100, China (e-mail: qinjing@sdu.edu.cn)

Jixin Ma is with the Centre for Computer and Computational Science, School of Computing and Mathematical Sciences, University of Greenwich, SE10 9LS London, U.K. (e-mail: j.ma@greenwich.ac.uk)

Copyright (c) 20xx IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Authorized licensed use limited to: University of Greenwich. Downloaded on March 12,2025 at 20:42:48 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3550198

encryption remains secure even in the presence of untrusted cloud servers. In the Oblivious Keyword Search (OKS) scheme [14], the data sender generates a trapdoor for a specific keyword chosen by the data receiver, without knowing the keyword itself. At the same time, the data receiver is only able to retrieve the data associated with the corresponding searched keyword. The OKS protects the data privacy of both the data sender and the data receiver. For some precious data, the data sender may authorize only a part of the data to the data receiver. For certain sensitive data, the data sender may choose to authorize access to only a subset of the data for the receiver.itive data, the data sender may choose to authorize access to only a subset of the data for the receiver.

To address this issue, Peng Jiang et al. introduced Oblivious Keyword Search with Authorization (OKSA) [15], which enables a provider to verify whether the keywords searched by a data searcher are part of an authorized keyword set before executing the OKS protocol. This mechanism ensures that data receivers are restricted to retrieving only data containing specific, authorized keywords. However, due to the inherent unreliability of cloud servers, there is a risk that cloud data may not be updated promptly as required by the data sender. Specifically, when the data sender submits updated information to the cloud server, the server may fail to reflect these changes in a timely manner. As a result, when the data receiver performs a search, the cloud server might still return outdated data, posing a significant security risk regarding data accuracy and freshness. Existing schemes do not address the validation of cloud server data updates or guarantee that data recipients retrieve the most up-to-date information from the database. To resolve this, we implement both functions by utilizing a cryptographic accumulator to verify that the authorized message set $\{M_i\}_W$ is part of the total message set $\{M_i\}_X$.

If the data is stored locally, the relationship $\{M_i\}_W \subseteq \{M_i\}_X$ is evident. However, when the data sender transmits the ciphertext to the cloud server, there is a possibility that the cloud server may fail to update the ciphertext. In such cases, we can demonstrate that $\{M_i\}_W \nsubseteq \{M_i\}_X$ using cryptographic accumulators. Specifically, there are four possible scenarios: 1)If the authorized message set $\{M_i\}_W$ adds new data, but the total message set $\{M_i\}_X$ on the cloud is not updated, cryptographic accumulator validation will fail. 2)If the authorized message set $\{M_i\}_W$ deletes data, but the total message set $\{M_i\}_X$ on the cloud is not updated, cryptographic accumulator validation will fail. 3)If the authorized message set $\{M_i\}_W$ remains unchanged, but the total message set $\{M_i\}_X$ adds new data, and the updated set $\{M_i\}_X{}'$ is not reflected on the cloud, cryptographic accumulator validation will fail. 4)If the authorized message set $\{M_i\}_W$ remains unchanged, but the total message set $\{M_i\}_X$ deletes data, and the updated set $\{M_i\}_X{}'$ is not reflected on the cloud, cryptographic accumulator validation will fail.

These scenarios are designed to verify whether the data stored on the cloud is properly synchronized and updated, thus preventing the retrieval of outdated information. This ensures that data recipients always access the most current ciphertext from the cloud. In this paper, we address this issue by building upon the OKSA scheme and propose a new approach that achieves efficient verification in untrusted cloud environments. The basic model is illustrated in Figure 1.

In summary, our scheme involves three parties: the data sender, the cloud server, and the data receiver. The data sender extracts keywords x_i from plaintext M_i. Suppose a subset of plaintexts and their corresponding keyword set W are authorized for the data receiver. The sender encrypts all plaintexts and their keywords into ciphertexts C_i and uploads them to the cloud server. Using a cryptographic accumulator, the sender computes: An accumulator value acc for all keywords. A verification value v for the authorized keyword set W. Both acc and v are uploaded to the cloud server, while v and W are sent to the data receiver. The data receiver downloads ciphertexts and acc from the cloud server. Using v and acc, the receiver verifies the integrity and freshness of the cloud-stored data. If verification passes: The receiver selects a search keyword x^* from W, generates a token $K(x^*)$, and sends it to the cloud server. The cloud server verifies (without learning x^*) whether the token corresponds to an authorized keyword in W. Upon successful verification, the server converts the token into a search trapdoor and returns it to the receiver. The receiver uses the trapdoor to decrypt the ciphertexts associated with x^*. When updating data, the data sender: Modifies the corresponding ciphertexts on the cloud server. Recomputes the accumulator acc and verification value v for the updated dataset. Uploads the new acc and v to the cloud server. This scheme ensures secure keyword-based search with forward privacy, integrity verification, and efficient updates while maintaining confidentiality of unauthorized data.

A. Contribution

Our contribution is mainly three-fold.

1) In this paper, we propose a scheme for efficient verification in cloud servers that enhances oblivious keyword search with authorization by utilizing a cryptographic accumulator. The cryptographic accumulator is used to aggregate all messages and disclose the accumulated value, which is then used to generate the verification value for authorized messages. This approach enables the data sender to verify that the data stored on the cloud server is up to date. Additionally, the data sender can dynamically update the cloud data and efficiently modify both the accumulated value of the cryptographic accumulator and the corresponding witness value.
2) If the data receiver questions the validity of the results returned by the cloud service, they can verify the authenticity of the results with the assistance of the data sender. The data receiver can use the cryptographic accumulator to check whether the collection of authorized messages $\{M_i\}_W$ is a subset of the total collection of all messages $\{M_i\}_X$. This process ensures that the data receiver is retrieving information from the most up-to-date database. If the validation fails, it indicates that the server-side data is outdated.
3) We provide a formal security proof for the proposed scheme. Additionally, we conduct experimental simulations and compare the results with those of the

Authorized licensed use limited to: University of Greenwich. Downloaded on March 12,2025 at 20:42:48 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3550198

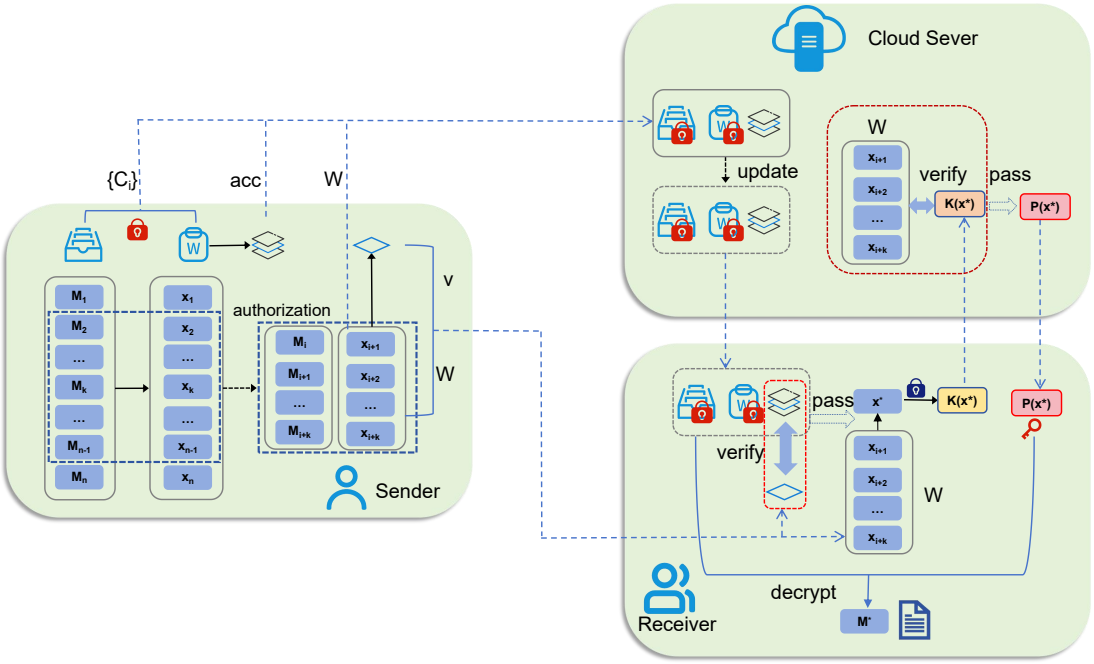

Fig. 1. The framework of Oblivious Keyword Search with Authorization and Verification for IoT Devices in Untrusted Cloud Environments

OKSA scheme. The analysis and experimental results demonstrate that our scheme is both efficient and well-suited to meet the practical requirements of the Internet of Things (IoT).

