ABKS-PBM: Attribute-Based Keyword Search With Partial Bilinear Map

SHAWAL KHAN¹, SHAHZAD KHAN², MAHDI ZAREEI³, (Senior Member, IEEE), FAISAL ALANAZI⁴, NAZRI KAMA⁵, MASOOM ALAM¹, AND ADEEL ANJUM¹

¹COMSATS Institute of Information Technology, Islamabad 44000, Pakistan
²National University of Science and Technology (NUST), Islamabad 44000, Pakistan
³School of Engineering and Sciences, Tecnologico de Monterrey, Monterrey 64849, Mexico
⁴College of Engineering, Prince Sattam Bin Abdulaziz University, Al Khobar 11942, Saudi Arabia
⁵Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia, Kuala Lumpur 54100, Malaysia

Corresponding author: Faisal Alanazi (faisal.alanazi@psau.edu.sa)

This work was supported by the Deanship of Scientific Research at Prince Sattam bin Abdulaziz University, Alkhobar, Saudi Arabia.

ABSTRACT The way services offered by cloud computing gets its unprecedented and undisputed popularity, so its security concerns. Among them the storage as service model (SaaS) is of the forefront of these concerns. SaaS liberates individuals and enterprises from management of IT infrastructure and data centers to concentrate on their core business. Because of untrusted and out-of-premise architecture users are reluctant to outsource their personal and important data. Encryption before outsourcing addresses some of these issues but at the same time strips the data of its useful operation such as sharing and searching. Now to address this issue, the combination of keyword based searchable encryption (KSE) and attribute-based encryption (ABE) leads to an attribute-based keyword searching (ABKS). The resultant combined concept is capable of fine-grained search operation in the multi-owner/multi-user (M/M) setting. However, the underlying costly pairing operation and complex secret sharing mechanism of ABE makes it unsuitable in practical application for resource-limited devices. On top of it, in most of the existing ABKS schemes the size of the secret key and its associated pairing operation linearly expands to the number of attributes. This paper aims at presenting a novel ABKS scheme with pairing-free access verification and constant size secret key based on AND gate access structure and ciphertext-policy (CP) framework. The security of the proposed work is reduced to the standard Decisional Diffie-Hellman (DDH) assumption, and also collision free and error tolerant. Finally, the performance evaluation and experimental results shows that the proposed scheme improved the overall efficiency and communication overhead.

INDEX TERMS Attribute-based encryption, access control, searchable encryption.

I. INTRODUCTION

Nowadays cloud computing provides an attractive computing architecture, enabling the on-demand computing allocation and ubiquitous access over the Internet. This computing paradigm relieves individuals and enterprises from establishing their own IT infrastructure and proprietary data centers to focus on their core business. In practice, cloud computing comes into three architectures models, including the private, the public cloud, and the hybrid cloud. Public cloud is usually regarded as a better choice for individuals and organizations because of greater capital expenditure saving, flexibility and better customer support. Since, the trusting domain of public cloud storage servers is out of user control premises, and hence users are reluctant to outsource their private data to the public domain. Therefore, the protection of outsourcing data currently becomes a prime barrier to the wide adaptation of cloud services for broader range of applications [1], [2]. Generally speaking for protection and privacy of data, encryption is performed before storing it on untrusted cloud servers. Reducing privacy and security risks, encryption at the same time also strips the data off its useful operation; searching capabilities. A simple approach to address this dilemma is to first download all the ciphertext data and then perform decryption operation locally. But, this naive approach will incur huge computation, bandwidth and storage cost and is inefficient.

Keyword-based searchable encryption (KSE) [3] was introduced to address the above-mentioned problem. The KSE schemes make use of client/server models.
The data owner uploads their encrypted data to the untrusted remote server. Where one or more data users submits the search trapdoor corresponding to its intended content to the remote server. As a result, the server performs the searching operation over encrypted data on behalf of the data user. Currently there are four KSE architectures: One-to-one (O/O), also called symmetric searchable scheme (SSE); Many-to-One (M/O); One-to-Many (O/M); Many-to-Many (M/M). All of these schemes were intensively researched in [4]–[9]. However, they are either relay on the risky third party server or involve sophisticated key management, which burden both the data owner and data user with high computation overhead.

