IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 3, 2022, accepted February 1, 2022, date of publication February 14, 2022, date of current version March 2, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3151429

Trapdoor Privacy in Public Key Encryption With Keyword Search: A Review

KOON-MING CHAN[1], **SWEE-HUAY HENG**[1], **WEI-CHUEN YAU**[2], **(Member, IEEE),**
AND SHING-CHIANG TAN[1]

[1]Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, Melaka 75450, Malaysia
[2]School of Computing and Data Science, Xiamen University Malaysia, Sepang, Selangor Darul Ehsan 43900, Malaysia

Corresponding authors: Koon-Ming Chan (koonming1996@gmail.com), Swee-Huay Heng (shheng@mmu.edu.my), and Wei-Chuen Yau (wcyau@xmu.edu.my)

This work was supported by the Malaysia Government's Fundamental Research Grant Scheme under Grant FRGS/1/2018/ICT04/MMU/01/01. The work of Wei-Chuen Yau was supported by Xiamen University Malaysia Research Fund under Grant XMUMRF/2019-C4/IECE/0011.

ABSTRACT The public key encryption with keyword search (PEKS) scheme allows searches to be performed over ciphertext by a server in a public-key setting. The PEKS scheme suffers from a major drawback which is keyword guessing attack. A keyword guessing attack (KGA) allows the attacker to successfully guess the correct keyword encrypted in a searchable ciphertext and trapdoor. To overcome this vulnerability, security notions, such as keyword privacy and trapdoor privacy were introduced. Keyword privacy prevents any information leaked from the keyword itself, and similarly trapdoor privacy prevents any information leaked from the trapdoor side. A PEKS scheme that is secure against KGA should satisfy trapdoor privacy. In this paper, we compare various types of PEKS schemes in terms of their underlying computational hardness, system model, search function, security properties of keyword privacy and trapdoor privacy, and security against offline KGA and online KGA. From the comparison analysis, we highlight that trapdoor privacy and keyword privacy are essential for a PEKS scheme to be secure against KGA. Lastly, we draw some potential research directions.

INDEX TERMS PEKS, searchable encryption, trapdoor privacy, keyword privacy.

I. INTRODUCTION

With the increasing number of information technology devices on the internet, the amount of data that must be maintained has dramatically increased over the years. One of the options to resolve this issue is to use cloud storage technology by outsourcing a cloud server to store data and retrieve them from the cloud when needed. Storing the data in the plaintext format would put the confidentiality of the data owner at risk, but storing the data in the encrypted format would pose a significant problem when searching for data on the cloud.

To overcome this challenge, searchable encryption (SE) scheme was introduced by [1] to search over encrypted data. A searchable encryption scheme allows a server to search for the data in the encrypted form on behalf of a client without learning any information about the plaintext data and thus, with the smallest possible loss of data confidentiality [2]. Figure 1 shows the general structure of the searchable encryption scheme. It consists of three main entities, data owner, data user, and cloud server. **Data owner:** the one who encrypts the data and index before uploading to the cloud server. **Data user:** the one who generates the trapdoor to enable the server to search over the encrypted data. **Cloud server:** the server stores the encrypted data and helps to perform searching operations on the cloud using the trapdoor. There are two types of searchable encryption scheme.

1) Symmetric searchable encryption (SSE) In the SSE scheme, the data is encrypted with user's secret key before outsourcing. The first SSE scheme was proposed by Song *et al.* [1]. The advantage of this scheme is its efficiency because the SSE scheme is based on symmetric primitives; thus, it requires less computational overhead. The disadvantage of this scheme is that its functionality is usually applicable only to single user scenario.
2) Public Key Encryption with Keyword Search (PEKS) In the PEKS scheme, the data is encrypted with user's public key before outsourcing. The first PEKS scheme

The associate editor coordinating the review of this manuscript and approving it for publication was Zhipeng Cai.

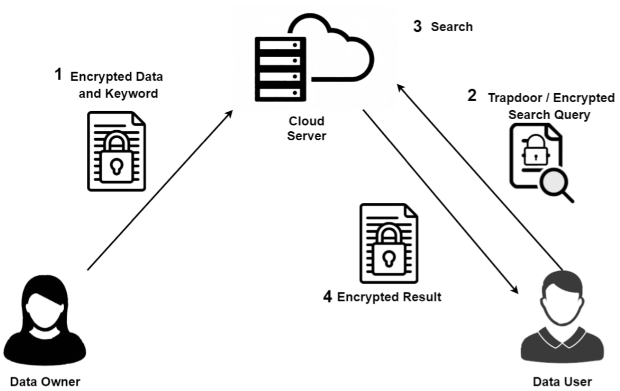
1 Encrypted Data and Keyword
Cloud Server
3 Search
2 Trapdoor / Encrypted Search Query
4 Encrypted Result
Data Owner
Data User

FIGURE 1. General structure of searchable encryption scheme.

was proposed by Boneh *et al.* [3]. The advantage of this scheme is its functionality, because it can be used in a multiuser setting. However, this scheme exhibits a low efficiency. According to Kamara and Lauter [4], most of the PEKS schemes require the evaluation of pairings on elliptic curves, which is relatively slow.

Current research works focus on improving the security and practicality of the PEKS scheme before deploying it in real-world applications. A keyword guessing attack (KGA) allows the attacker to successfully guess the correct keyword encrypted in a searchable ciphertext and trapdoor. To overcome this vulnerability, security notions such as keyword privacy and trapdoor privacy were introduced. Keyword privacy prevents any information leaked from the keyword itself, and similarly trapdoor privacy prevents any information leaked from the trapdoor side. Trapdoor privacy is an important property that needs to be satisfied by a PEKS scheme so that it is secure against keyword guessing attack.

In this paper, our survey is mainly focused on the trapdoor privacy of various types of PEKS schemes. First, we provide a summary on the development of PEKS schemes. We then introduce the property of trapdoor privacy followed by a comparison analysis on various PEKS schemes in terms of their underlying tools, computational hardness, system model, search function, security properties of keyword privacy and trapdoor privacy, and the security against offline KGA and online KGA. Subsequently, we outline some potential research directions for the PEKS scheme and conclude this review.

A. KEYWORD GUESSING ATTACK

A keyword guessing attack or KGA is the greatest vulnerability suffered by the PEKS scheme. This attack exploits the property of low entropy in the keyword space and allows the attacker to correctly guess the keyword encrypted in the given trapdoor. This attack can be categorised into two types, namely, offline keyword guessing attack and online keyword guessing attack.

Offline keyword guessing attack consists of two types of attackers, namely, outsider attacker and insider attacker. An outsider attacker is a malicious party that is not related to the service provider. They can eavesdrop on the public channel between the server and receiver to obtain a trapdoor transmitted over the public channel. An insider attacker usually refers to a malicious server that can obtain the trapdoor from any receiver. Both outsider and insider attackers can obtain the keyword ciphertext and the trapdoor, and the only difference between them is that the outsider attacker cannot perform the test algorithm (in the case of the dPEKS scheme), while the insider attacker can perform the test algorithm which makes it difficult to resist the insider attacker.

Online KGA only occurs for an outsider attacker. Instead of running the test algorithm, the attacker uploads the specially crafted ciphertext of the chosen keyword to the server and eavesdrops on the channel until the crafted ciphertext is queried by a receiver. Then the attacker will be able to guess the correct keyword for the corresponding trapdoor.

Byun *et al.* [5] first pointed out the vulnerability of offline KGA in the PEKS scheme and showed that previously proposed schemes [3], [6] are vulnerable to both insider and outsider offline KGA. Yau *et al.* [7] performed offline KGA on PEKS schemes and showed that Boneh *et al.*'s [3], Park *et al.*'s [6], and Baek *et al.*'s [8] schemes all are susceptible to offline KGA. Jeong *et al.* [9] showed that it is impossible to construct a secure and consistent PEKS scheme against KGA when the number of possible keywords is bounded by a polynomial. Yau *et al.* [10] presented an online KGA by an outsider attacker on previous dPEKS schemes. They demonstrated their proposed attack on Rhee *et al.*'s [11] scheme and claimed that their attack is generic which can be applied to all existing dPEKS schemes.

II. PUBLIC KEY ENCRYPTION WITH KEYWORD SEARCH

The first type of searchable encryption construction SSE is based on symmetric encryption, where only a secret key is involved in the encryption and decryption processes. Owing to the nature of symmetric encryption, it is not favourable for multiuser settings, and it has a secret key distribution issue. To resolve the problem of SSE, public key encryption with keyword search was subsequently introduced. The construction is based on asymmetric encryption, where a pair of public and private keys is involved in the encryption and decryption processes, which is suitable for multiuser settings. Figure 2 illustrates the structure of the PEKS scheme.

A PEKS scheme mainly consists of four polynomial time randomised algorithms [3]:

1) **KeyGen** (s)**:** this is a key generation algorithm that is run by a data receiver. This algorithm takes in a security parameter s, and outputs a public key A_{pub} and private key A_{priv}.
2) **PEKS** (A_{pub}, W)**:** this encryption algorithm that is run by a data sender. It takes in data receiver's public key A_{pub} and a keyword W and outputs a keyword ciphertext S of W.
3) **Trapdoor** (A_{priv}, W)**:** this is a keyword trapdoor generation algorithm that is run by the data receiver.