B. Related Works

Searchable encryption against keyword guessing attacks

Symmetric searchable encryption was first proposed by Song et al. [11] in 2000. The design of two-layer encryption allows the cloud server to securely scan document content and perform a search on the ciphertext. However, the need to traverse the document content results in search efficiency that is linearly related to the file size. Public key searchable encryption was introduced by Boneh et al. [12] in 2004. In this scheme, the user encrypts the data and keywords using a public key before uploading them to the cloud server. Subsequently, the user generates a search trapdoor using their private key. The cloud server receives the search trapdoor, executes a matching algorithm, and returns the ciphertext that successfully passes the match to the user as the search result. The user can then decrypt the ciphertext to retrieve the target data.

Ensuring keyword privacy is a critical concern in Public Key Encryption with Keyword Search (PEKS) and its variants. Boneh et al.'s scheme achieves semantic security against Chosen Keyword Attacks (CKA) [16] but requires a secure channel between the receiver and the server. The reliance on a secure channel limits the practical application of the scheme. When the channel between the receiver and the server is public, Byun et al. [13] pointed out that any adversary could infer the corresponding keyword by eavesdropping on the trapdoor, a type of attack known as External Keyword Guessing Attack (KGA). Moreover, deploying a secure channel between the server and receiver is often infeasible due to the high cost involved. Baek et al. [16] also highlighted the issue of secure channels in the context of external keyword guessing attacks. Specifically, in the absence of a secure channel, an external attacker can infer keyword information from the trapdoor by generating a search ciphertext. They eliminated the need for a secure channel, as well as the public and private keys of the application server, and introduced the concept of Public Key Searchable Encryption under a public channel, termed Secure Channel Free PEKS (SCF-PEKS).

Oblivious Keyword Search

Ogata and Kurosawa [14] proposed the concept of Oblivious Keyword Search (OKS) to address user privacy concerns in keyword search, utilizing an unintended transmission protocol between the provider and the user. The OKS scheme employs a blind signature method, where the ciphertext is generated using the provider's master private key and the selected keywords. The transmission of each trapdoor from the provider to the user requires the use of the master secret key and a user-generated keyword passphrase. Rhee et al. [17] introduced the Oblivious Joint Search scheme, employing boolean combinations of keywords. Freedman et al. [18] addressed privacy issues by utilizing pseudo-random functions derived from unintended evaluation values. Zhu and Bao et al. [19] developed an OKS scheme applicable to public databases, employing both linear and nonlinear unintended polynomial evaluations. Camenisch et al. [20] proposed an unintended keyword search method

Authorized licensed use limited to: University of Greenwich. Downloaded on March 12,2025 at 20:42:48 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3550198

based on public key encryption, constructing a mechanism that integrates Private Information Retrieval (PIR) attributes for public key encryption databases. However, this approach relies on the computationally intensive Zero-Knowledge Proof (ZKP) method. Notably, the aforementioned oblivious search schemes do not address the issue of authorization verification for the searched keywords. Additionally, the computational overhead associated with ZKPs results in less efficient schemes that are not directly applicable to cloud storage systems.

Peng Jiang et al. [15] proposed Oblivious Keyword Search with Authorization (OKSA), which incorporates authorization keyword validation based on unintentional search principles. During the transmission phase, the OKSA protocol requires only a single round of interaction between the provider and the user, achieving a communication cost complexity of $O(1)$. However, the untrustworthiness of cloud servers raises concerns, as they may fail to update cloud data in a timely manner as requested by the data sender.

Cryptographic Accumulator

Benaloh et al. [21] first introduced the concept of a cryptographic accumulator in 1993. However, their construction is a static accumulator, meaning that the set of accumulators is fixed. In 2002, Camenisch et al. [22] proposed the concept of a dynamic accumulator, which allows for the dynamic addition and deletion of elements within the set of accumulators. Nevertheless, both types of accumulators only support proofs of membership for elements in the set (i.e., $x_i \in X$) and do not provide proofs of non-membership (i.e., they cannot demonstrate that $y \notin X$). To address this limitation, Li et al. first proposed a universal accumulator construction in 2007, capable of facilitating both membership and non-membership proofs. Since the introduction of the cipher accumulator, extensive research has been conducted, leading to specific construction schemes. Based on various cryptographic tools, cryptographic accumulators can be classified into those based on the RSA mechanism [21]–[23], those based on bilinear mapping [24]–[27], and those based on the Merkle hash tree [28]. Cryptographic accumulators have a wide range of application scenarios. In access control systems, the access rights of authorized users are aggregated into accumulators, allowing these users to access the system using membership proofs as credentials. In anonymous credential systems [22], user identities must be further concealed, and combining accumulators with zero-knowledge proofs [29] provides an effective solution to this challenge. In remote data storage, cryptographic accumulators can function as Authenticated Data Structures (ADS) [30]–[32], where users utilize computation results and corresponding proofs to validate the correctness of the data. In cryptocurrencies, cryptographic accumulators reduce communication overhead and enhance authentication efficiency by replacing the Merkle hash tree [33].

Identified Verifiable Search Encryption Schemes

The aforementioned schemes are based on the assumption that cloud servers are inherently honest but curious entities. However, in reality, these servers may return incorrect or incomplete results due to external attacks or internal misconfigurations, potentially deceiving users. In response to these concerns, researchers have developed verifiable search encryption schemes to ensure the integrity and authenticity of search results [34]–[38] as a promising avenue for addressing the challenges posed by malicious servers. These schemes allow users to validate the retrieved results, enabling them to determine whether the server is acting legitimately or maliciously. However, the dataset is not a static environment. It is subject to constant updates and changes. As a result, researchers must address how to verify the results in such a dynamic environment, ensuring the integrity and accuracy of data even as it evolves over time. In the literature [39], the introduction of timestamps into the verification process was proposed as a solution for verifying documents in a dynamic environment. However, this approach has been shown to exceed linear time complexity. Subsequently, researchers developed a method for validating results by storing the Message Authentication Codes (MACs) corresponding to the indexes in a tree structure [22], with the root node serving as the validation point. However, each update requires recalculating all relevant nodes in the tree, which imposes a significant computational burden on the data owner. Furthermore, the aforementioned scheme is based on a single-user model, involving only two entities, the data owner and the server. In recent years, the combination of searchable encryption and the Internet of Things has become a research hotspot, and the more representative articles are as follows. Yang et al. [40] proposed an efficient and verifiable searchable encryption scheme named OpenSE, which utilizes a fuzzy polynomial evaluation protocol to hide access and search patterns in cloud-IoT environments. Sultan et al. [41] introduced a multi-client searchable encryption scheme for IoT environments, supporting dynamic updates, forward and backward privacy, and distributed storage to enhance security and privacy. Yang et al. [42] presented an offline/online attribute-based searchable encryption scheme based on ideal lattices, designed to improve the efficiency of encryption and search processes in IoT environments.

TABLE I
OUR SCHEME AND THE CURRENT SCHEMES.

Schemes	CKA	KGA	Authorization	Verification	Efficiency
Song [11]	✓	×	×	×	$O(n)$
Boneh [12]	✓	×	×	×	$O(n)$
Baek [16]	✓	✓	×	×	$O(n)$
Rhee [43]	✓	✓	×	×	$O(n)$
Ogata [14]	✓	✓	✓	✓	$O(n)$
Peng [15]	✓	✓	✓	×	$O(1)$
Ours	✓	✓	✓	✓	$O(1)$

"✓" indicates that corresponding requirements are supported; otherwise, "×" is used.

C. Organization

Section 2 presents essential foundational concepts, details the updatable and verifiable oblivious keyword search with authorization algorithm, outlines the security model, and discusses the underlying complexity assumptions. Section 3 introduces the protocol for the updatable and verifiable oblivious keyword search with authorization. Section 4 provides a formal security proof. Section 5 offers a comprehensive

Authorized licensed use limited to: University of Greenwich. Downloaded on March 12,2025 at 20:42:48 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3550198

TABLE II
NOTATIONS

Notation	Description
$\mathbb{N}$	The set of positive integers
λ	Safety parameter
(G, G_T, e, p, g)	Bilinear pair parameters parameter
H	Hash function
$\mathcal{S}$	A data sender
$\mathcal{C}$	The cloud server
$\mathcal{R}$	The receiver
X	Authorization keyword set
x^*	The search keyword
W	Authorized keyword set
$\{M_i\}_X$	A collection of all messages
$\{M_i\}_W$	A collection of authorized messages
(MPK, MSK)	The master public and secret key pair
(pk, sk)	The receiver's public and secret key pair
$(\mathrm{pk}_{\mathrm{acc}}, \mathrm{sk}_{\mathrm{acc}})$	The accumulator public and secret key pair
$\mathbf{K}(x^*)$	Token of $*$
$\mathbf{T}(x^*)$	Trapdoor cryptosystem of $*$
ℓ	The actual number of keywords retrieved
$acc\left(\{C_i\}_X\right)$	Total value of all ciphertexts
v	The witness of all authorized ciphertexts
$\mathcal{H}$	a hash function $\mathcal{H}: \{0,1\}^l \to \mathbb{Z}_p$

analysis and evaluation of the system's performance. The paper concludes in Section 6.