To counter the above mentioned problems, the combination of keyword-based searchable encryption (KSE) and attribute-based encryption (ABE) [10]–[13] comes into being. This combined concept resulted in an attribute-based keyword search (ABKS) technique, which equips the M/M setting with fine-grained search capability. However, when the underlying costly pairing operations and complex secret sharing mechanism of attribute-based encryption combine with the fine-grained search, which is itself an intense computation task, increases its complexity many folds. Furthermore, in most of the existing work the running time for an index generation and trapdoor generation scale with the number of attributes. This makes the resulting technique unsuitable for mobile computing devices, especially the mobile phones, which has limited resources such as memory and battery life. However, nowadays for most people mobile phones are the most essential cloud computing gadget. Therefore, the existing approach for searchable encryption using pairing operation in its primitive form may not be suitable, owing to its complexity and intense computation demand. Our proposed scheme liberates the end users from this intense bilinear operation. The resource rich cloud server requires only two bilinear operations for keyword matching. We term this concept as a partial bilinear map in this work, which is the prime contribution of this paper along with constant size secret keys. The benefit of having these features is two-fold: first, it reduces transmission on secure channels and second, it significantly minimizes the computation overhead.

II. RELATED WORK

Here, we give a brief overview of how the traditional ABE has evolved into the attribute based keyword searching (ABKS).

A. ATTRIBUTE-BASED SEARCHABLE ENCRYPTION

Sahai and water [14] were the first to introduce ABE for encryption and fine grained access control. This idea was based on identity based encryption (IBE) in which users email ID, phone number used as a set of expressive attributes and any user can be identified using these attribute sets. The ABE schemes are categorized into two variants: ciphertext-policy (CP-ABE) and key-policy (KP-ABE). Using CP-ABE the data owner embed access policy inside ciphertext and the private key of the end user is attached to the attribute set. Anyone can perform the decryption operation if his/her attributes matched with specified access policy. While in KP-ABE, private keys are attached with the access control policy, and ciphertext are attached with the attribute set [15].

The data owner in the searchable encryption (SE) scheme encrypts the keywords and documents before outsourcing it to the cloud service provider. Xiaoding Song et al. [4] was the first to introduce an SE scheme to allow data users to access the encrypted data stored on cloud servers. Liu et al. [16] presented a novel scheme to verify the result acquired from the cloud service provider using KP-ABSE, correctness and integrity of search result can be verified. Further the offline keyword guessing attack can be thwarted. This paper [17] introduced a ciphertext-policy attribute-based searchable encryption (CP-ABSE) scheme to attain access control policy along user revocation ability and also achieve efficient attribute-based searching capability. To enhance the efficiency, [18] proposed a new verifiable attribute-based search scheme to overcome the problem of curious cloud servers to verify the search result such as whether the server executed the operation faithfully, for access control this scheme was based on access tree. Though this scheme fails the sharing capability of encrypted data. For keyword updating and well organized data sharing Liang and Susilo [19] used proxy re encryption and attribute-based keyword scheme, using this integration, anyone can share their encrypted data among those who match the specified access policy.

Another scheme by Miao et al. [20] that has the capability to check the accuracy of search results over encrypted data. The scheme attains attribute-based priority tree and also fine-grained access control, that can control access on the same data. Furthermore the scheme prevents chosen keyword attack (CKA). However, the tree structure used in the scheme is computationally expensive. To support multiple user and multiple owners models and improving efficiency Sun et al. [9] scheme runs in linear search along fine-grained authentication at file level to provide scalability. The data owner has the capability to control most of the important operations of the cloud server as a result user revocation is computationally efficient. The scheme prevents CKA attack, however, not satisfactory for other security attacks.