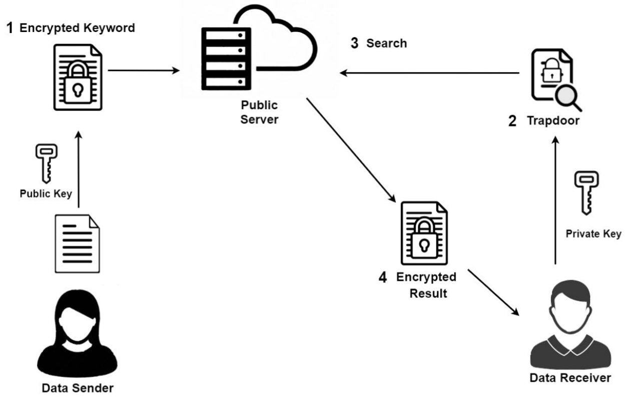

FIGURE 2. PEKS structure.

This algorithm takes in the data receiver's private key A_{priv} and a query keyword W and outputs the trapdoor T_W for the query keyword W.

4) **Test** (A_{pub}, S, T_W)**:** this test algorithm is run by the public server. It takes in the data receiver's public key A_{pub}, a keyword ciphertext $S = PEKS(A_{pub}, W')$ and a trapdoor $T_W = Trapdoor(A_{priv}, W)$. It outputs 'yes' if $W = W'$ else the output is 'no'.

A. DEVELOPMENT OF PEKS SCHEMES

Table 1 shows a list of abbreviations used for PEKS scheme. The first PEKS scheme introduced by Boneh *et al.* [3] was based on a public key cryptosystem using bilinear pairings. Their scheme was transformed from the identity-based encryption scheme proposed by Boneh and Franklin [12]. A generic PEKS scheme takes in a keyword and a public key to generate keyword ciphertext by running the PEKS algorithm. The keyword ciphertext is stored on a cloud server. The receiver generates a trapdoor by running the Trapdoor algorithm, using a private key and the desired keyword as the input. The trapdoor is sent to the cloud server to run the Test algorithm for searching. Abdalla *et al.* [13] improved the definition of the PEKS scheme and showed that Boneh *et al.*'s [3] scheme is computationally consistent. They also provided a transformation technique that can construct a secure PEKS scheme that guarantees consistency from an anonymous identity-based encryption scheme. Gu *et al.* [14] proposed a PEKS scheme that is more efficient than Boneh *et al.*'s scheme by removing the pairing operation in the encryption procedure. Sun *et al.* [15] improved Boneh *et al.*'s [3] PEKS scheme to be secure against insider keyword guessing attack by using the signcryption algorithm in the generation of searchable ciphertext. Zhang *et al.* [16] proposed a PEKS scheme that achieved trapdoor privacy in the random oracle model and logarithmic time pairing free searching over encrypted data.

Baek *et al.* [8] first noticed that the PEKS scheme proposed by Boneh *et al.* [3] required a secure channel for communication. They have also mentioned that building a secure channel is usually expensive, which may be unsuitable for some applications. In order to solve this problem, they introduced an improved PEKS scheme that eliminated the secure channel, which is called secure channel free public key encryption with keyword search (SCF-PEKS). Rhee *et al.* [11] noted that the security model of the scheme proposed by Baek *et al.* [8] limited the ability of an adversary to capture the attacks in a real-world environment. They improved the security model of Baek *et al.* [8] and proposed a new PEKS scheme called searchable public key encryption with designated tester (dPEKS). SCF-PEKS or dPEKS is a variant of the PEKS scheme that has the advantage of allowing only a designated server to run the Test algorithm, and the trapdoor can be transmitted over a public channel. The SCF-PEKS or dPEKS scheme requires an additional server public key to generate the keyword ciphertext and trapdoor. The server private key is also used as an input to run the Test algorithm. The disadvantage of this variant of the PEKS scheme is that it is vulnerable to offline KGA from insider attacker.

Fang *et al.* [17] noted that all previous SCF-PEKS schemes are proven secure in the random oracle model, which may lead to insecure scheme when the random oracles are implemented in real life. To resolve this issue, they have presented an efficient SCF-PEKS scheme that is proven secure in the standard model. Rhee *et al.* [18] first proposed the concept of trapdoor indistinguishability and showed that this property is sufficient to be against offline outsider keyword guessing attack. They have also proposed the first dPEKS scheme that is secure against offline keyword guessing attack and proved that their scheme satisfied ciphertext indistinguishability and trapdoor indistinguishability. Wang *et al.* [19] later noted that Rhee *et al.*'s [18] scheme was still vulnerable to offline keyword guessing attack in their test phase. Wang *et al.* [19] later improved Rhee *et al.*'s [18] scheme by adding a random parameter in the test phase to improve the scheme to be secure against offline keyword guessing attack from both outsider and insider attacker.

Rhee *et al.* [20] presented two generic transformations to construct a dPEKS scheme from an identity-based encryption scheme. They also claimed that the anonymity and confidentiality properties in an identity-based encryption scheme were sufficient to achieve consistency and confidentiality in a dPEKS scheme. Zhao *et al.* [21] proposed a new SCF-PEKS scheme that guaranteed trapdoor indistinguishability and performed better than the previous SCF-PEKS scheme. Yau *et al.* [22] proposed a new security models that captured keyword guessing attack in PEKS scheme and dPEKS scheme. They also claimed that their proposed security models achieved stronger keyword guessing notion as compared to Rhee *et al.*'s [18] security models. Fang *et al.* [23] introduced the notions of security against chosen keyword and chosen ciphertext attack (IND-SCF-CKCA) and keyword guessing attack (IND-KGA) for SCF-PEKS scheme. They later proposed a SCF-PEKS scheme in the standard model that is IND-SCF-SKCA and IND-KGA secure. Guo and Yau [24] proposed an efficient SCF-PEKS scheme that is

TABLE 1. Abbreviation for PEKS.

Name	Abbreviation
Public Key Encryption with Keyword Search	PEKS
Searchable Public Key Encryption with Designated Tester	dPEKS
Secure Channel Free Public Key Encryption with Keyword Search	SCF-PEKS
Secure Server-Designation Public Key Encryption with Keyword Search	SPEKS
Public Key Encryption with Conjunctive Field Keyword Search	PECKS
Public Key Encryption with Fuzzy Keyword Search	PEFKS
k-resilient Public Key Encryption with Keyword Search	KR-PEKS
Searchable Proxy Re-Encryption	Re-PEKS
Dual-Server Public Key Encryption with Keyword Search	DS-PEKS
Server-Aided Public Key Encryption with Keyword Search	SA-PEKS
Public Key Authenticated Encryption with Keyword Search	PAEKS

proven secure against chosen keyword, chosen ciphertext, and keyword guessing attack in the standard model. They also claimed that their scheme was more efficient that previously proposed SCF-PEKS schemes. The SPEKS scheme is similar to the usual dPEKS scheme but with additional encryption and decryption processes after the server performs the Test algorithm. In the SPEKS scheme, after the server identifies the matching keyword, it encrypts the keyword-matching data by using the receiver's public key. After the receiver receives the encrypted data, he/she runs the decryption algorithm to retrieve the plaintext data. The advantage of this scheme is that it is secure against online KGA because of the additional encryption process, which is also the disadvantage because it causes the scheme to be inefficient. Chen [25] proposed secure server-designation public key encryption with keyword search (SPEKS) to solve the problem faced by the dPEKS scheme due to an online keyword guessing attack. Emura *et al.* [26] presented two generic constructions of the adaptive SCF-PEKS scheme from an anonymous identity-based encryption scheme. They used a hybrid encryption technique called key encapsulation mechanism (KEM) and data encapsulation mechanism (DEM) framework for their generic construction.

Meanwhile, Park *et al.* [6] noticed that the PEKS scheme proposed by Boneh *et al.* [3] was limited by the number of keywords being searched in a single query. Thus, they introduced the notion of public key encryption with conjunctive field keyword search (PECKS).The difference between PECKS and PEKS is the number of keywords they can process. In PECKS keyword ciphertext generation, a group of keywords is used as the input, unlike PEKS uses only a single keyword. Similarly, in the trapdoor generation, a group of keywords is used as the input. In the Test algorithm, the PECKS tests a group of keywords together in a single query. Their scheme was further improved to multiuser setting by Hwang and Lee [27]. Xu *et al.* [28] proposed a public key encryption with fuzzy keyword search scheme (PEFKS) that was transformed from an anonymous identity-based encryption scheme. The PEFKS scheme allows to perform fuzzy search operations unlike normal PEKS scheme that only performs exact search operations. In the PEFKS scheme, the generated trapdoor consists of two parts, the exact test trapdoor and the fuzzy test trapdoor. The Test algorithm of the PEFKS scheme also consists of two parts, an exact test and a fuzzy test. The exact test uses the exact trapdoor to generate exact results, whereas the fuzzy test uses the fuzzy trapdoor to generate fuzzy results. Hwang *et al.* [29] proposed an efficient PEFKS scheme that is secure channel free and secure against offline keyword guessing attack in the standard model. Lu *et al.* [30] showed that Hwang *et al.*'s [29] scheme was vulnerable against keyword guessing attack.