II. PRELIMINARIES

A. Notations

The symbols and terms used in this paper are efficiently compiled in Table II.

B. Bilinear pair

Bilinear pairing operations are widely utilized in public key cryptography to solve complex problems within elliptic curve groups. Essentially, a bilinear map is used to associate two group elements from an elliptic curve group with a third group element in a multiplicative group, while preserving isomorphic properties. This method transforms a specific problem instance from the elliptic curve group into a corresponding instance in the multiplicative group. A sub-exponential time algorithm is then applied to solve the problem in the multiplicative group, and the solution is subsequently used to address the original problem in the elliptic curve group. Cryptographic schemes that rely on bilinear pairings should be based on groups that support the identification of isomorphisms between elliptic curve groups and multiplicative groups. Generally, for the bilinear map $e: G_1 \times G_2 \to G_T$, the following conditions must be satisfied:

(1) Non-degenerate: For generators g_1, g_2 in G_1 and $g_1, g_2, e(g_1, g_2)$ is the generator of G_T ;

(2) Bilinear: For any $u \in G_1, v \in G_2, a, b \in \mathbb{Z}_p$, there is $e\left(u^a, v^b\right) = e(u, v)^{ab}$;

(3) Computability: For any $u \in G_1, v \in G_2$, there is a polynomial time algorithm that can calculate $e(u, v)$.

C. Cryptographic Accumulator

Cryptographic accumulators are capable of compressing all elements in a set into a concise value, while generating a proof for each element to demonstrate that it belongs to the set. Any participating entity with access to publicly available information can verify that the elements it possesses are part of the set represented by the accumulator value.

Nguyen et al. [24] proposed a bilinear mapping accumulator based on the $q-SDH$ assumption, where the number of elements to accumulate is limited to q. We extend the definitions of accumulators provided in [22], [44] as follows:

(1) $\mathrm{Gen}\left(1^\lambda\right)$: The accumulator administrator generates public-private key pairs. Input 1^λ, generate $\mathrm{pk}_{\mathrm{acc}} = \left(g, g^s, \cdots, g^{s^q}, u\right), \mathrm{sk}_{\mathrm{acc}} = s \in {}_R Z_p^*$, where q is the upper bound of the cumulative element, the value domain of the cumulative element is $Z_p \backslash \{-s\}$, and u is randomly generated by Z_p^*.

(2)$\mathrm{Eval}\left(X, \mathrm{pk}_{acc}\right)$: The accumulator administrator computes the cumulative value acc_X. Input set $X = \{x_1, \cdots, x_n\} \subset Z_p \backslash \{-s\}$, where $n \leqslant q$, computes the accumulator $\mathrm{acc}_X = g^{u\Pi_{x \in X}(x+s)}$ using pk_{acc} without the private key s.

(3)$WitCreate\left(\mathrm{pk}_{\mathrm{acc}}, x_i, \mathrm{acc}_X, X\right)$: The accumulator administrator creates a proof for user x_i. Enter the public key pk_{acc}, the accumulator value acc_X, the set X, and the element x_i, and create the witness for $x_i \in X$ as $w_i = g^{u\prod_{x \in X \backslash \{x_i\}}^{(x+s)}}$.

(4)$\mathrm{VerMem}\left(\mathrm{acc}_X, x_i, w_i\right)$: The verifier checks whether x_i, is in the accumulator. By checking if $\left(w_i, g^{x_i} g^s\right) = e\left(\mathrm{acc}_X, g\right)$. If it holds, output 1, otherwise output 0.

(5) $\mathrm{Add}\left(acc_X, x, \mathrm{sk}_{acc}\right)$: Add element $x \in X$ to the accumulator, updated accumulator $acc_{X'} = acc_{X \cup \{x\}} = acc_X^{x+s}$.

(6) $\mathrm{Del}\left(\mathrm{acc}_X, x, \mathrm{sk}_{acc}\right)$: If the value x is removed from the accumulator, the updated accumulator $acc'_X = acc_{X \backslash \{x\}} = acc_X^{1/(x+s)}$.

(7)$\mathrm{MemWitUp}\left(x, w_i, x_i, \mathrm{pk}_{acc}, acc_X, acc_{X'}\right)$: The user updates the evidence of the element x_i. When the element x is added to the accumulator, the proof $w'_i = acc_X w_i^{x-x_i}$ of x is updated. This update operation is correct because $acc_X w_i^{x-x_i} = w_i^{x_i+s} w_i^{x-x_i} = w_i^{x+s} = w'_i$. When the element x is deleted, where $x \neq x_i \in X$, the proof for updating x_i is correctly computed by calculating $w'_i = w_i^{1/(x-x_i)} acc_{X'}^{1/(x_i-x)}$.

It is important to note that in the previously mentioned construction, the size of the public key is linearly dependent on the upper bound q of the cumulative elements. As a result, if q is large, the size of the public key will also increase significantly.

D. Algorithm Definition

The updatable and verifiable oblivious keyword search with authorization interacts with three parties: the data sender, the cloud server, and the data receiver.

Setup

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. $\mathcal{S}$ creates a set of all keywords X that the size of X does not exceed n. Furthermore, in collaboration with each user, $\mathcal{S}$ establishes a distinct keyword set W, ensuring that the size

Authorized licensed use limited to: University of Greenwich. Downloaded on March 12,2025 at 20:42:48 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3550198

of W does not exceed k. And send the accumulation value $acc\left(\{M_i\}_X\right)$ of all the messages $\{M_i\}_X$ to all of the users.

Commit

$\mathcal{S}$ processes a message m_i, a designated keyword x_i, and the MPK to produce the corresponding encrypted data $\{C_i\}$. Subsequently, $\mathcal{S}$ securely transmits the collection of ciphertexts $\{C_i\}$ to the cloud storage $\mathcal{C}$.

Transfer

$\mathcal{S} \rightarrow \mathcal{R}$: It is assumed that $\mathcal{S}$ establishes a authorized keyword set W in negotiations with each receiver $\mathcal{R}$. $\mathcal{S}$ calculates the witness for $\{M_i\}_W \subseteq \{M_i\}_X$ as υ and sends it to $\mathcal{R}$.

$\mathcal{R} \rightarrow \mathcal{C}$: At first, $\mathcal{R}$ verify that the data provided by the server is the latest data through the accumulation value $acc\left(\{M_i\}_X\right)$ and υ. Otherwise, it results in $\perp$. If the verification passes, $\mathcal{R}$ inputs the designated keyword set W, including specific keyword x_i. With the main public key MPK and the user's private key sk as input, the key password $\mathbf{K}\left(x_i\right)$ and proof information are output Σ. Then $\mathcal{R}$ sends $\left(\mathbf{K}\left(x_i\right), \Sigma\right)$ to $\mathcal{C}$. Here, $\mathbf{K}\left(x_i\right)$ is calculated from sk, x_i, W, MPK. Σ helps $\mathcal{C}$ verify that the received token can only generate a search trap for the authorized keyword.

$\mathcal{C} \rightarrow \mathcal{R}$: $\mathcal{C}$ inputs the token $\mathbf{K}\left(x_i\right), W$, and the MSK to affirm accountability, verifying that $\left|\mathbf{K}\left(x_i\right)\right| = 1$. It then generates a trapdoor T for $\mathcal{R}$. And send the accumulation value $acc\left(\{M_i\}_X\right)$ to $\mathcal{R}$.

$\mathcal{R}$: The data receiver $\mathcal{R}$ downloads all the ciphertext data $\{C_i\}$ from the cloud server, and upon receiving inputs T, sk will yield m_i; otherwise, it results in $\perp$.

If the data sender $\mathcal{R}$ needs to update the data, just update the data on the cloud, and then generate the updated $acc\left(\{M_i\}_X\right)'$ and witness υ', and send it to the data recipient $\mathcal{R}$.

Correctness

For the updatable and verifiable oblivious keyword search with authorization to be deemed effective, it is essential that the recipient accurately retrieves the specified message, assuming all involved parties adhere to the outlined protocol. Furthermore, the authenticity is validated when accountability checks confirm that the trapdoor generated from the acquired token corresponds exclusively to specific keywords. It is also crucial that these keywords are within the scope of the authorized keyword collection.