Miao et al. [21] adopted CP-ABE technique to introduce a sample attribute-based keyword search on hierarchical data. Because the desirable requirements cannot be satisfied by the basic scheme in cloud, they proposed two improvements to support revocation and multi keyword search. However, they did not consider any attack models for the proposed scheme. In [22] proposed a privacy preserving scheme using CP-ABE technique including hidden policy and selective security is achieved in generic bilinear group model, the scheme is capable of preventing offline keyword guessing attack. Another scheme by [23] is based on KP-ABE, their scheme creates constant size user secret keys and trapdoor. It also performs pairing operations which vary in other schemes and is based on attributes attached with them. It also supports user revocation and delegation.
In [24] the scheme is capable of updating the access tree, named as dynamic policy which also creates constant size secret keys. The author also presents a multi-keyword scheme which has same features as the proposed scheme and also supports fast search and generates a constant size trapdoor. Wang et al. in [25] presented a scheme that is capable of verifying the searched keyword through integrating privacy preserving search technique and Message Authentication Code (MAC), homomorphic technique to get verifiability and privacy. The result can be verified without storing it locally to construct index and query as they used bit vector. Data users can encrypt their data then authentication is done using MAC. The encrypted data is then outsourced to a cloud using authenticated trapdoor, which can perform search operations.

Hence, with the introduction of bilinear pairing many unrealistic problem in the field of cryptography being solved [26]–[28]. One of the bottleneck of all the contemporary attribute-based keyword searching (ABKS) is that its efficiency is based on computation speed of bilinear pairing operations. As a result a large number of research has been directed toward the optimization of these operations [29]–[32]. A more direct way is to replace intense computational bilinear pairing operations with more light and efficient basic arithmetic operations. In this work, we achieve the implementation of this basic idea with the help of Fermat’s principle along with single a multiplicative group.

**B. MOTIVATION AND CONTRIBUTION**

Existing literature reveals that a large number of the contemporary Attribute-based Keyword Searching (ABKS) schemes use access tree or linear secret sharing scheme (LSSS) for their access control mechanism. While incorporating the access control policy, secret sharing scheme is the de-facto standard for these schemes, which are dependent on costly bilinear pairing operations. Furthermore, for their practical realization the number of pairing operations increases linearly with the attributes increase, hence, they always demand intensive computing resources to execute the encryption and decryption algorithms. This frequent pairing operations present a new challenge is how to incorporate wireless resource constrained devices, especially light weight sensors and mobile phones, into the cloud computing. This motivates us to develop a pairing free control access mechanism for ABKS. Our contribution is based on Fermat’s principle [33]. In this work our contribution can be stated as follows:

1) Our constructed (ABKS-PBM) scheme is free from costly bilinear pairing operations for access control mechanisms. The design still inherits all the expected advantages of (ABKS) architecture including fine-grained data access control and security.

2) The proposed scheme reduces the transmission overhead of the secret channel as trusted attribute authority (TAA) needs to securely send one constant-size secret key for each user instead of one against each data user (DU) attribute. To our knowledge, aggregation of secret key into one item is the first of its types in the model of (ABKS) schemes.

3) The proposed scheme also avoids access tree and LSSS matrix, pairing-based construction for extraction of secret keys from the specified set of attributes of the data users.

4) The security proof is given in more preferable security assumptions, namely Decisional Bilinear Diffie–Hellman assumption (DBDH) instead of hypothetical random oracle model (ROM).

5) Also the detailed performance analysis demonstrates the better computational efficiency, especially for resource-constraint devices.

**III. PRELIMINARIES**

This section presents discussion about the cryptographic assumptions and techniques being utilized in the design of our proposed scheme.

**A. BILINEAR MAP**

Let $G_1$ and $G_2$ be two multiplicative cyclic groups of identical order $p$, having generator $g$. We define a bilinear map is a map $e : G_1 \times G_2 \rightarrow G_T$ with the following three conditions:

1) **Bilinear.**
   - For every $u, v \in G_1$ and $a, b \in \mathbb{Z}_p^*$, $e(u^a, v^b) = e(u, v)^{ab}$ holds.
   - For every $u, v, y \in G_1$, $e(uv, y) = e(u, y)e(v, y)$ holds.
   - For every $u, v, y \in G_1$, $e(u, vy) = e(u, v)e(u, y)$ holds.