The *k*-resilient PEKS scheme is the first proposed PEKS scheme without bilinear pairing, and its security has been proven in the standard model. The advantage of this scheme is that it is more efficient than the other pairing based PEKS schemes. Khader [31] first proposed a *k*-resilient PEKS scheme based on the *k*-resilient identity-based encryption (*k*-resilient IBE) proposed by Heng and Kurosawa [32] in the standard model. Yang *et al.* [33] claimed that Khader's [31] proposed scheme did not satisfy the required for the PEKS scheme functionality. They later improved Khader's [31] scheme that fulfilled computational consistency and improved efficiency. Tang [34] proposed an interactive PEKS scheme to address the trapdoor vulnerability in Boneh *et al.*'s [3] scheme. Their interactive PEKS scheme required both the sender and receiver to interactively

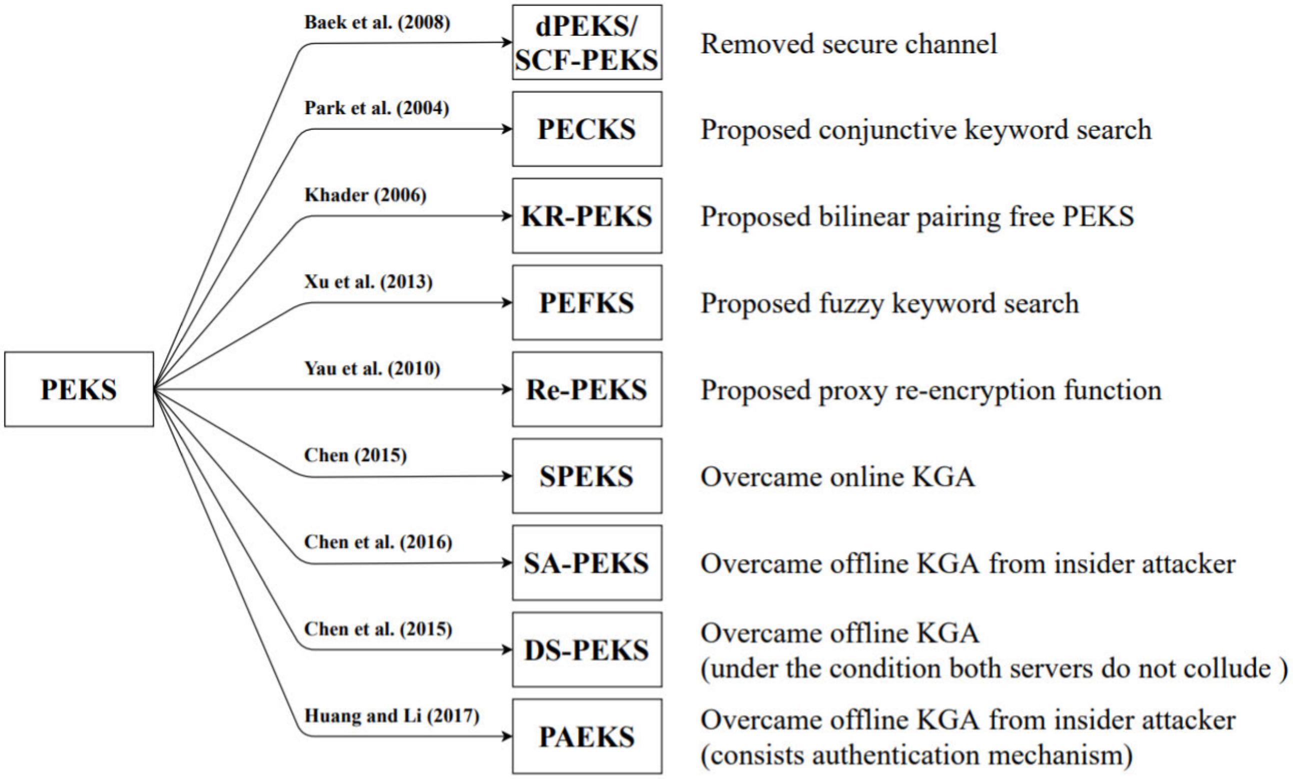
PEKS
Baek et al. (2008)
dPEKS/ SCF-PEKS
Removed secure channel
Park et al. (2004)
PECKS
Proposed conjunctive keyword search
Khader (2006)
KR-PEKS
Proposed bilinear pairing free PEKS
Xu et al. (2013)
PEFKS
Proposed fuzzy keyword search
Yau et al. (2010)
Re-PEKS
Proposed proxy re-encryption function
Chen (2015)
SPEKS
Overcame online KGA
Chen et al. (2016)
SA-PEKS
Overcame offline KGA from insider attacker
Chen et al. (2015)
DS-PEKS
Overcame offline KGA
(under the condition both servers do not collude)
Huang and Li (2017)
PAEKS
Overcame offline KGA from insider attacker
(consists authentication mechanism)

FIGURE 3. PEKS development.

generate a trapdoor. Shao *et al.* [35] proposed the concept of proxy re-encryption with keyword search (PRES) with keyword privacy secure in the random oracle model. Yau *et al.* [36] also have proposed a concept of searchable proxy re-encryption scheme (Re-PEKS) and proved their scheme secure in the random oracle model. Re-PEKS integrates a proxy re-encryption scheme with the PEKS scheme. It can translate a keyword ciphertext encrypted from a public key into a different public key without learning any information. The main difference between Yau *et al.*'s scheme and Shao *et al.*'s scheme lies in the structure of the scheme. Yau *et al.* [36] extended the original PEKS structure by adding a re-encryption key generation and keyword ciphertext algorithm. This means that the proposed scheme is more flexible in terms of the selection of different standard proxy re-encryption techniques to be used in the Re-PEKS scheme.

Chen *et al.* [37] proposed a new PEKS framework called dual-server public key encryption with keyword search (DS-PEKS) scheme. They proved that their scheme could withstand an offline keyword guessing attack if both servers were not colluded. In the DS-PEKS scheme, there are two servers running the Test algorithm. In keyword ciphertext and trapdoor generation, the public keys of both servers are required to execute the algorithms. The DS-PEKS Test algorithm is divided into FrontTest and BackTest. The FrontTest is first run by the front server to produce an internal testing state, which later serves as an input for the back server to run the BackTest to output the actual test result. The advantage of this scheme is that it is secure against offline KGA but relies on two servers which makes it inefficient. Chen *et al.* [38] also pointed out that the DS-PEKS scheme proposed by Chen *et al.* [37] suffers from inefficiency because the keyword search process is handled by two servers separately. They later proposed a new PEKS system named server-aided public key encryption with keyword search (SA-PEKS), which is more practical and secure against offline insider keyword guessing attack. In SA-PEKS, an additional server (keyword server) is responsible for preprocessing the keyword before it is encrypted into a keyword ciphertext or trapdoor. The sender and receiver are requires to run an interactive protocol with the keyword server to obtain the preprocess keyword. This provided an authentication mechanism. This allows DS-PEKS scheme to be secure against offline KGA form insider attacker by the disadvantage is that the scheme is inefficient because it required sender and receiver to interactively run a protocol to generate keyword ciphertext and trapdoor.

The PAEKS scheme offers authentication because it uses a sender key pair. In PAEKS, the sender's private key and receiver's public key are used to produce the keyword ciphertext. The sender's public key and receiver's private key are later used to generate the trapdoor. In the Test algorithm, both parties' public keys are required along with a trapdoor and a keyword ciphertext. In this setting, any third party generates a valid keyword ciphertext is impossible. Thus, the advantage of this scheme is that it can be secure against offline KGA from insider attacker. Huang and Li [39] proposed the notion of public key authenticated encryption with keyword search (PAEKS) to solve the problem of insider keyword guessing attack. Their proposed scheme requires a sender to authenticate the encrypted keyword upload to the cloud server. Qin *et al.* [40] revisited Huang and Li's [39] PAEKS scheme. They mentioned that the security model of PAEKS scheme proposed by Huang and Li [39] did not capture the outsider chosen multi-ciphertext attack. To solve this problem, they proposed a new security model that captured outsider chosen multi-ciphertext attack and insider keyword guessing attack. Miao *et al.* [41] proposed a verifiable PEKS scheme to address the issue of inaccurate search results from the cloud server. Their proposed scheme achieves trapdoor privacy and secures against insider keyword guessing attack. They also extended their work to multi-keyword search and record dynamic updates.

Figure 3 shows a summary of the development of PEKS schemes and the limitations they aim to overcome.

B. KEYWORD PRIVACY

Keyword privacy was first defined by Boneh *et al.* [3] where the adversary should not be able to distinguish between two ciphertexts of keywords W_0 and W_1, respectively, under the condition that no trapdoors are obtained for the respective keywords. Boneh *et al.* [3] defined a game between an attacker and a challenger to show that the PEKS scheme is indistinguishability against chosen keyword attack (IND-CKA).

A PEKS IND-CKA game is defined as follows:

1) The challenger first runs the KeyGen (s) algorithm to generate public keys A_{pub} and private key A_{priv}. Public key A_{pub} is given to the attacker.
2) The attacker can adaptively query for the trapdoor T_W for any keyword W of his/her choice from the challenger.
3) When the attacker is ready, he/she will send two words W_0 and W_1 that he/she wishes to be challenged to the challenger. The words chosen by the attacker should not be queried for trapdoor previously. The challenger chooses a random b and sends a ciphertext $C = PEKS(A_{pub}, W_b)$ to the attacker.
4) The attacker can continue to query for trapdoor T_W for any keyword W, except for the challenge keywords W_0 and W_1.
5) The attacker wins the game if he/she guessed the correct random b.

C. TRAPDOOR PRIVACY

Keyword privacy guarantees that no information about the keyword should be leaked from the searchable ciphertext of the PEKS scheme. This property was satisfied by almost all the PEKS schemes. However, Rhee *et al.* [18] found that the security of trapdoor is also significant to construct a PEKS scheme that is secure against keyword guessing attack. Trapdoor privacy ensures that no information about the keyword is leaked from the trapdoor, and Rhee *et al.* [18] proposed the notion of trapdoor indistinguishability to capture this issue. The notion of trapdoor indistinguishability should not allow an outsider attacker to distinguish between the trapdoor of two challenge keywords of its choice, under the situation that it is allowed to obtain trapdoors for any non-challenge keywords.

Nishioka [42] also proposed a security notion to address trapdoor privacy, which they called perfect keyword privacy (PKP) and search pattern privacy (SPP). This notion was later fomalised by Arriaga *et al.* [43] and is called weak key unlinkability. They also showed that weak key unlinkability failed to hide the search patterns when more than two trapdoors were queried. They later proposed a stronger notion called strong key unlinkability to overcome this deficiency. Their strong key unlinkability notion allows adversary to query multiple trapdoors and protect the search pattern at the same time.