E. Security Notions

Based on the previous article [24], [45], [46], our framework establishes four key security criteria: confidentiality for the receiver, indistinguishability, accountability and collision resistance for the dynamic accumulator $\mathcal{D}$. The first aspect, user confidentiality, ensures that during the $i-th$ transaction phase. The sender, denoted as $\mathcal{S}$, is not possible to infer the search keywords from the users token. The principle of indistinguishability serves as a protective measure, preventing a potentially adversarial receiver, denoted as $\mathcal{R}$, from deciphering both the message and keyword embedded within the ciphertext. The accountability aspect ensures that each user's trapdoor request is uniquely linked to a specific keyword within the authorized keyword set. Additionally, the collision resistance of the dynamic accumulator guarantees the correct addition and validation of elements, preventing any potential conflicts or inconsistencies.

Receiver Privacy

For a receiver, discerning whether x is x_0 or x_1 becomes challenging when presented with $(\mathbf{K}(x), \Sigma)$ 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 $(\mathbf{K}(W), W, sk)$ meets the condition $|W| > 1$, crafting $(\mathbf{K}(W), \Sigma)$ 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 $\mathcal{A}$.

Collision Resistance

Collision Resistance is the difficulty of finding two different inputs such that they add up to the same value in a given accumulator. For a valid accumulator $\mathcal{D}$, we want that for any set of inputs X and Y, if $X \neq Y$, then $\mathcal{D}(X) \neq \mathcal{D}(Y)$.

Defintion 1. *An accumulator $\mathcal{D}$ is said to be collision resistant if there is no valid attacker $\mathcal{A}$, such that: $Pr[\mathcal{A}(\mathcal{D}(X), \mathcal{D}(Y)) = 1] \geq \epsilon$. Here, $X \neq Y$ and ϵ is a negligible quantity representing the probability of a successful attack.*

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 $(\mathbf{K}(W), \Sigma)$.

Guess.

$\mathcal{A}$ attempts to guess by announcing θ' and succeeds if θ' matches θ. The advantage of $\mathcal{A}$ is quantified as $\mathrm{Adv} = \left|\Pr\left[\theta' = \theta\right] - 1/2\right|$.

Theorem 1. *The updatable and verifiable oblivious keyword search with authorization framework is considered to meet user privacy standards when it is computationally infeasible for any adversary, operating within probabilistic polynomial time, to gain a significant advantage in the specified user privacy context.*

2) Indistinguishability

Setup.

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

First Stage.

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

Authorized licensed use limited to: University of Greenwich. Downloaded on March 12,2025 at 20:42:48 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3550198

Challenge.

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

Second Stage.

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

Guess.

$\mathcal{A}$ proposes a guess of θ' and is victorious if it matches θ. The advantage of $\mathcal{A}$ is defined as $\text{Adv} = |\Pr[\theta' = \theta] - 1/2|$.

Theorem 2. *The updatable and verifiable oblivious keyword search with authorization achieves indistinguishability against chosen keyword attacks if no adversary, employing probabilistic polynomial time approaches, can significantly succeed in the described game.*

3) Accountability

In the updatable and verifiable oblivious keyword search with authorization framework, the core of accountability verification lies in ensuring that each trapdoor is uniquely associated with a single authorized keyword. This process counteracts scenarios where an adversary, denoted as $\mathcal{A}$, might construct a legitimate keyword token $\mathbf{K}(W')$, with W' being a fraction of the endorsed keyword set W, maintaining that $1 < |W'| < |W| \leq n$, indicating $\mathcal{A}$'s awareness of W', W, and the secret key sk used in the token's formation.

Setup.

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

Challenge.

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

Win.

The adversary $\mathcal{A}$ submits $(\mathbf{K}(W'), \Sigma)$ and is deemed victorious if this submission successfully clears the verification process. The edge of $\mathcal{A}$ in this scenario is determined as Adv in the formulation of $(\mathbf{K}(W'), \Sigma)$.

Theorem 3. *Accountability within the updatable and verifiable oblivious keyword search with authorization framework is established when no adversary, operating within polynomial time bounds, can succeed in the previously outlined game with a significant advantage.*

4) Collision Resistance

Theorem 3 shows that the collision resistant property of $\mathcal{D}$ which is based on the Strong Diffie Hellman assumption is as follows. Its proof is given in [24].

Theorem 4. *The accumulator $\mathcal{D}$ ensures Collision Resistance if the $q - SDH$ assumption not to be breached, furthermore the accumulator accumulated the number of elements is not more than q.*

F. Assumptions

The foundational security constructs of the updatable and verifiable oblivious keyword search with authorization 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 [47], here we provide only a brief overview, directing readers to these references for a thorough exploration of its complexities.

Theorem 5. *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 $\left(f(x), h^{f(a)}\right)$ 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, 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 [48]. Detailed intractability analysis of this problem will be presented later. For more details, please refer to the paper[12].

Since the $q - SDH$ Problem has been proposed and discussed in [24], we only give its expression. The $q - SDH$ assumption stems from a weaker assumption introduced by Mitsunari et al. [49] in constructing a traitor tracking scheme, and was later well articulated by Boneh and Boyen [50]. Intuitively, it implies that no PPT algorithm can compute a pair $\left(c, \frac{1}{s+c}P\right)$, where $c \in \mathbb{Z}_p$, from a tuple $(P, sP, ...)$, where s is a tuple $(P, sP, ...)$, and where s is a tuple $(P, sP, \ldots, s^qP)$, where $s \in_R \mathbb{Z}_p^*$.

Theorem 6. *$q-$**Strong Diffie-Hellman** ($q - SDH$) **Assumption**. For every PPT algorithm $\mathcal{A}$, the following function $Adv_{\mathcal{A}}^{q-SDH}(l)$ is negligible.*

$$Adv_{\mathcal{A}}^{q-SDH}(l) = \Pr[(\mathcal{A}(\mathbf{t}, P, sP, \ldots, s^qP) = (c, \frac{1}{s+c}P)) \wedge (c \in \mathbb{Z}_p)].$$

where $\mathbf{t} = (p, \mathbb{G}_1, \mathbb{G}_M, e, P) \leftarrow \mathcal{G}\left(1^l\right)$ and $s \leftarrow \mathbb{Z}_p^*$.

III. The Updatable and Verifiable Oblivious Keyword Search with Authorization Algorithm

Our protocol engages three entities: a data sender $\mathcal{S}$, a cloud server $\mathcal{C}$, a receiver $\mathcal{R}$, consiting of three distinct phases: **Setup, Commit, and Transfer**.

Setup

Entity $\mathcal{S}$ begins by accepting a security parameter denoted as 1^λ, an integer value n , and the group system $\mathcal{PG} = (p, \mathbb{G}, \mathbb{G}_T, e)$. Following this, $\mathcal{S}$ selects generator elements g, h from $\mathbb{G}$, and chooses random numbers α from $\mathbb{Z}_p$. It proceeds to calculate g^α and $h_i = h^{\alpha^{i-1}}$ for $i = 1, 2, \cdots, n+1$. A cryptographic one-way hash function H mapping from $(\{0,1\}, \mathbb{G}_T)$ to $\{0,1\}^\ell$ is also selected. Let $\mathcal{H}$ be a collision-

Authorized licensed use limited to: University of Greenwich. Downloaded on March 12,2025 at 20:42:48 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3550198

free hash function $\mathcal{H} : \{0,1\}^l \to \mathbb{Z}_p$. The ensuing steps involve the generation of the master key pair are

$$MPK = (\mathcal{PG}, H, g, g^{\alpha}, h_1, h_2, \cdots, h_{n+1}), MSK = \alpha.$$
$$\mathrm{pk_{acc}} = \left(g, g^{s}, \cdots, g^{s^q}, u\right), \mathrm{sk_{acc}} = s \in {}_R Z_p^*.$$

$\mathcal{S}$ compute the cumulative value acc_X, $\mathcal{H}(m_i) = M_i$.

$$acc(\{M_i\}_X) = g^{u\prod_{M_i \in \{M_i\}_X}(M_i+s)}. \tag{1}$$

$\{M_i\}_X$ is the set of all messages, $\{M_i\}_X = \{M_1, \cdots, M_n\} \subset Z_p \backslash \{-s\}$. Subsequently, $\mathcal{S}$ securely transmits the collection of ciphertexts $\{M_i\}_X$ and $acc(\{M_i\}_X)$ to the cloud server $\mathcal{C}$. Where q is the upper bound of the cumulative element whose range is $Z_p \backslash \{-s\}$, u is randomly generated by Z_p^*. Entity $\mathcal{S}$ makes the MPK, $\mathrm{pk_{acc}}$ and $acc(\{M_i\}_X)$ available to everyone and maintains the confidentiality of MSK and $\mathrm{sk_{acc}}$.

Commit

Within the framework, the complete keyword space is represented by $\mathcal{KS}$, encompassing a total of n 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 from $\mathcal{KS}$, the entity $\mathcal{S}$ selects a random value r_i from ${}_R\mathbb{Z}_p$.

$$\begin{aligned} C_i = (c_{1i} = g^{r_i(\alpha+x_i)}, c_{2i} = H(0, e(g,h)^{r_i}), \\ c_{3i} = H(1, e(g,h)^{r_i}) \oplus m_i). \end{aligned} \tag{2}$$

Transfer

We assume that $\mathcal{S}$ identify a particular keyword set W with each authorized user, where $W \subseteq X$ and $|W|$ is denoted as k not more than n.