2) **Computable.**
   - For $u, v \in G_1$, there exists an efficient algorithm for computing $e(u, v) \in G_2$.

3) **Non-degenerate.**
   - $e(g, g) \neq 1$.

**B. ACCESS STRUCTURE**

Definition 1 (Access Structure): Let $U = \{\text{att}_1, \text{att}_2, \ldots, \text{att}_n\}$ be the set of attributes. An access structure is a set of collection $A \subseteq 2^{\{\text{att}_1, \text{att}_2, \ldots, \text{att}_n\}} \setminus \emptyset$. An access structure is monotone if $\forall X, Y : if X \in Z$ and $X \subseteq Y$ then $Y \in Z$. When we confine the access structure to the threshold setting, it is termed as threshold access structure.

**C. SECURITY ASSUMPTIONS**

Definition 2 (Computational Diffie–Hellman Assumption): It is computationally intractable for a probabilistic polynomial time (PPT) adversary $A$, to compute the value of $g^{ab}$ from a given tuple $(g, g^a, g^b)$, where $g$ is a randomly chosen generator of a cyclic group $\mathbb{G}$ of order $p$ and $a, b \in \{0, \ldots, p - 1\}$. The advantage $\epsilon$ of an adversary $A$, to solve the CDH assumptions is given as

$$\Pr[(G, p, g, g^a, g^b) = g^{ab}] \geq \epsilon.$$
Definition 3 (Decisional Bilinear Diffe-Hellmen (DBDH) Assumption): For PPT adversary \(A\), it is computationally intractable to distinguish between the tuple \((g^a, g^b, g^c, e(g, g)^{abc})\) and the tuple \((g^a, g^b, g^c)\), for a given bilinear map tuple \((G_1, G_2, p, e, g)\), where \(G_1\) and \(G_2\) are two multiplicative groups with order \(p\) and three random elements \((a, b, c) \in \mathbb{Z}_p^3\). To solve the DBDH assumptions the advantage \(\epsilon\) of an adversary \(A\) is given as follows:

\[
|\Pr[A(g^a, g^b, g^c, e(g, g)^{abc})] - \Pr[A(g^a, g^b, g^c)]| \geq \epsilon.
\]

Definition 4: For a group \(G_p\), let \(p\) be a prime number, \(k\) be any integer in \(\mathbb{Z}_p\), \(\phi(p)\) be a totient function of \(G\), we have

\[
ge^{k \cdot \phi(p) + 1} = g \mod p.
\]

This is deduced directly from the Fermat’s little theorem, which can be defined as: for any positive integer \(a\), that is not divisible by \(p\), then

\[
a^{p-1} \equiv 1 \mod p \Rightarrow (a^{p-1})^k \equiv 1^k \mod p \Rightarrow (a^{p-1})^k \equiv a \equiv 1 \mod p \Rightarrow a^{k \cdot \phi(p) + 1} \equiv a \mod p.
\]

We exploit this concept in our proposed ABKS-PBM scheme in multiplicative group.

IV. SYSTEM MODEL AND SECURITY DEFINITION

A. SYSTEM MODEL

Proposed ABKS-PBM system model is shown in Fig 1, which comprises four participants: data owner (DO), data user (DU), cloud service provider (CSP) and trusted attribute authority (TAA). The trusted attribute authority role is to generate and issue credentials public key PK or secret key SK to interested users. These credentials are sent over open and secure communication channel respectively. Multiple DO encrypt the data files along with their corresponding keyword set so that TAA can repeatedly ask for the outsourced data according to some specific trapdoor, it submit it to the TAA. After the TAA checks whether the DU attribute satisfy the chosen set of attributes of DO and the keyword index matches to the trapdoor, retrieves and send the encrypted data file to the interested DU.

In our threat model, we take the CSP to be a trusted entity but curious about user personal information. The CSP will honestly run the protocol but attempt to infer some additional personal details from the data available to it. Most of the previous work on secure search over encrypted data also employed this assumption. Besides DO, DU and TAA are considered to be fully trusted entities.