With a keyword guessing attack as the main challenge for the PEKS scheme, the security of the trapdoor also needs to be considered. In a PEKS scheme, the searchable keyword and trapdoor are transmitted over the network which makes them vulnerable to points of attack. Thus, keyword privacy alone is insufficient for constructing a secure PEKS scheme because it only protects the privacy from the sender side. For a PEKS scheme to be secure against an offline keyword guessing attack from an outsider attacker, the minimum requirement is to satisfy keyword privacy and trapdoor privacy which protects the privacy from both the sender and the receiver side. In the literature, a number of studies have proposed a PEKS scheme with trapdoor privacy and security against offline keyword guessing attack, but some of them suffer from inefficiency; that is, using the computationally expensive bilinear pairing operation, only allows single keyword search functionality and higher communication cost.

1) THE RHEE *et al.* [18] SCHEME

Rhee *et al.* [18] proposed the security notion of trapdoor indistinguishability, which was limited to the dPEKS scheme, and they only captured the trapdoor security from an outsider attacker. Their proposed security notion guaranteed that the outsider attacker should not be able to differentiate between the trapdoors of two challenge keywords of its choice, under the condition that the outsider attacker is allowed to query trapdoors for non-challenge keywords.

Rhee *et al.* [18] have modelled the trapdoor indistinguishability game between a challenger and an attacker as follows and Figure 4 is a visual representation of the game:

Setup: In this phase, the public parameters and the private and public keys for the server and receiver are generated.

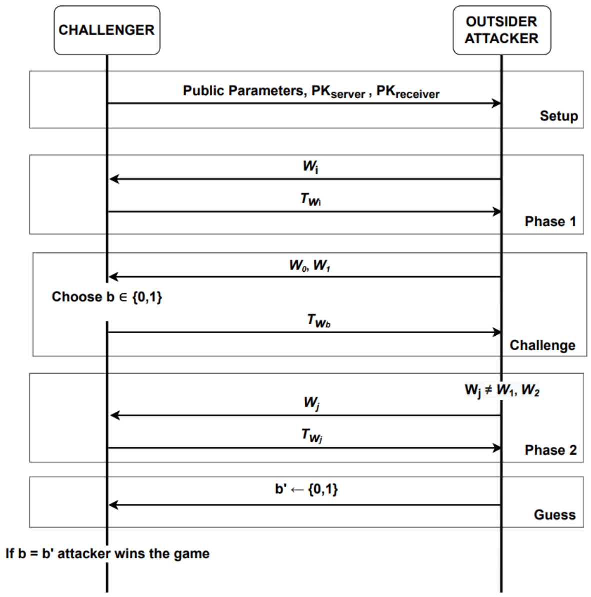

FIGURE 4. Trapdoor indistinguishability by [18].

Only the public key of the server and receiver is provided to the outsider attacker.

Phase 1 (Trapdoor Queries): In this phase, the outsider attacker is allowed to query the trapdoor of any keyword of its choice.

Challenge: In this phase, the outsider attacker selects two keywords to be challenged. The selected keywords should not be queried in the previous phase. Challenged keywords were provided to the challenger. The challenger computes the trapdoor with a random bit and returns it to the attacker.

Phase 2 (Trapdoor Queries): In this phase, the outsider attacker can continue to query for trapdoor as long as the keyword is not the challenge keyword.

Guess: This is the final phase of the game, in which the outsider attacker needs to guess the random bit chosen by the challenger. The outsider attacker wins the game if and only if the random bit is correctly guessed.

2) THE NISHIOKA [42] SCHEME

Nishioka [42] presented the security notion of perfect keyword privacy (PKP) that ensures not only the privacy of the keyword but also the trapdoor. The security notion of perfect keyword privacy guarantees that there is no efficient way to guess the keyword from the given trapdoor and searchable ciphertext. Nishioka [42] also proposed search pattern privacy (SSP) as an additional security notion for PKP because of the inability of PKP to capture search pattern privacy. The trapdoors are generated in a deterministic manner; therefore, it is easy for the adversary to guess the corresponding keyword from two trapdoors generated from the same private key.

The game for PKP is modelled as follows and Figure 5 is a visual representation of the game:

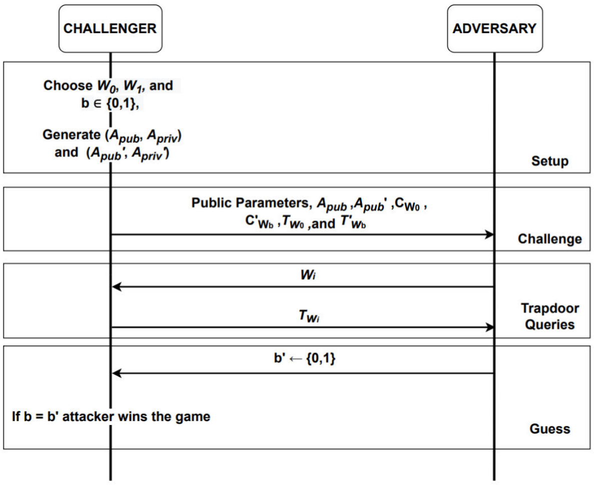
CHALLENGER
ADVERSARY
Choose W_0, W_1, and b ∈ {0,1},
Generate (A_{pub}, A_{priv}) and (A_{pub}', A_{priv}')
Setup
Public Parameters, A_{pub} ,A_{pub}' ,C_{W_0} , C'_{W_b} ,T_{W_0} ,and T'_{W_b}
Challenge
W_i
T_{W_i}
Trapdoor Queries
b' ← {0,1}
If b = b' attacker wins the game
Guess

FIGURE 5. PKP by [42].

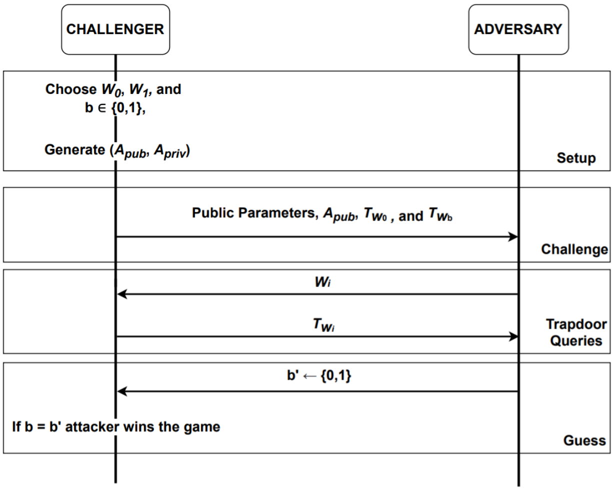
CHALLENGER
ADVERSARY
Choose W_0, W_1, and b ∈ {0,1},
Generate (A_{pub}, A_{priv})
Setup
Public Parameters, A_{pub}, T_{W_0} , and T_{W_b}
Challenge
W_i
T_{W_i}
Trapdoor Queries
b' ← {0,1}
If b = b' attacker wins the game
Guess

FIGURE 6. SPP by [42].

Setup: In this phase, two keywords (W_0 and W_1) are chosen from the keyword space. A challenge bit ($b = 0$ or 1) is selected, and the key generation algorithm generates two sets of public key and private keys ((A_{pub}, A_{priv}), (A'_{pub}, A'_{priv})).

Challenge: In this phase, the trapdoor and searchable ciphertext of first keyword (W_0) are generated using the first pair of public and private key (A_{pub}, A_{priv}). Generate the trapdoor and the searchable ciphertext of keyword chosen randomly based on the challenge bit (W_b) using the second pair of the public and private key (A'_{pub}, A'_{priv}). Both generated trapdoors and searchable ciphertexts were given to the adversary along with both public keys.

Trapdoor Queries: In this phase, the adversary can continue to query for trapdoor.

Guess: The adversary must guess the chosen random bit. If it correctly guesses the challenge bit, it wins the game.

The game for SSP is modelled as follows and Figure 6 is a visual representation of the game:

Setup: In this phase, two keywords (W_0 and W_1) were chosen from the keyword space. A challenge bit ($b = 0$ or 1) is selected and using the key generation algorithm to generate a set of public key and private key (A_{pub}, A_{priv}).

Challenge: In this phase, two trapdoors are generated, one using the first keyword (W_0) and the other one using the randomly chosen keyword based on the challenge bit (W_b). Both trapdoors are given to the adversary.

Trapdoor Queries: In this phase, the adversary can continue to query for trapdoor.

Guess: The adversary must guess the challenge bit.If it correctly guesses the challenge bit, it wins the game.

3) THE ARRIAGA *et al.* [43] SCHEME

Arriaga *et al.* [43] noted that the security notion of SSP proposed by Nishioka [42] could not be reflected in real-world scenarios because it limits the adversary to query only two trapdoors instead of multiple trapdoors. They first formulated the SSP notion to weak key unlinkability and then further enhanced it to strong key unlikability, where the adversary can query multiple trapdoors. Their proposed security notions were used for an identity-based encryption scheme (IBE), but after applying black-box transformation [13], the PEKS scheme will be achieved with a stronger guarantee of trapdoor privacy.

The weak key unlinkability for the IBE scheme is modelled as follows and Figure 7 is a visual representation of the game:

Setup: In this phase, the public parameters and master key are generated. A challenge bit is selected ($b = 0$ or 1). Two identities (id_0, id_1) were selected from the identity space.

Challenge: In this phase, two partial private keys are generated, first partial private key is generated with first identity (id_0), and the second partial private key is generated based on the challenge bit (id_b). Both the partial private keys are given to the adversary.

Queries: In this phase, the adversary can continue to query for partial private key.