$\mathcal{S} \to \mathcal{R}$: let $v = (\{M_i\}_W, \{M_i\}_X)$ be the witness of $\{M_i\}_W \subseteq \{M_i\}_X$.

$$v = g^{u\prod_{M_i \in \{M_i\}_X \backslash \{M_i\}_W}(M_i+s)}. \tag{3}$$

$\mathcal{R} \to \mathcal{C}$: Determine whether the following equation is true by verifying $\{M_i\}_W \subseteq \{M_i\}_X$ whether it is true.

$$e\left(v, g^{\prod_{M_i \in \{M_i\}_W}(M_i+s)}\right) = e(\mathrm{acc}(\{M_i\}_X), g). \tag{4}$$

If the equation is not satisfied, it indicates that the cloud server has failed to update the data in a timely manner, and the program will be terminated. If the equation holds, the following program will be executed.

With the predefined keyword set W, and specific keywords $\{x_1, x_2 \ldots x_k\} \in W$ alongside the MPK, the receiver $\mathcal{R}$ chooses a random number t from ${}_R\mathbb{Z}_p$. The receiver $\mathcal{R}$ let $sk = t$ and formulates tokens $\mathbf{K}(x_i)$, where x_i corresponding to the chosen keyword for which $\mathcal{R}$ is seeking for.

$$\mathbf{K}(x_i) = h^{t\prod_{x_j \in W, j \neq i}(\alpha+x_j)}. \tag{5}$$
$$\Sigma = \left(\Sigma_1 = h^{\frac{\alpha+x_i}{t}}, \Sigma_2 = \Sigma_1^{\alpha^{n-1}}\right). \tag{6}$$

Then $\mathcal{R}$ sends $(\mathbf{K}(x_i), \Sigma)$ to $\mathcal{C}$.

$\mathcal{C} \to \mathcal{R}$: Given the tuple $(\mathbf{K}(x_i), W, \Sigma)$, with the MPK in hand, $\mathcal{C}$ proceeds to verify accountability using the subsequent equations:

$$e(\Sigma_2, h^{\alpha}) = e\left(\Sigma_1, h^{\alpha^n}\right). \tag{7}$$
$$e\left(h, h^{\Pi_{x_i \in W}(\alpha+x_i)}\right) = e(\mathbf{K}(x_i), \Sigma_1). \tag{8}$$

When both equations are satisfied, it confirms that the tokens $\mathbf{K}(x_i)$ is associated with the authorized set of keywords, indicated as:

$$|\mathbf{K}(x_i)| = 1. \tag{9}$$

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 trapdoor T as

$$T = \mathbf{K}(x_i)^{\frac{1}{\Pi_{x_j \in W(\alpha+x_j)}}}. \tag{10}$$

Then $\mathcal{C}$ returns the trapdoor T to $\mathcal{R}$.

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

$$c_{2i} = H\left(0, e(c_{1i}, T)^{\frac{1}{t}}\right). \tag{11}$$

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

$$m_i = c_{3i} \oplus H\left(1, e(c_{1i}, T)^{\frac{1}{t}}\right). \tag{12}$$

Update

If the data sender adds the element $M_i \in \{M_i\}_X$ to the accumulator, the updated accumulator is:

$$\begin{aligned} acc\left(\{M_i\}'_X\right) &= acc\left(\{M_i\}_X \cup \{M_i\}\right) \\ &= acc\left(\{M_i\}_X\right)^{M_i+s}. \end{aligned} \tag{13}$$

When the value M_i is removed from the accumulator, the new accumulator is

$$\begin{aligned} acc\left(\{M_i\}'_X\right) &= acc\left(\{M_i\}_X \setminus \{M_i\}\right) \\ &= acc\left(\{M_i\}_X\right)^{1/(M_i+s)}. \end{aligned} \tag{14}$$

The data sender updates the authorization set witness, and when adding the element M_i to the accumulator, updates the authorization set witness if M_i is not in the authorization messages set

$$v' = v^{M_i+s}. \tag{15}$$

If M_i is in the authorization message set, the authorization set witness does not change. When the element M_i is deleted, if M_i is not in the authorization ciphertext set, the evidence of the authorization set is updated

$$v' = v^{1/(M_i+s)}. \tag{16}$$

If M_i is in the authorization ciphertext set, the authorization set evidence remains unchanged.

Correctness

Authorized licensed use limited to: University of Greenwich. Downloaded on March 12,2025 at 20:42:48 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3550198

With the (MPK, MSK) obtained from the **Setup**, and $(\mathbf{K}(x_i), \Sigma)$ at hand, the verification of accountability's correctness hinges on the fulfillment of the following expression.

$$\begin{aligned} &H\left(0, e\left(c_{1i}, T\right)^{\frac{1}{t}}\right) = H\left(0, e\left(g^{r_i(\alpha+x_i)}, h^{\frac{t}{\alpha+x_i}}\right)^{\frac{1}{t}}\right) \\ &= H\left(0, e(g,h)^{r_i}\right) = c_{2i}. \end{aligned} \tag{17}$$

$$\begin{aligned} &c_{3i} \oplus H\left(1, e\left(c_{1i}, T\right)^{\frac{1}{t}}\right) \\ &= H\left(1, e(g,h)^{r_i}\right) \oplus m_i \oplus H\left(1, e\left(g^{r_i(\alpha+x_i)}, h^{\frac{t}{\alpha+x_i}}\right)^{\frac{1}{t}}\right) \\ &= H\left(1, e(g,h)^{r_i}\right) \oplus m_i \oplus H\left(1, e(g,h)^{r_i}\right) = m_i. \end{aligned} \tag{18}$$

With the ciphertext C_i obtained through the execution of the **Commit** algorithm, coupled with the trapdoor T generated via **Transfer**, and the user's sk, the process to validate the accuracy of the scheme's function involves

$$e\left(\Sigma_2^{(2)}, h^{\alpha}\right) = e\left({\Sigma_1}^{(2)^{\alpha^{n-1}}}, h^{\alpha}\right) = e\left(\Sigma_1^{(2)}, h^{\alpha^n}\right). \tag{19}$$

The correctness procedure of the accumulator verification is

$$\begin{aligned} &e\left(v, g^{\prod_{M_i \in \{M_i\}_W}(M_i+s)}\right) \\ &= e\left(g^{u\prod_{M_i \in \{M_i\}_X \setminus \{M_i\}_W}(M_i+s)}, g^{\prod_{M_i \in \{M_i\}_W}(M_i+s)}\right) \\ &= e(g^{u\prod_{M_i \in \{M_i\}_X}(M_i+s)}, g) = e(\mathrm{acc}(\{M_i\}_X), g). \end{aligned} \tag{20}$$

It can be seen from the above calculation that the updated data also satisfies the equation.

IV. Security Analysis

Our analysis focuses on the security aspects of the updatable and verifiable oblivious keyword search with authorization protocol, in line with the framework presented in Section 2.4. This security reduction on account of the complex problems defined in Section 2.6. A detailed and systematic proof process is provided in the following discussion.

A. Receiver Privacy

Theorem 7. *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. Consider W as the set of authorized keywords, and the pair $(\mathbf{K}(W), \Sigma)$ derived from the condition $x = x_0$. The resulting formulation of the keyword token and its corresponding proof is presented as follows:

$$\mathbf{K}(W) = h^{t\prod_{x_j \in W, x_j \neq x_0}(\alpha+x_j)}. \tag{21}$$

$$\Sigma = \left(\Sigma_1 = h^{\frac{\alpha+x_0}{t}}, \Sigma_2 = h^{\frac{\alpha^{n-1}(\alpha+x_0)}{t}}\right). \tag{22}$$

In the case of a unique keyword x_1, and with t' chosen from $\mathbb{Z}_p$, we adopt the implicit assignment $t' = t \cdot \frac{\alpha+x_1}{\alpha+x_0}$. 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:

$$\begin{aligned} \mathbf{K}(x_0) &= h^{t\prod_{x_j \in W, x_j \neq x_0}(\alpha+x_j)} \\ &= h^{t'\prod_{x_j \in W, x_j \neq x_1}(\alpha+x_j)} = \mathbf{K}(x_1). \end{aligned} \tag{23}$$

Let's consider Σ' to be composed of (Σ_1', Σ_2'). In this scenario, the accountability proofs are found to be congruent, signifying Σ_1 is equal to Σ_1' and Σ_2 is equal to Σ_2'. This equivalency can be substantiated through the following validation:

$$\Sigma_1 = h^{\frac{\alpha+x_0}{t}} = h^{\frac{\alpha+x_1}{t'}} = \Sigma_1' \tag{24}$$

$$\Sigma_2 = h^{\frac{\alpha^{n-1}(\alpha+x_0)}{t}} = h^{\frac{\alpha^{n-1}(\alpha+x_1)}{t'}} = \Sigma_2'. \tag{25}$$

The equivalence $(\mathbf{K}(x_0), \Sigma) = (\mathbf{K}(x_1), \Sigma')$ holds true. Given that t is selected at random from $\mathbb{Z}_p$, it follows that t' possesses uniform randomness within $\mathbb{Z}_p$ as well. Consequently, the distributions of $(\mathbf{K}(W), \Sigma)$ corresponding to both x_0 and x_1 are indistinguishable, offering no leverage to an adversary $\mathcal{A}$ in predicting the keyword encapsulated in $\mathbf{K}(W)$.