B. SECURITY MODEL

Similar to basic ABKS, the proposed scheme should satisfy data confidentiality and collision resistance in addition to the privacy of the index keyword and query trapdoor. Thus, the privacy of the DO and DU in terms of their respective data are protected. We formalize the security of ABKS-PBM by following a given security game between an adversary \(A\) and challenger \(C\).

Definition 5: Our proposed ABKS-PBM scheme provide security against chosen-keyword attack (CKA) to win the CKA game if the advantage of an adversary \(A\) is negligible.

After challenger \(C\) publishes the public parameters, \(A\) defines the challenge set of attributes \(W^*\). \(A\) can repeatedly asks for a number of key-extraction queries \(k_A, k_A^2, \ldots, k_A^n\) of an attribute set \(w_1, w_2, \ldots, w_n\) and for ciphertext \(I_{w_1}, I_{w_2}, \ldots, I_{w_m}\) of corresponding index keyword \(w_1, w_2, \ldots, w_m\). The only restriction is that none of the extracted private keys satisfies the challenge set of attributes \(W^*\). \(A\) sends two challenge index keywords \(w_0\) and \(w_1\) to \(C\). Which flips a fair binary coin \(b \in \{0, 1\}\) and compute the ciphertext \(W_b\), which is submit to \(A\). \(A\) can repeatedly asks for a number of key-extraction \(k_A, k_A^2, \ldots, k_A^n\) corresponding to the attribute set \(w_{n+1}, w_{n+2}, \ldots\) and none of them satisfies \(W^*\). \(A\) submits b’s guess \(b’\). In this game, we define the advantage of \(A\) by

\[
Adv_{\text{cka}}^{\text{cpa}}(A) = |Pr(b = b') - \frac{1}{2}|
\]

If \(Adv_{\text{cka}}^{\text{cpa}}(A)\) is negligible, hence our proposed ABKS-PBM keyword encryption is also secure against CKA-CPA attack model.

Definition 6: The trapdoor generation algorithm is secure against Trapdoor Recoverable Attack (TRA) in eavesdropper attack model.

Adversary \(A\) query the challenger \(C\) corresponding ciphertext. Then \(A\), submits two challenge keywords \(q_0\) and \(q_1\) to \(C\), which has never been submitted earlier. \(C\) flips a fair binary coin \(b \in \{0, 1\}\) and compute ciphertext of \(q_b\) and sends it to \(A\). \(A\) can repeatedly asks \(C\) for any ciphertext of keywords except for keywords \(q_1\) and \(q_0\). Finally, \(A\) submits \(b\)’s guess \(b’\). We define the advantage of \(A\) by

\[
Adv_{\text{tra}}^{\text{evs}}(A) = |Pr(b = b') - \frac{1}{2}|
\]

If \(Adv_{\text{tra}}^{\text{evs}}(A)\) is negligible. Hence we say that our proposed trapdoor encryption is secure against trapdoor recoverable attack in eavesdropper attack model.

Definition 7: Our proposed scheme ensures that if many unauthorized data user integrate their secret keys for the decryption of ciphertext they cannot succeed in decryption, if none of them can individually succeed in decrypting it in polynomial time \(T\).
V. SYSTEM OVERVIEW AND CONSTRUCTION

Here, we first present system level view of our scheme then give description of each algorithm in detail.

A. SYSTEM OVERVIEW

This section presents an abstraction of our ABKS-PBM scheme as follows:

Definition 8: Propose scheme construction consists of five polynomial time algorithms as follows:

• Setup (λ) → (PK, SK). This algorithm is invoked by trusted attribute authority (TAA). It takes as input the security parameter λ while output the authority public key PK and master key MK.
• KeyGen (MK, S_u) → (SK_u). This algorithm is run by TAA and generates a secret key SK_u for DU according to an attribute set S_u. It takes as input the authority master key MK along with a data user set of attributes S_u and output data user secret key SK_u.
• EncInd (T, PK, w) → (\mathcal{I}_w). This algorithm is executed at the data owner side. The algorithm takes keyword w as input and TAA public key PK, encrypt it under chosen set of attribute T and output w’s ciphertext \mathcal{I}_w.
• TokenGen (SK_u, q) → \mathcal{T}_u(q). This algorithm is invoked by a data user. It takes as input the query keyword q and u’s secret key SK_u and output a trapdoor \mathcal{T}_u(q).
• Search (T, \mathcal{T}_u(q), CT) → \{0, 1\} Cloud server executes this algorithm. It takes the data owner ciphertext CT and data user encrypted index keyword \mathcal{T}_u(q). It outputs 1 if u’s attribute set S_u satisfy the chosen set of attributes T and \mathcal{T}_u.q = CT. \mathcal{I}_w simultaneously otherwise output 0.

B. CONSTRUCTION

This section present working mechanism of each algorithm of our proposed ABKS-PBM scheme.

1) SETUP PHASE

This phase is implemented in the system setup algorithm, which is run by the TAA. It will create a running environment for ABKS-PBM scheme. It takes a security parameter λ and a set of attributes \( U = 1, 2, \ldots, n \in \mathbb{Z}^* \) as input. It defines a symmetric bilinear group \( \mathbb{G} \) with order \( p \), which is a \( k \)-bit prime with generator \( g \) and \( H : \{0, 1\}^* \rightarrow \mathbb{Z}^*_p \) as a one-way hash function. Then, chooses randomly \((a, b, y, t_1, t_2, \ldots, t_n) \in (\mathbb{Z}^*)^{n+3}\) and finally the authority set PK and MK as follows

\[
PK = \{g^y, e(g, g)^a, g^b, Y = g^y, T_1 = g^{t_1}, T_2 = g^{t_2}, \ldots, T_n = g^{t_n}\} \\
MSK = \{a, b, y, t_1, t_2, \ldots, t_n\} \text{ for all } i \in U.
\]

2) KeyGen PHASE

This phase is implemented in attribute-based secret key generation algorithm, run by TAA to generates private key SK for a given data user according to an attribute set S. First the authority randomly chooses \( R_i \leftarrow \mathbb{Z}^*_p, \forall i \in S \) such that

\[
\sum_{i=1}^{\lfloor |S| / |T| \rfloor} (R_i t_i) = y \times \sum_{i=1}^{\lfloor |S| / |T| \rfloor} t_i
\]

After that, it compute and sets

\[
R' = [(p - 1) n + 1], \text{ where } n \in \mathbb{Z}^*.
\]

Then, the decryption key is computed as

\[
SK = \{D, D', D''\}, \quad \text{where } D = g^{R_t h}, \quad D' = \frac{R_i}{\sum_{i=1}^{\lfloor |S| / |T| \rfloor} t_i} \quad \text{and } \quad D'' = \frac{R'}{\sum_{i=1}^{\lfloor |S| / |T| \rfloor} t_i}, \quad \forall i \in S.
\]

3) EncInd PHASE

This phase is called in the encrypt index algorithm, invoked by the DO to generate a secure index keyword for searching. The DO randomly chooses \( K \in \mathbb{Z}^* \) as a symmetric key to encrypt its data file \( E_k(\text{File}) \) using symmetric encryption (AES etc) cipher. Then it masks the K by multiplying it with the common secret \( g^{y+\sum S_i} \), which is only re-constructable at the DU side having valid attributes as specified by the DO in its policy.