Guess: The adversary must guess the challenge bit. The adversary wins the game if it correctly guesses the challenge bit.

The strong key unlinkability for the IBE scheme is modelled as follows and Figure 8 is a visual representation of the game:

Setup: In this phase, the public parameters and master key are generated. A challenge bit was selected ($b = 0$ or 1). Two empty lists are generated, one for storing identity and the other for storing the partial private key. Two lists ($list_0$ and $list_1$) were generated with size L identities.

Challenge: In this phase, the challenger randomly chooses a list based on the challenge bit ($list_b$) to generate a list of partial private keys ($list_{sk}$). The generated list of partial private keys is given to the adversary.

Queries: In this phase, the adversary can continue to query for partial private key.

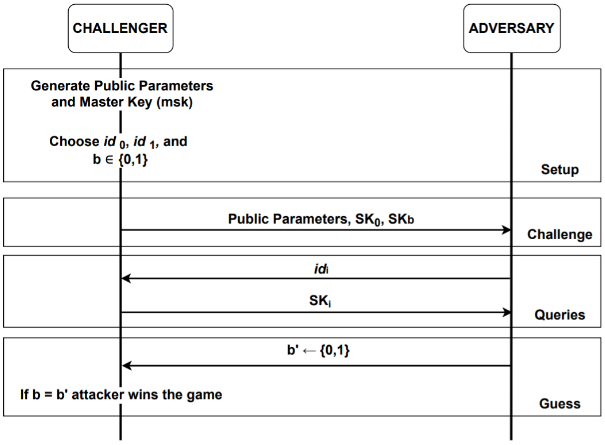
CHALLENGER
ADVERSARY
Generate Public Parameters and Master Key (msk)
Choose id 0, id 1, and b ∈ {0,1}
Setup
Public Parameters, SK0, SKb
Challenge
idi
SKi
Queries
b' ← {0,1}
If b = b' attacker wins the game
Guess

FIGURE 7. Weak key unlinkability by [43].

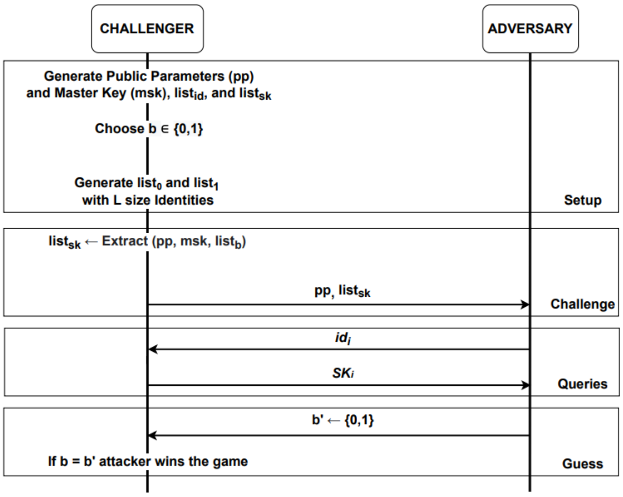
CHALLENGER
ADVERSARY
Generate Public Parameters (pp) and Master Key (msk), listid, and listsk
Choose b ∈ {0,1}
Generate list0 and list1 with L size Identities
Setup
listsk ← Extract (pp, msk, listb)
pp, listsk
Challenge
idi
SKi
Queries
b' ← {0,1}
If b = b' attacker wins the game
Guess

FIGURE 8. Strong key unlinkability by [43].

Guess: The adversary must guess the challenge bit. The adversary wins the game if it correctly guesses the challenge bit.

4) THE LU AND LI [44] SCHEME

Lu and Li [44] proposed a new trapdoor privacy security notion applicable to the PAEKS scheme. Their proposed security notion is called the search trapdoor indistinguishability against KGA (ST-IND-KGA). In their proposed security notion, an adversary can be assumed to be either a malicious insider attacker or an outsider attacker. Lu and Li [44] mentioned that the previously proposed security notions for PAEKS are vulnerable to adaptive chosen attack, which means that the adversary is allowed to choose their challenge target adaptively. To overcome this vulnerability, an adversary in the security notion proposed by Lu and Li [44] can adaptively choose their challenge target.

Lu and Li modelled the ST-IND-KGA game between a challenger and an attacker as follows, and 9 is a visual representation of the game:

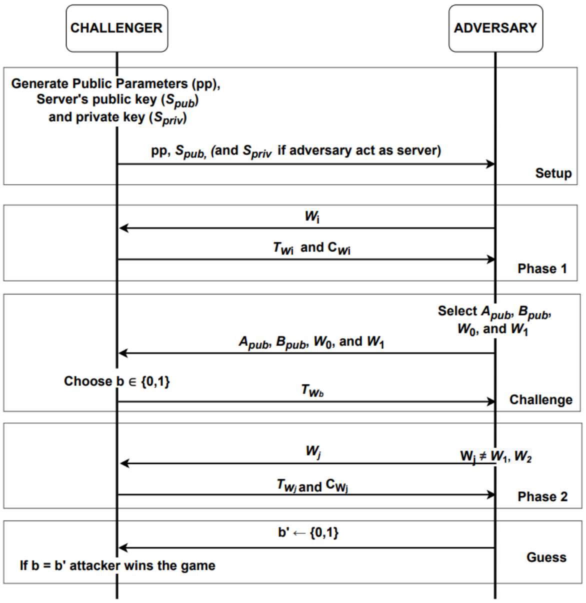
CHALLENGER
ADVERSARY
Generate Public Parameters (pp), Server's public key (S_{pub}) and private key (S_{priv})
pp, S_{pub}, (and S_{priv} if adversary act as server)
Setup
W_i
T_{Wi} and C_{Wi}
Phase 1
Select A_{pub}, B_{pub}, W_0, and W_1
A_{pub}, B_{pub}, W_0, and W_1
Choose b ∈ {0,1}
T_{W_b}
Challenge
W_j
$W_j \neq W_1, W_2$
T_{W_j} and C_{W_j}
Phase 2
b' ← {0,1}
If b = b' attacker wins the game
Guess

FIGURE 9. ST-IND-KGA by [44].

Setup: In this phase, the public parameters and the server's public and private keys (S_{pub}, S_{priv}) are generated. The public parameters and the server's public key (S_{pub}) are given to the adversary. If the adversary acts as the server, the server's private key (S_{priv}) is also given.

Phase 1 (Trapdoor Queries): In this phase, the adversary is allowed to adaptively query for keyword ciphertext and trapdoor of its choice.

Challenge: In this phase, the adversary selects two public keys (A_{pub} and B_{pub}) and two keywords (W_0 and W_1) to be challenged. The selected keywords should not be queried by the adversary in the previous phase. The challenger randomly selects a challenge bit ($b = 0$ or 1) and returns the adversary with the trapdoor of a randomly selected keyword (W_b) encrypted with the server's public key (S_{pub}), the selected public key (A_{pub}), and the selected private key (B_{priv}).

Phase 2 (Trapdoor Queries): In this phase, the adversary is allowed to adaptively query for keyword ciphertext and trapdoor of its choice except for the challenge keywords (W_0 and W_1).

Guess: In this phase, the adversary must guess the challenge bit. The adversary wins the game if it correctly guesses the challenge bit.

III. COMPARISON ANALYSIS FOR DIFFERENT PEKS VARIANTS

Table 2 shows a list of computational hardness abbreviations used for following PEKS variants comparison. In this section, we compare various types of PEKS schemes in terms of the underlying, computational hardness, system model, search function, security properties of keyword privacy and trapdoor privacy, and the security against offline KGA and online KGA.

A. PEKS

Table 3 shows a comparison of PEKS schemes. Boneh *et al.* [3] proposed the first PEKS scheme based on bilinear pairing, but their scheme only guaranteed keyword privacy. Park *et al.* [6] proposed the first PECK scheme based on bilinear pairing in the random oracle model that allowed multiple keywords in a single search query. Their proposed scheme was time efficient because it only used one pairing operation in the Test algorithm. *k*-resilient public key encryption with keyword search (KR-PEKS) was first proposed by Khader [31]. Her proposed scheme was transformed from *k*-resilient IBE without a pairing operation. Tang and Chen [45] proposed the first PERKS scheme that achieved keyword privacy and secure against offline KGA from both attackers. The pre-registration of the keyword in their proposed scheme is a crucial technique that protects against offline KGA, but it is also the main drawback of the PERKS scheme because it requires an interaction between the sender and receiver. Yau *et al.* [36] first proposed the RE-PEKS scheme based on bilinear map in the random oracle model scheme that uses a proxy server to translate a keyword encrypted under a public key into the same keyword encrypted under a different public key. Their proposed scheme satisfied keyword privacy.

Yang *et al.* [33] noted that Khader's scheme does not satisfy consistency, which is necessary for the PEKS scheme. They improved Khader's scheme to achieve computational consistency and greatly improved the efficiency. Yau *et al.* [46] pointed out that in the Khader's [31] scheme has some unnecessary steps, and some can be simplified to fewer steps. Yang *et al.*'s [33] scheme also suffers from these issues. Yau *et al.* [46] later improved Khader's [31] scheme to achieve better efficiency. Yau *et al.* [46] also noticed that Khader's scheme strongly relied on the security of the underlying building block, that is, IND-CCA *k*-resilient IBE in order to achieve the security of keyword privacy for the proposed scheme. According to Yau *et al.* [46], it is unnecessary to include this requirement. They proposed a more relaxed requirement that only requires the *k*-resilient IBE scheme to be IND-CPA, which is easier to achieve than IND-CCA, to achieve the same security as Khader's [31] scheme after transformation.