B. Indistinguishability

Theorem 8. *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.*

Proof. Imagine an adversary, referred to as $\mathcal{A}$, capable of undermining this indistinguishability. Under this hypothesis, we can devise an algorithm, named $\mathcal{B}$, 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 $\mathcal{B}$ 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 $\mathcal{B}$ and $\mathcal{A}$ is elaborated in the following explanation. □

Setup

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

$$f(\alpha) = \alpha + x_\theta. \tag{26}$$

$$q(\alpha) = \prod_{\substack{x_j \in \mathcal{KS} \\ x_j \neq x_\theta}} (\alpha + x_j). \tag{27}$$

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

$$MPK = (g_0, g_0^{\alpha}, h_1, h_2, \cdots, h_{n+1}, \mathcal{PG}). \tag{28}$$

H-Query

The entity $\mathcal{B}$ operates a hash list designated as $L\left(a_i, X_i, h^i\right)$, 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 $\mathcal{B}$ provides the associated h^i back to $\mathcal{A}$. In

Authorized licensed use limited to: University of Greenwich. Downloaded on March 12,2025 at 20:42:48 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3550198

cases where (a_i, X_i) is not present in L, $\mathcal{B}$ determines the hash value h^i using the following approach.

$$h^i = H\left(a_i, X_i\right) = \begin{cases} b_0^i, & \text{if } a_i = 0, \\ b_1^i, & \text{if } a_i = 1. \end{cases} \tag{29}$$

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

Phase 1

In this phase, $\mathcal{A}$ selects a set of keywords W, confined within the bounds of $\mathcal{KS}$, and limited to a maximum size of $|W|$ not more than n. For a trapdoor request corresponding to any keyword x_i from W,$\mathcal{A}$ opts for a random $t \in \mathbb{Z}_p$ as $sk = t$, and communicates (x_i, t) to $\mathcal{B}$. Should x_i matches x_θ, the process is terminated. In contrast, If x_i differs from x_θ, $\mathcal{B}$ delivers $T = h_0^{tq_i(\alpha)f(\alpha)}$ to $\mathcal{A}$, with $q_i(\alpha)$ defined as $\frac{q(\alpha)}{\alpha + x_i}$. Subsequent steps involve verifying the trapdoor.

$$\begin{aligned} T &= \mathbf{K}\left(x_i\right)^{\frac{1}{\prod_{x_j \in W}(\alpha + x_j)}} = h^{\frac{t\prod_{x_j \in W, j \neq i}(\alpha + x_j)}{\prod_{x_j \in W}(\alpha + x_j)}} \\ &= h^{\frac{tf(\alpha)q(\alpha)}{\alpha + x_i}} = h_0^{tq_i(\alpha)f(\alpha)}. \end{aligned} \tag{30}$$

It becomes apparent that the expression $h_0^{q_i(\alpha)f(\alpha)}$ It becomes apparent that the expression $h_0^{f(\alpha)}, h_0^{\alpha f(\alpha)}$, and so on, up to $h_0^{\alpha^{n-2} f(\alpha)}$ as provided in the instance.

Challenge

When $\mathcal{A}$ presents the pairs (m_0, x_0) and (m_1, x_1) as a challenge to $\mathcal{B}$, 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, $\mathcal{B}$ verifies the presence of $(0, Z)$ and $(1, Z)$ within the list L. If found, $\mathcal{B}$ retrieves and labels the corresponding hash values as b_0^* and b_1^*. In the absence of these entries, $\mathcal{B}$ randomly generates b_0^* and b_1^* from the binary set $\{0,1\}^\ell$ and proceeds to set them accordingly.

$$H(0, Z) = b_0^*, H(1, Z) = b_1^*. \tag{31}$$

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

$$C^* = \left(c_1 = g_0^r, c_2 = b_0^*, c_3 = b_1^* \oplus m_\theta\right). \tag{32}$$

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.

$$c_1 = g^{r_i(\alpha + x_\theta)} = g_0^{r_i f(\alpha)} = g_0^r. \tag{33}$$

$$\begin{aligned} c_2 &= H\left(0, e(g,h)^{r_i}\right) \\ &= H\left(0, e\left(g_0, h_0\right)^{rq(\alpha)}\right) = H(0, Z) = b_0^*. \end{aligned} \tag{34}$$

$$\begin{aligned} c_3 &= H\left(1, e(g,h)^{r_i}\right) \oplus m_\theta \\ &= H\left(1, e\left(g_0, h_0\right)^{rq(\alpha)}\right) \oplus m_\theta \\ &= H(1, Z) \oplus m_\theta = b_1^* \oplus m_\theta. \end{aligned} \tag{35}$$

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 $\mathcal{A}$, C^* emerges as a random ciphertext.

Phase 2

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

Guess

The adversary $\mathcal{A}$ 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 $\mathcal{B}$ in unraveling the intricate hard problem. Lets 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 $\mathcal{B}$ persevering without an abortive outcome is delineated in the following sections.

$$\begin{aligned} \Pr[abort] &= \left(1 - \frac{1}{n}\right)\left(1 - \frac{1}{n-1}\right)\cdots\left(1 - \frac{1}{n - q_T + 1}\right) \\ &= \frac{n - q_T}{n}. \end{aligned} \tag{36}$$

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 $\frac{2}{n}$. Assuming $\mathcal{A}$'s success in compromising the security game has a value of at least ϵ, the reduction in advantage can be represented as:

$$\begin{aligned} \epsilon_{\text{reduction}} = \Pr\left[\theta' = \theta \mid Z = e(g_0, h_0)^{rq(\alpha)}\right] - \\ \Pr[\theta' = \theta \mid Z \text{ is an arbitrary element}] = \epsilon. \end{aligned} \tag{37}$$

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

C. Accountability

Theorem 9. *The framework's competence in ensuring accountability is validated within the Accountability game, contingent on the intractability of the $(f, n) - DHE$ Problem [51].*

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

Setup

Algorithm $\mathcal{B}$ 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, $\mathcal{B}$ selects H akin to the actual scheme . Then $\mathcal{B}$ circulates the MPK for $\mathcal{A}$,

$$MPK = \left(g, g^a, h_1, h_2, \cdots, h_{n+1}, H, \mathcal{PG}\right). \tag{38}$$

Challenge

Authorized licensed use limited to: University of Greenwich. Downloaded on March 12,2025 at 20:42:48 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3550198

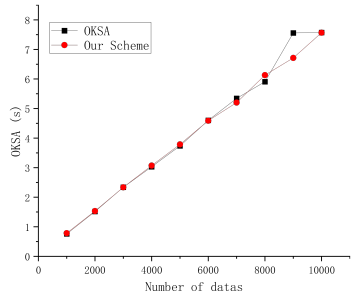

Fig. 2. Setup time

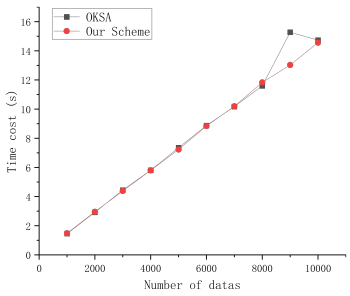

Fig. 3. Commit time

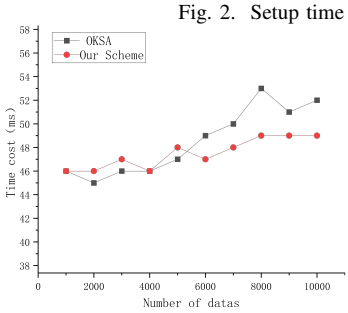

Fig. 4. Transfer time

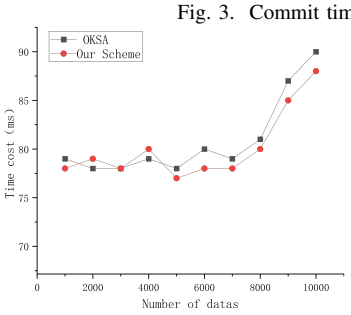

Fig. 5. Generate Trapdoor time

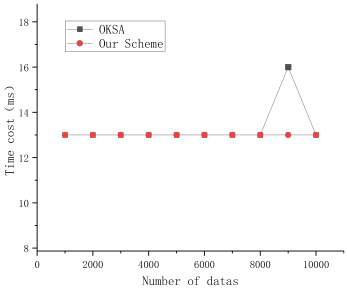

Fig. 6. Get message time

$\mathcal{A}$ selects two unique keyword sets, W and W', ensuring $|W'| > 1$ and $|W| \leq n$. A secret key sk is then established by randomly choosing a value t from $\mathbb{Z}_p$, with $sk = t$. Following this, $\mathcal{A}$ proposes $(\mathbf{K}(W'), W, W', sk)$ along with a challenge indication set to 1. Here, the token $\mathbf{K}$ is characterized as