Nishioka [42] introduced the notion of search pattern privacy that guaranteed trapdoor privacy, which was later improved by Arriaga *et al.* [43] because the privacy of the trapdoor would be compromised if more than two trapdoors were queried. The security notion is called strong key unlinkability. Hwang *et al.* [29] proposed a PECK scheme based on bilinear pairing in the standard model. Xu *et al.* [28] proposed the first PEFKS scheme that satisfied keyword privacy and secure against offline KGA from outsider attacker. They also proposed a universal transformation from anonymous identity-based encryption to a secure PEFKS scheme. Sun *et al.* [15] proposed a hybrid framework of PEKS and SSE that requires the sender to send the trapdoor generation key to the receiver for trapdoor generation. Their proposed

TABLE 2. Abbreviation for computational hardness.

Name	Abbreviation
Augmented Bilinear Diffie-Hellman Exponent	ABDHE
Bilinear Decisional Diffie-Hellman	BDDH
Bilinear Diffie-Hellman	BDH
Bilinear Diffie-Hellman Inversion	BDHI
Computational Bilinear Diffie-Hellman	CBDH
Computational Decisional Diffie-Hellman	CDDH
Computational Diffie-Hellman	CDH
Decisional Bilinear Diffie-Hellman	DBDH
Decisional Diffie-Hellman	DDH
Decisional Linear Diffie-Hellman	DLDH
Decisional Linear	DLIN
Discrete Logarithm Problem	DLP
Hash Diffie-Hellman	HDH
Modified Decisional Linear	mDLIN
One-more Discrete Logarithm	OMDL
Truncated Decisional Augmented Bilinear Diffie-Hellman Exponent	q-ABDHE
Quotient Decisional Bilinear Diffie-Hellman	QDBDH
Symmetric External Diffie-Hellman	SXDH

scheme is secures against the offline KGA attack from insider attacker but suffers from key distribution problem because the trapdoor generation key needs to be sent to the receiver secretly. Wu *et al.* [47] proposed a new PEKS scheme based on bilinear pairing with Diffie-Hellman shared secret key protocol to achieve keyword privacy, trapdoor privacy, and secure against offline KGA from outsider and insider attacker. Lu *et al.* [48] proposed a new PEKS scheme without bilinear pairing. Their proposed scheme was based on a prime order elliptic curve group, and it satisfied keyword privacy, trapdoor privacy, and secure against offline KGA from outsider attacker. Xu *et al.* [49] proposed a new PECK scheme based on bilinear pairing that satisfied keyword privacy, trapdoor privacy, and secure against offline KGA from insider attacker. Liu *et al.* [50] proposed a new PEKS scheme based on a distributed two-trapdoor public key cryptosystem (DT-PKC) and proven their scheme achieved keyword privacy, trapdoor privacy, and secured against offline KGA from insider attacker.

B. dPEKS/SCF-PEKS

Designated public key encryption with keyword search (dPEKS) and secure channel free public key encryption with keyword search (SCF-PEKS) are variants of the PEKS scheme that allow only the designated server to perform the search operation and allow the transmission of trapdoor via a public channel. Table 4 shows a comparison of various dPEKS/SCF-PEKS schemes. Rhee *et al.* [18] first introduced the security notion of trapdoor indistinguishability to achieve trapdoor privacy against offline KGA from outsider attacker. Zhao *et al.* [21] proposed a new efficient SCF-PEKS scheme that achieved trapdoor privacy. Fang *et al.* [23] proposed a new SCF-PEKS scheme based on bilinear pairing in the standard model that achieved keyword privacy, trapdoor privacy, and secure against offline KGA from outsider attacker. Shao and Yang [52] proposed a dPEKS scheme based on Fang *et al.*'s scheme [23], which achieves security against offline KGA from insider attacker. They used a digital signature scheme to generate searchable ciphertext and trapdoor to prevent the server from executing the Test algorithm using searchable ciphertext generated by the server itself. However, their scheme was later shown by Lu *et al.* [54] to be susceptible to offline KGA from insider attacker.

Chen [25] proposed a dPEKS scheme without bilinear pairing that achieved keyword privacy, trapdoor privacy, and secure against offline KGA from outsider attacker and

TABLE 3. Comparison among PEKS schemes.

PEKS									
Scheme	**Tool**	**Hardness**	**Model**	**Search Function**	**Keyword Privacy**	**Trapdoor Privacy**	**Security against Online KGA**	**Security against Offline KGA**	
								Outsider	**Insider**
[3]	Pairing-based	BDH	Random Oracle	Single	✓				
[6]	Pairing-based	DBDH	Random Oracle	Conjunctive	✓				
[31]	*k*-resilient IBE	DDH	Standard	Conjunctive	✓				
[27]	Pairing-based	DLDH	Random Oracle	Conjunctive	✓				
[34]	Pairing-based	BDH	Random Oracle	Single	✓				
[45]	Pairing-based	BDDH	Random Oracle	Single	✓			✓	✓
[36]	Pairing-based	BDH	Random Oracle	Single	✓				
[33]	*k*-resilient IBE	DDH	Standard	Conjunctive	✓				
[46]	*k*-resilient IBE	DDH	Standard	Conjunctive	✓				
[42]	Pairing-based	DDH	Random Oracle	Single	✓	✓		✓	
[28]	Pairing-based	DBDH	Random Oracle	Fuzzy	✓			✓	✓
[29]	Pairing-based	DDH	Standard	Conjunctive	✓				
[43]	Pairing-based	CDDH, DBDH, DLIN	Random Oracle	Single	✓	✓		✓	
[16]	Pairing-based	BDH	Random Oracle	Single	✓				
[15]	-	-	Random Oracle	Single	✓	✓		✓	✓
[47]	Pairing-based	CDH, DBDH	Random Oracle	Single	✓	✓		✓	✓
[48]	prime-order elliptic curve group	CDH, DDH	Random Oracle	Single	✓	✓		✓	
[49]	Pairing-based	DBDH, mDLIN	Random Oracle	Conjunctive	✓	✓			✓
[50]	DT-PKC	-	-	Single & Multi	✓	✓			✓

online KGA. Chen *et al.* [37] proposed the first DS-PEKS scheme without bilinear pairing and satisfied keyword privacy and secure against offline KGA from insider attacker in the random oracle model. Their proposed scheme consists of two servers that run the test query. The front server first pre-processes the trapdoor and searchable ciphertext before forwarding to the back server. The back server then decides which documents are queried by the receiver. Their proposed scheme is secure against offline KGA from insider attacker based on the assumption that both servers do not collude with each other, which is difficult to prove in a real scenario. Their scheme also showed inefficiency in practice because of the need for two servers to carry out the trapdoor testing process. Chen *et al.* [38] proposed an SA-PEKS scheme based on a bilinear map and blind signature in the random oracle model that requires the user to query a semi-trusted third party (i.e., keyword server) to generate keyword ciphertext and trapdoor. Their proposed scheme satisfied keyword privacy and secure against online KGA and offline KGA from insider attacker. They also proposed a universal transformation framework from any PEKS scheme to a secure SA-PEKS scheme. Lee *et al.* [53] proposed a new SCF-PEKS scheme that achieved trapdoor privacy and secure against offline KGA form outsider attacker. Their scheme also has an authentication mechanism that protects the cloud service provider from being tricked by the attacker that sends fake ciphertext.

Lu *et al.* [54] presented cryptanalyses on the Fang *et al.* [23] and Shao and Yang [52] SCF-PEKS scheme. They showed that Fang *et al.*'s [23] scheme is vulnerable

TABLE 4. Comparison among dPEKS/SCF-PEKS schemes.

dPEKS/SCF-PEKS									
Scheme	**Tool**	**Hardness**	**Model**	**Search Function**	**Keyword Privacy**	**Trapdoor Privacy**	**Security against Online KGA**	**Security against Offline KGA**	
								Outsider	**Insider**
[14]	Pairing-based	n-BDHI	Random Oracle	Single	✓				
[8]	Pairing-based	BDH	Random Oracle	Single	✓				
[17]	Pairing-based	DBDH, q-ABDHE	Standard	Single	✓				
[11]	Pairing-based	1-BDHI, BDH	Random Oracle	Single	✓				
[51]	Pairing-based	CDH	Random Oracle	Single	✓				
[18]	Pairing-based	BDH, HDH, BDHI	Random Oracle	Single	✓	✓		✓	
[19]	Pairing-based	BDH, HDH, BDHI	Random Oracle	Single	✓	✓		✓	✓
[21]	Pairing-based	DLP, CDH, BDH	Random Oracle	Single	✓	✓		✓	
[23]	Pairing-based	DBDH, SXDH, truncated q-ABDHE	Standard	Single	✓	✓		✓	
[52]	Pairing-based	DBDH, SXDH, truncated q-ABDHE	Standard	Single	✓	✓		✓	
[24]	Pairing-based	DBDH, QDBDH, HDH	Standard	Single	✓	✓		✓	
[25]	Traditional Encryption	-	-	Single	✓	✓	✓	✓	
[37]	Smooth Projective Hash Function	DDH	Random Oracle	Single	✓	✓		✓	
[38]	Pairing-based, FDH-RSA blind signature	CDH, RSA known-target inversion	Random Oracle	Single	✓		✓		✓
[53]	Pairing-based	DLP, CDH, BDH	Random Oracle	Single	✓	✓		✓	
[54]	Pairing-based	DBDH, SXDH, truncated q-ABDHE	Standard	Single	✓	✓		✓	✓
[26]	Pairing-based	OMDL, ABDHE, gap DLIN	Standard	Single	✓	✓		✓	

to online KGA and Shao and Yang *et al.*'s [52] scheme is vulnerable to offline KGA from insider attacker. They further improved Fang *et al.*'s scheme [23] to be secure against offline KGA from insider attacker by embedding a secret in both searchable ciphertext and the trapdoor that is shared between the sender and the receiver. They claimed that their method of achieving security against offline KGA from insider attacker can be generically adopted by other existing PEKS or SCF-PEKS schemes.

C. PAEKS

Public key authenticated encryption with keyword search (PAEKS) is a variant of the PEKS scheme that allows the verifier to verify that the searchable ciphertext is generated by the sender. Table 5 shows a comparison of various PAEKS schemes. Huang and Li [39] proposed the first PAEKS scheme based on a bilinear map that satisfied keyword privacy and trapdoor privacy. Wu *et al.* [55] proposed a new PAEKS scheme that requires the sender to compute the authorisation token of a keyword using a receiver public key.

TABLE 5. Comparison among PAEKS schemes.

PAEKS									
Scheme	Tool	Hardness	Model	Search Function	Keyword Privacy	Trapdoor Privacy	Security against Online KGA	Security against Offline KGA	
								Outsider	Insider
[39]	Pairing-based	DBDH, mDLIN	Random Oracle	Single	✓	✓			
[55]	Trapdoor Permutation Function	CDH	Random Oracle	Single	✓	✓			✓
[56]	Pairing-based	DBDH	Random Oracle	Single	✓	✓		✓	✓
[57]	Pairing-based	DBDH, mDLIN	Random Oracle	Single	✓	✓			✓
[40]	Pairing-based	CBDH, CDH	Random Oracle	Single	✓	✓			✓
[44]	Elliptic Curved	DDH	Random Oracle	Single	✓	✓			✓
[58]	Pairing-based	BDH	Random Oracle	Fuzzy	✓	✓		✓	✓

The authorisation token is later used by the receiver to generate the trapdoor. Due to this mechanism, the proposed scheme is secure against offline KGA from insider attacker because the server cannot freely generate searchable ciphertext by itself. Li *et al.* [56] also proposed a PAEKS scheme based on a bilinear map. Their proposed scheme is more efficient than Huang and Li's [39] scheme in terms of the trapdoor generation algorithm and searchable ciphertext generation algorithm.

Noroozi and Eslami [57] found out that Huang and Li's [39] scheme was insecure against KGA in multiuser settings because of their proposed security model only considers two types of attackers namely, sender and receiver. Noroozi and Eslami [57] justified that the security model should also consider other users, as they may also be potential attackers to meet the practicality of multiusers in public key settings. They further improved the scheme to be secure against offline KGA from insider attacker in multiuser settings and satisfied keyword privacy and trapdoor privacy. Qin *et al.* [40] showed that Huang and Li's [39] scheme failed to capture the multi ciphertext attack in their security model, and they presented a new PAEKS scheme that satisfied keyword privacy and trapdoor privacy that can withstand multi ciphertext attack and the offline KGA from insider attacker. Lu and Li [44] noted that Huang and Li's [39] scheme is insecure against adaptive chosen target adversaries, which later improved the security notion to capture the adaptive chosen target attacks. Lu and Li [44] also proposed a lightweight PAEKS scheme that is bilinear pairing free and satisfies keyword privacy and trapdoor privacy in the random oracle model. They also claimed that their proposed scheme outperformed other existing pairing based PAEKS schemes.

Ma and Kazemian [58] proposed a new type of PAEKS scheme that integrates with the fuzzy logic technique to achieve fuzzy search functionality for their proposed PAEKS scheme. Their proposed scheme also satisfied keyword privacy, trapdoor privacy, and secure against offline KGA from both types of attackers.

IV. POTENTIAL RESEARCH DIRECTIONS

We draw potential research directions based on our observations in section III. Keyword guessing attack is a major weakness faced by the PEKS schemes. To achieve security against keyword guessing attack, some proposed PEKS schemes must tradeoff between security and efficiency of their schemes.

Before trapdoor privacy was introduced, all previous PEKS schemes guaranteed privacy only in ciphertext. Some research [18], [59] showed that the least requirement for a PEKS scheme to be secure against offline keyword guessing attack is to satisfy at least keyword privacy and trapdoor privacy.

As noted in section III, most PEKS schemes are based on bilinear pairing, which is computational expensive. IoT devices and smart devices with limited computationally resources are at the disadvantage of using these schemes. Therefore, it is interesting to explore the possibility of constructing a PEKS scheme without bilinear pairing that possesses both keyword privacy and trapdoor privacy and can withstand the keyword guessing attack.

Another possible research direction is to investigate the relationship between the security notions of trapdoor privacy, as presented in section II-C. If it is possible to establish concrete findings on these security notions, it would also be significant to explore the possibility of constructing a secure PEKS scheme in the standard model that satisfies the trapdoor privacy security notions proposed by Nishioka [42] or Arriaga *et al.* [43].

For search functionality, a single keyword search is the most adopted search function. A single keyword search allows only one keyword to perform a search operation at a time, which is a disadvantage of the PEKS scheme from the functionality aspect. It would also be interesting to explore the possibility of constructing a PEKS scheme that has other search functionalities, such as conjunctive, disjunctive, and fuzzy search, while preserving keyword privacy and trapdoor privacy.

V. CONCLUSION

The security properties of keyword privacy and trapdoor privacy are essential for the PEKS schemes to be secure against offline keyword attack from outsider attacker. In this paper, we have performed comparison analysis on various types of PEKS schemes. We have drawn some potential research directions for future research.

REFERENCES

[1] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2000, pp. 44–55.

[2] C. Bösch, P. Hartel, W. Jonker, and A. Peter, "A survey of provably secure searchable encryption," *ACM Comput. Surv.*, vol. 47, no. 2, pp. 1–51, Jan. 2015.

[3] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2004, pp. 506–522.

[4] S. Kamara and K. Lauter, "Cryptographic cloud storage," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2010, pp. 136–149.

[5] 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 *Proc. Workshop Secure Data Manage.* Berlin, Germany: Springer, 2006, pp. 75–83.

[6] D. J. Park, K. Kim, and P. J. Lee, "Public key encryption with conjunctive field keyword search," in *Proc. Int. Workshop Inf. Secur. Appl.* Berlin, Germany: Springer, 2004, pp. 73–86.

[7] W.-C. Yau, S.-H. Heng, and B.-M. Goi, "Off-line keyword guessing attacks on recent public key encryption with keyword search schemes," in *Proc. Int. Conf. Autonomic Trusted Comput.* Berlin, Germany: Springer, 2008, pp. 100–105.

[8] J. Baek, R. S. Naini, and W. Susilo, "Public key encryption with keyword search revisited," in *Proc. Int. Conf. Comput. Sci. Appl.* Berlin, Germany: Springer, 2008, pp. 1249–1259.

[9] I. R. Jeong, J. O. Kwon, D. Hong, and D. H. Lee, "Constructing PEKS schemes secure against keyword guessing attacks is possible?" *Comput. Commun.*, vol. 32, no. 2, pp. 394–396, Feb. 2009.

[10] W.-C. Yau, R. C.-W. Phan, S.-H. Heng, and B.-M. Goi, "Keyword guessing attacks on secure searchable public key encryption schemes with a designated tester," *Int. J. Comput. Math.*, vol. 90, no. 12, pp. 2581–2587, 2013.

[11] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Improved searchable public key encryption with designated tester," in *Proc. 4th Int. Symp. Inf., Comput., Commun. Secur.*, 2009, pp. 376–379.

[12] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, 2001, pp. 213–229.

[13] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions," in *Proc. Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, 2005, pp. 205–222.

[14] C. Gu, Y. Zhu, and H. Pan, "Efficient public key encryption with keyword search schemes from pairings," in *Proc. Int. Conf. Inf. Secur. Cryptol.* Berlin, Germany: Springer, 2007, pp. 372–383.

[15] L. Sun, C. Xu, M. Zhang, K. Chen, and H. Li, "Secure searchable public key encryption against insider keyword guessing attacks from indistinguishability obfuscation," *Sci. China Inf. Sci.*, vol. 61, no. 3, pp. 1–3, Mar. 2018.

[16] J. Zhang, C. Song, Z. Wang, T. Yang, and W. Ma, "Efficient and provable security searchable asymmetric encryption in the cloud," *IEEE Access*, vol. 6, pp. 68384–68393, 2018.

[17] L. Fang, W. Susilo, C. Ge, and J. Wang, "A secure channel free public key encryption with keyword search scheme without random Oracle," in *Proc. Int. Conf. Cryptol. Netw. Secur.* Berlin, Germany: Springer, 2009, pp. 248–258.

[18] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Trapdoor security in a searchable public-key encryption scheme with a designated tester," *J. Syst. Softw.*, vol. 83, no. 5, pp. 763–771, 2010.

[19] W. Bingjian, C. Tzungher, and J. Fuhgwo, "Security improvement against malicious server's attack for a dPEKS scheme," *Int. J. Inf. Educ. Technol.*, vol. 1, no. 4, pp. 350–353, 2011.

[20] H. S. Rhee, J. H. Park, and D. H. Lee, "Generic construction of designated tester public-key encryption with keyword search," *Inf. Sci.*, vol. 205, pp. 93–109, Nov. 2012.

[21] Y. Zhao, X. Chen, H. Ma, Q. Tang, and H. Zhu, "A new trapdoor-indistinguishable public key encryption with keyword search," *J. Wireless Mobile Netw. Ubiquitous Comput. Dependable Appl.*, vol. 3, nos. 1–2, pp. 72–81, 2012.

[22] W.-C. Yau, R. C.-W. Phan, S.-H. Heng, and B.-M. Goi, "Security models for delegated keyword searching within encrypted contents," *J. Internet Services Appl.*, vol. 3, no. 2, pp. 233–241, 2012.

[23] L. Fang, W. Susilo, C. Ge, and J. Wang, "Public key encryption with keyword search secure against keyword guessing attacks without random Oracle," *Inf. Sci.*, vol. 238, pp. 221–241, Jul. 2013.