$$\mathbf{K}(W') = h^{t \prod_{x_j \in W - W'}(a+x_j)}. \tag{39}$$

Win

The adversary $\mathcal{A}$ selects two keyword sets, W and W', conforming to the constraints $|W'| > 1$ and $|W| \leq n$. Upon choosing a random secret key t from $\mathbb{Z}_p$, denoted as $sk = t$, $\mathcal{A}$ then formulates and submits $(\mathbf{K}(W'), W, W', sk)$ with a challenge indicator set to 1. The token within this setup is characterized as

$$\begin{aligned} \Sigma = (\Sigma_1 &= h^{\frac{1}{t} \prod_{x_j \in W'}(a+x_j)}, \\ \Sigma_2 &= \Sigma_1^{a^{n-1}} = h^{\frac{1}{t} a^{n-1} \prod_{x_j \in W'}(a+x_j)}). \end{aligned} \tag{40}$$

Consequently, both the token and its associated proof are capable of successfully undergoing the matching process, as described below.

$$e(\Sigma_2, h^a) = e(\Sigma_1, h^{a^n}). \tag{41}$$

$$e\big(h, h^{\prod_{x_i \in W}(a+x_i)}\big) = e\big(\mathbf{K}(W'), \Sigma_1\big). \tag{42}$$

Let $f(x) = \frac{1}{t} x^{n-1} \prod_{x_j \in W'}(x + x_j)$, then $\Sigma = h^{f(a)}$. Given that the function $f(x)$ is classified as a polynomial with a degree, $\deg f(x)$, surpassing n, it follows that $\mathcal{B}$ furnishes the pair $(f(x), \Sigma)$ 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 W' is singular, $|W'| = 1$, and correspondingly, the token $\mathbf{K}(W')$ also possesses a singular count, $|\mathbf{K}(W')| = 1$.

D. Collision Resistance

Theorem 10. *The accumulator $\mathcal{D}$ ensures collision resistance if the $q - SDH$ assumption is satisfied, where q denotes the maximum number of elements that can be accumulated.*

Due to the previously proposed and analyzed collision resistance problem in references [16], we will provide only a description here and omit the analysis. For further details, the reader is referred to the corresponding references.

V. Performance Evalution

We evaluated the performance of the proposed scheme through a series of experiments. The experiments were conducted on an Ubuntu 22.04.3 desktop (AMD64) operating system, with the following hardware specifications: an 11th Gen Intel (R) Core i5-1135G7 CPU @2.40GHz and 40.0 G of RAM. We selected the OKSA scheme for a comparative experiment, as it is the closest to the proposed scheme in terms of functionality. The comparison focuses on the time consumption of the two schemes across various algorithms, including the setup algorithm, commit algorithm, transfer algorithm, generate trapdoor algorithm, and get message algorithm.

The experiments were conducted using a data range of 1,000 to 10,000 entries. In Figure 2, Figure 3 and Figure 6, the algorithms of the two schemes are almost identical in the setup, commit, and get message phases, resulting in similar time consumption. In Figure 4 and Figure 5, our scheme initially consumes more time due to the added cryptographic accumulator verification step. However, this slight increase in time consumption is acceptable. As the data volume increases, the time consumption of both schemes converges because cryptographic accumulator verification is more efficient than the traditional verification method used in the original scheme. The slight increase in time consumption is an acceptable trade-off for the added functionality.

Authorized licensed use limited to: University of Greenwich. Downloaded on March 12,2025 at 20:42:48 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3550198

VI. Conclusion

To address the challenge of ensuring timely updates of information from untrustworthy servers and guaranteeing that data recipients retrieve the most up-to-date information from the database, we propose a verifiable oblivious keyword search with authorization scheme. This scheme employs a cryptographic accumulator to aggregate all messages, revealing the ciphertext accumulator value while generating a witness value for authorized messages. These mechanisms allow data recipients to verify the freshness of the data before retrieval. Furthermore, the data sender can dynamically update cloud-stored data while efficiently adjusting the cryptographic accumulator and its verification values. This ensures that the cloud server updates the data promptly and that recipients consistently access the latest information. Under a rigorous security model, we provide a formal security proof for the proposed protocol. Experimental evaluations further demonstrate that the scheme achieves high efficiency and meets the practical application requirements of the Internet of Things (IoT).

Conflicts of Interest

The authors declare no conficts of interest.

VII. References Section

References

[1] Y.-W. Ti, C.-F. Wu, C.-M. Yu, and S.-Y. Kuo, "Benchmarking Dynamic Searchable Symmetric Encryption Scheme for Cloud-Internet of Things Applications," *IEEE Access*, vol. 8, pp. 1715–1732, 2020.

[2] Y. Su, X. Zhang, J. Qin, and J. Ma, "Efficient and Flexible Multiauthority Attribute-Based Authentication for IoT Devices," *IEEE Internet of Things Journal*, vol. 10, no. 15, pp. 13 945–13 958, Aug. 2023.

[3] J. Shi, Y. Yu, Q. Yu, H. Li, and L. Wang, "Toward Data Security in 6G Networks: A Public-Key Searchable Encryption Approach," *IEEE Network*, vol. 36, no. 4, pp. 166–173, Jul. 2022.

[4] H. Liu, Y. Ming, C. Wang, Y. Zhao, S. Zhang, and R. Lu, "Blockchain-assisted verifiable certificate-based searchable encryption against untrusted cloud server for Industrial Internet of Things," *Future Generation Computer Systems*, vol. 153, pp. 97–112, Apr. 2024.

[5] W. Deng, J. Li, H. Yan, A. S. Voundi Koe, T. huang, J. Wang, and C. Peng, "Self-sovereign identity management in ciphertext policy attribute based encryption for IoT protocols," *Journal of Information Security and Applications*, vol. 86, p. 103885, Nov. 2024.

[6] M. Ali, M.-R. Sadeghi, X. Liu, Y. Miao, and A. V. Vasilakos, "Verifiable online/offline multi-keyword search for cloud-assisted Industrial Internet of Things," *Journal of Information Security and Applications*, vol. 65, p. 103101, Mar. 2022.

[7] J. Yao and L. Xu, "Online/Offline Attribute-Based Boolean Keyword Search For Internet Of Things," *The Computer Journal*, vol. 66, no. 12, pp. 2948–2960, Dec. 2023.

[8] Y. Zhou, J. Nan, and L. Wang, "Fine-Grained Attribute-Based Multi-keyword Search for Shared Multiowner in Internet of Things," *Security and Communication Networks*, vol. 2021, pp. 1–14, May 2021.

[9] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *Journal of Network and Computer Applications*, vol. 126, pp. 45–58, Jan. 2019.

[10] W. Shen, J. Yu, M. Yang, and J. Hu, "Efficient Identity-Based Data Integrity Auditing With Key-Exposure Resistance for Cloud Storage," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 6, pp. 4593–4606, Nov. 2023.

[11] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceeding 2000 IEEE symposium on security and privacy. S&P 2000*. IEEE, 2000, pp. 44–55.

[12] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23*. Springer, 2004, pp. 506–522.

[13] 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," in *Workshop on secure data management*. Springer, 2006, pp. 75–83.

[14] W. Ogata and K. Kurosawa, "Oblivious keyword search," *Journal of complexity*, vol. 20, no. 2-3, pp. 356–371, 2004.

[15] P. Jiang, X. Wang, J. Lai, F. Guo, and R. Chen, "Oblivious keyword search with authorization," in *Provable Security: 10th International Conference, ProvSec 2016, Nanjing, China, November 10-11, 2016, Proceedings 10*. Springer, 2016, pp. 173–190.

[16] J. Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," pp. 1249–1259, 2008.

[17] M. J. Freedman, Y. Ishai, B. Pinkas, and O. Reingold, "Keyword search and oblivious pseudorandom functions," in *Theory of Cryptography: Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005. Proceedings 2*. Springer, 2005, pp. 303–324.