[24] L. F. Guo and W.-C. Yau, "Efficient secure-channel free public key encryption with keyword search for EMRs in cloud storage," *J. Med. Syst.*, vol. 39, no. 2, pp. 1–11, Feb. 2015.

[25] Y.-C. Chen, "SPEKS: Secure server-designation public key encryption with keyword search against keyword guessing attacks," *Comput. J.*, vol. 58, no. 4, pp. 922–933, Apr. 2015.

[26] K. Emura, A. Miyaji, M. S. Rahman, and K. Omote, "Generic constructions of secure-channel free searchable encryption with adaptive security," *Secur. Commun. Netw.*, vol. 8, no. 8, pp. 1547–1560, 2020.

[27] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in *Proc. Int. Conf. Pairing-Based Cryptogr.* Berlin, Germany: Springer, 2007, pp. 2–22.

[28] P. Xu, H. Jin, Q. Wu, and W. Wang, "Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack," *IEEE Trans. Comput.*, vol. 62, no. 11, pp. 2266–2277, Nov. 2013.

[29] M.-S. Hwang, S.-T. Hsu, and C.-C. Lee, "A new public key encryption with conjunctive field keyword search scheme," *Inf. Technol. Control*, vol. 43, no. 3, pp. 277–288, 2014.

[30] Y. Lu, G. Wang, and J. Li, "On security of a secure channel free public key encryption with conjunctive field keyword search scheme," *Inf. Technol. Control*, vol. 47, no. 1, pp. 56–62, Mar. 2018.

[31] D. Khader, "Public key encryption with keyword search based on K-resilient IBE," in *Proc. Int. Conf. Comput. Sci. Appl.* Berlin, Germany: Springer, 2006, pp. 298–308.

[32] S.-H. Heng and K. Kurosawa, "k-Resilient identity-based encryption in the standard model," in *Proc. Cryptographers Track RSA Conf.* Berlin, Germany: Springer, 2004, pp. 67–80.

[33] H.-M. Yang, C.-X. Xu, and H.-T. Zhao, "An efficient public key encryption with keyword scheme not using pairing," in *Proc. 1st Int. Conf. Instrum., Meas., Comput., Commun. Control*, Oct. 2011, pp. 900–904.

[34] Q. Tang, "Revisit the concept of PEKS: Problems and a possible solution," Centre Telematics Inf. Technol., Univ. Twente, Enschede, The Netherlands, Tech. Rep. DTR08-9/TR-CTIT-08-54, 2008.

[35] J. Shao, Z. Cao, X. Liang, and H. Lin, "Proxy re-encryption with keyword search," *Inf. Sci.*, vol. 180, no. 13, pp. 2576–2587, 2010.

[36] W.-C. Yau, R. C.-W. Phan, S.-H. Heng, and B.-M. Goi, "Proxy re-encryption with keyword search: New definitions and algorithms," in *Security Technology, Disaster Recovery and Business Continuity*. Berlin, Germany: Springer, 2010, pp. 149–160.

[37] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "Dual-server public-key encryption with keyword search for secure cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 4, pp. 789–798, Apr. 2016.

[38] R. Chen, Y. Mu, G. Yang, F. Guo, X. Huang, X. Wang, and Y. Wang, "Server-aided public key encryption with keyword search," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 12, pp. 2833–2842, Dec. 2016.

[39] Q. Huang and H. Li, "An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks," *Inf. Sci.*, vols. 403–404, pp. 1–14, Sep. 2017.

[40] B. Qin, Y. Chen, Q. Huang, X. Liu, and D. Zheng, "Public-key authenticated encryption with keyword search revisited: Security model and constructions," *Inf. Sci.*, vol. 516, pp. 515–528, Apr. 2020.

[41] Y. Miao, Q. Tong, R. Deng, K.-K.-R. Choo, X. Liu, and H. Li, "Verifiable searchable encryption framework against insider keyword-guessing attack in cloud storage," *IEEE Trans. Cloud Comput.*, early access, Apr. 21, 2020, doi: 10.1109/TCC.2020.2989296.

[42] M. Nishioka, "Perfect keyword privacy in PEKS systems," in *Proc. Int. Conf. Provable Secur.* Berlin, Germany: Springer, 2012, pp. 175–192.

[43] A. Arriaga, Q. Tang, and P. Ryan, "Trapdoor privacy in asymmetric searchable encryption schemes," in *Proc. Int. Conf. Cryptol. Afr.* Berlin, Germany: Springer, 2014, pp. 31–50.

[44] Y. Lu and J. Li, "Lightweight public key authenticated encryption with keyword search against adaptively-chosen-targets adversaries for mobile devices," *IEEE Trans. Mobile Comput.*, early access, May 4, 2021, doi: 10.1109/TMC.2021.3077508.

[45] Q. Tang and L. Chen, "Public-key encryption with registered keyword search," in *Public Key Infrastructures, Services and Applications*. Berlin, Germany: Springer, 2009, pp. 163–178.

[46] W.-C. Yau, S.-H. Heng, S.-Y. Tan, B.-M. Goi, and R. C.-W. Phan, "Efficient encryption with keyword search in mobile networks," *Secur. Commun. Netw.*, vol. 5, no. 12, pp. 1412–1422, Dec. 2012.

[47] L. Wu, B. Chen, S. Zeadally, and D. He, "An efficient and secure searchable public key encryption scheme with privacy protection for cloud storage," *Soft Comput.*, vol. 22, no. 23, pp. 7685–7696, 2018.

[48] Y. Lu, J. Li, and F. Wang, "Pairing-free certificate-based searchable encryption supporting privacy-preserving keyword search function for IIoTs," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2696–2706, Apr. 2021.

[49] C. Xu, L. Mei, J. Cheng, Y. Zhao, and C. Zuo, "IoT services: Realizing private real-time detection via authenticated conjunctive searchable encryption," *J. Cyber Secur.*, vol. 3, no. 1, pp. 55–67, 2021.

[50] X. Liu, G. Yang, W. Susilo, J. Tonien, X. Liu, and J. Shen, "Privacy-preserving multi-keyword searchable encryption for distributed systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 3, pp. 561–574, Mar. 2021.

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

[52] Z. Shao and B. Yang, "On security against the server in designated tester public key encryption with keyword search," *Inf. Process. Lett.*, vol. 115, no. 12, pp. 957–961, Dec. 2015.

[53] C.-C. Lee, C.-T. Li, and S.-T. Chiu, "A secure trapdoor-indistinguishable public encryption scheme with keyword search for cloud storage service," *J. Adv. Comput. Netw.*, vol. 7, no. 1, pp. 1–6, 2019.

[54] Y. Lu, G. Wang, and J. Li, "Keyword guessing attacks on a public key encryption with keyword search scheme without random Oracle and its improvement," *Inf. Sci.*, vol. 479, pp. 270–276, Apr. 2019.

[55] L. Wu, B. Chen, K.-K. R. Choo, and D. He, "Efficient and secure searchable encryption protocol for cloud-based Internet of Things," *J. Parallel Distrib. Comput.*, vol. 111, pp. 152–161, Jan. 2008.

[56] H. Li, Q. Huang, J. Shen, G. Yang, and W. Susilo, "Designated-server identity-based authenticated encryption with keyword search for encrypted emails," *Inf. Sci.*, vol. 481, pp. 330–343, May 2019.

[57] M. Noroozi and Z. Eslami, "Public key authenticated encryption with keyword search: Revisited," *IET Inf. Secur.*, vol. 13, no. 4, pp. 336–342, Jul. 2019.

[58] Y. Ma and H. Kazemian, "Public key authenticated encryption with multiple keywords search using Mamdani system," *Evolving Syst.*, vol. 12, no. 3, pp. 687–699, Sep. 2021.

[59] S.-T. Hsu, C.-C. Yang, and M.-S. Hwang, "A study of public key encryption with keyword search," *Int. J. Netw. Secur.*, vol. 15, pp. 71–79, Mar. 2013.

KOON-MING CHAN is currently pursuing the Ph.D. degree in I.T. with the Faculty of Information Science and Technology, Multimedia University, Malaysia. His research interests include cryptography and information security.

SWEE-HUAY HENG received the Doctor of Engineering degree from the Tokyo Institute of Technology, Japan. She is currently a Professor with the Faculty of Information Science and Technology, Multimedia University, Malaysia. Her research interests include cryptography and information security. She was the Program Chair of ProvSec 2010, CANS 2010, and ISPEC 2019. She served as the technical program committee member of many international security conferences.

WEI-CHUEN YAU (Member, IEEE) received the B.S. and M.S. degrees from the National Cheng Kung University, Taiwan, and the Ph.D. degree from Multimedia University. He is currently an Associate Professor with the School of Computing and Data Science, Xiamen University Malaysia. He is also a Chartered Engineer (C.Eng.) and a Certified Information Systems Security Professional (CISSP). His research interests include cryptography, security protocols, machine learning, and network security. He was a General Co-Chair of Mycrypt 2016. He has also served as a Guest Editor for the *ETRI Journal* special issue on cyber security and AI.

SHING-CHIANG TAN received the B.Tech. (Hons.) and M.Sc. (Eng.) degrees from Universiti Sains Malaysia, Malaysia, in 1999 and 2002, respectively, and the Ph.D. degree from Multimedia University, Malacca, Malaysia, in 2008. He is currently a Professor with the Faculty of Information Science and Technology, Multimedia University. His current research interests include computational intelligence (artificial neural networks, evolutionary algorithms, fuzzy logic, and decision trees), and their applications, data classification, condition monitoring, fault detection and diagnosis, and biomedical disease classification. He was a recipient of the Matsumae International Foundation Fellowship, Japan, in 2010.