[18] H. S. Rhee, J. W. Byun, D. H. Lee, and J. Lim, "Oblivious conjunctive keyword search," pp. 318–327, 2005.

[19] Z. Huafei and B. Feng, "Oblivious keyword search protocols in the public database model," pp. 1336–1341, 2007.

[20] J. Camenisch, M. Kohlweiss, A. Rial, and C. Sheedy, "Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data," in *Public Key Cryptography–PKC 2009: 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18-20, 2009. Proceedings 12*. Springer, 2009, pp. 196–214.

[21] J. Benaloh and M. De Mare, "One-way accumulators : A decentralized alternative to digital signature," pp. 274–285, 1993.

[22] J. Camenisch and A. Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials," in *Advances in CryptologyCRYPTO 2002: 22nd Annual International Cryptology Conference Santa Barbara, California, USA, August 18–22, 2002 Proceedings 22*. Springer, 2002, pp. 61–76.

[23] N. Barić and B. Pfitzmann, "Collision-free accumulators and fail-stop signature schemes without trees," pp. 480–494, 1997.

[24] L. Nguyen, "Accumulators from bilinear pairings and applications to id-based ring signatures and group membership revocation," *Topics in Cryptology-CT-RSA*, pp. 275–292, 2005.

[25] I. Damgård and N. Triandopoulos, "Supporting non-membership proofs with bilinear-map accumulators," *Cryptology ePrint Archive*, 2008.

[26] M. H. Au, P. P. Tsang, W. Susilo, and Y. Mu, "Dynamic universal accumulators for ddh groups and their application to attribute-based anonymous credential systems," pp. 295–308, 2009.

[27] J. Camenisch, M. Kohlweiss, and C. Soriente, "An accumulator based on bilinear maps and efficient revocation for anonymous credentials," pp. 481–500, 2009.

[28] P. Camacho, A. Hevia, M. Kiwi, and R. Opazo, "Strong accumulators from collision-resistant hashing," pp. 471–486, 2008.

[29] O. Goldreich and Y. Oren, "Definitions and properties of zero-knowledge proof systems," *Journal of Cryptology*, vol. 7, no. 1, pp. 1–32, 1994.

[30] D. Pennino, M. Pizzonia, and F. Griscioli, "Pipeline-integrity: Scaling the use of authenticated data structures up to the cloud," *Future Generation Computer Systems*, vol. 100, pp. 618–647, 2019.

[31] C. Martel, G. Nuckolls, P. Devanbu, M. Gertz, A. Kwong, and S. G. Stubblebine, "A general model for authenticated data structures," *Algorithmica*, vol. 39, pp. 21–41, 2004.

[32] J. Xu, L. Wei, W. Wu, A. Wang, Y. Zhang, and F. Zhou, "Privacy-preserving data integrity verification by using lightweight streaming authenticated data structures for healthcare cyber-physical system," *Future Generation Computer Systems*, vol. 108, pp. 1287–1296, 2020.

[33] D. Boneh, B. Bünz, and B. Fisch, "Batching techniques for accumulators with applications to iops and stateless blockchains," pp. 561–586, 2019.

[34] Y. Miao, Q. Tong, R. H. Deng, K.-K. R. Choo, X. Liu, and H. Li, "Verifiable searchable encryption framework against insider keyword-guessing attack in cloud storage," *IEEE Transactions on Cloud Computing*, vol. 10, no. 2, pp. 835–848, 2020.

[35] Y. Liang, Y. Li, Q. Cao, and F. Ren, "Vpams: Verifiable and practical attribute-based multi-keyword search over encrypted cloud data," *Journal of Systems Architecture*, vol. 108, p. 101741, 2020.

[36] W. Yang and Y. Zhu, "A verifiable semantic searching scheme by optimal matching over encrypted data in public cloud," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 100–115, 2020.

[37] Q. Tong, Y. Miao, X. Liu, K.-K. R. Choo, R. H. Deng, and H. Li, "Vpsl: Verifiable privacy-preserving data search for cloud-assisted internet of things," *IEEE Transactions on Cloud Computing*, vol. 10, no. 4, pp. 2964–2976, 2020.

Authorized licensed use limited to: University of Greenwich. Downloaded on March 12,2025 at 20:42:48 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3550198

[38] D. Sharma and D. Jinwala, "Simple index based symmetric searchable encryption with result verifiability," *Frontiers of Computer Science*, vol. 15, no. 2, p. 152805, 2021.

[39] K. Kurosawa and Y. Ohtaki, "How to update documents verifiably in searchable symmetric encryption," in *Cryptology and Network Security: 12th International Conference, CANS 2013, Paraty, Brazil, November 20-22. 2013. Proceedings 12*. Springer, 2013, pp. 309–328.

[40] Y. Yang, Y. Hu, X. Dong, J. Shen, Z. Cao, G. Yang, and R. H. Deng, "Opense: Efficient verifiable searchable encryption with access and search pattern hidden for cloud-iot," *IEEE Internet of Things Journal*, vol. 11, pp. 13 793–13 809, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:266061665

[41] N. H. Sultan, S. K. Kermanshahi, H. Y. Tran, S. Lai, V. Varadharajan, S. Nepal, and X. Yi, "A multi-client searchable encryption scheme for iot environment," *ArXiv*, vol. abs/2305.09221, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258715322

[42] Y. Yang, G. Zhang, S. Li, and Z. Liu, "Offline/online attribute-based searchable encryption scheme from ideal lattices for iot," *Frontiers Comput. Sci.*, vol. 18, p. 183817, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:267236297

[43] H. S. Rhee, W. Susilo, and H.-J. Kim, "Secure searchable public key encryption scheme against keyword guessing attacks," *IEICE Electronics Express*, vol. 6, no. 5, pp. 237–243, 2009.

[44] Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup, "Anonymous identification in ad hoc groups," pp. 609–626, 2004.

[45] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23*. Springer, 2004, pp. 506–522.

[46] W. Ogata and K. Kurosawa, "Oblivious keyword search," *Journal of complexity*, vol. 20, no. 2-3, pp. 356–371, 2004.

[47] F. Guo, Y. Mu, W. Susilo, and V. Varadharajan, "Membership encryption and its applications," in *Australasian Conference on Information Security and Privacy*. Springer, 2013, pp. 219–234.

[48] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2005, pp. 440–456.

[49] S. Mitsunari, R. Sakai, and M. Kasahara, "A new traitor tracing," *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 85, no. 2, pp. 481–484, 2002.

[50] D. Boneh and X. Boyen, "Short signatures without random oracles," in *International conference on the theory and applications of cryptographic techniques*. Springer, 2004, pp. 56–73.

[51] P. Jiang, X. Wang, J. Lai, F. Guo, and R. Chen, "Oblivious keyword search with authorization," in *Provable Security: 10th International Conference, ProvSec 2016, Nanjing, China, November 10-11, 2016, Proceedings 10*. Springer, 2016, pp. 173–190.

VIII. Biography Section

Zhongkai Wei received the BS degree from the School of Physics at Shandong University, Jinan University, P.R.China, in 2018. He is currently working toward the PhD degree with the School of Mathematics, Shandong University. His research interests include cloud computing, applied cryptography and Internet of Things.

Bo Zhao received the MS degree from the School of Mathematics, Shandong University, P.R.China, in 2022. From 2022 to 2024, he worked for Huakong Qingjiao Technology Co., Ltd., BeiJing, China. He is currently working toward the PhD degree in the School of Mathematics, Shandong University. His research interests include searchable encryption and secure multi-party computation.

Haining Yang received his Ph.D. degree in School of Mathematics, Shandong University, China, in 2021. He was a visiting Ph.D. student in School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore. He worked as a post-doctoral in State Key Laboratory of Cryptology, Beijing, China, from August, 2021 to August, 2023. He is working in School of Mathematics, Shandong University from September, 2023. His research interest mainly includes public-key cryptography and cloud security.

Jing Qin is a professor with the School of Mathematics, Shandong University P. R. China. Her research interests include information security, design and analysis of security about cryptologic protocols. She has coauthored two books and has published more than 30 professional research papers. She is a senior member of the Chinese Association for Cryptologic Research (CACR) as well as China Computer Federation (CCF).

Jixin Ma received the Ph.D. degree in computer sciences from the University of Greenwich, London, U.K., in 1994. He is currently a Reader of Computer Science with the School of Computing and Mathematical Sciences, University of Greenwich, and a Visiting Professor with Beijing Normal University, Beijing, China; Auhui University, Hefei, China; and Zhengzhou Light Industrial University, Zhengzhou, China. He has published more than 100 research articles in international journals and conferences. His main research areas include artificial intelligence and information systems, with special interests in temporal logic and information security.

Authorized licensed use limited to: University of Greenwich. Downloaded on March 12,2025 at 20:42:48 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.