4) TokenGen PHASE

This phase is called in the token generation algorithm, executed by the DU. The DU compute the token \( tok_1 = g^{b H(w)} \), \( tok_2 = g^a \) and send the \( \mathcal{T}_u(q) = (tok_1, tok_2, \{D\}) \) to the untrusted cloud server.
Algorithm 2 KeyGen
Input: DU attributes set $S$ and authority master secret key $MSK$.
Output: DU key $SK$.
1. Set $Y \leftarrow 0$ and $T \leftarrow 0$
2. for each attribute $i \in S$ do
3. Compute $Y \leftarrow Y + t_i$
4. Set $T = Y$
5. end for
6. Choose and set $(R_i)$ such that $\sum_{i=1}^{s} (R_i t_i) = Y \times y$.
7. Randomly choose $n \leftarrow \mathbb{Z}_{p}^*$
8. Compute and set $R' \leftarrow (p - 1)n + 1$
9. for each attribute $i \in S$ do
10. Compute $D_i \leftarrow$ Exponentiation $(g, R_i t_i)$
11. Set $D = D \cup D_i$
12. Compute $D_i \leftarrow \frac{R'}{T_i}$
13. Set $D' = D' \cup D_i$
end for
14. Compute $D'' \leftarrow \left( \frac{D'}{T_i} \right)$
15. Set $SK = \{ D, D', D'' \}$
16. return $SK$

Algorithm 3 Encind
Input: Trusted authority public key $PK$, DO chosen set of attributes $T$ and index keyword $w$.
Output: Encrypted authority public key ciphertext $CT$.
1. Choose at random $r, s, K \in \mathbb{Z}_{p}^*$
2. Calculate $Y_s \leftarrow$ Exponentiation $(g, s)$.
3. Compute $COM_s = Y_s \times Y$.
4. Compute $K_{enc} = COM_s \times K$.
5. for each attribute $i \in T$ do
7. Compute $E_i \leftarrow$ Exponentiation $(T_i, s)$
8. Set $E = E \cup E_i$
end for
10. Set $A_{verf} \leftarrow (Y)^s$
11. Compute $\mathcal{T}_w = e(g', g^{bH(w)})$, $\mathcal{T}_w = g^{as}$
12. return $CT = \{ T, \mathcal{T}_w, \mathcal{T}_w, K_{enc}, A_{verf}, E \}$

Algorithm 4 TokenGen
Input: DU secret key component and query keyword $q$.
Output: $q$’s token query $T_u(q)$.
1. Compute $tok1 = g^{bH(q)}$ and $tok2 = g^a$
2. Set $T_u(q) = \{ tok1, tok2, [D'] \}$
2. return $T_u(q)$

5) SEARCH PHASE
The cloud server invokes this phase in ABKS-PBM search algorithm. The CS performs the search over an outsourced encrypted index according to the submitted token by the DU, without gaining any useful information about either of them. In response to the interested DU, the CS will return the encrypted data, if and only if the two conditions (i) the DU’s attribute set $S$ satisfies the chosen set of attribute $T$ by the DO and (ii) the query token is equal to the index keyword, true simultaneously.

On other hand, there may arise two situations where the CS returns $0$ to DU. One is that the DU is not authorized to perform searching on encrypted data, technically, the DU attribute set $S$ does not satisfy the chosen attribute set $T$, during which the algorithm will terminate in advance. The other is, when the DU meets the access criteria, however the query keyword is not the same as the index keyword.

Algorithm 5 Search
Input: Index keyword $T_w$ and token $T_u(q)$ submitted by $D_u$.
Output: 1, if $S$ satisfies $T$ and $w = q$, 0 otherwise.
1. if $(T \cap S) \geq d$ then
2. Compute $A_{verf}' = \prod_{i \in S} (E_i)^{D'_i}$
3. if $(A_{verf} == A_{verf}')$ then
4. if $(T_{w} == T_{u}(q))$ then
5. return $1$
6. else
7. return $0$
8. else
9. return $0$
10. else
11. return $0$
12. end if

6) DecPHASE
This phase is called in the decryption algorithm of ABKS-PBM, DU runs this phase for the retrieval of symmetric key $K$. The retrieved key $K$ is then used by receiver to decrypt $D_k(File)$.

Algorithm 6 Decryption
Input: DU key $SK$ and ciphertext $CT$
Output: Symmetric key $K$.
1. Set $y' = 1$
2. for each attribute $i \in S$ do
3. Compute $Y' \leftarrow (Y' \times E_i \times D_i)$
end for
5. Compute $COM_s \leftarrow$ Exponentiation$(Y', D'')$
6. Compute $K \leftarrow (K_{enc}/COM_s)$
7. return $K$

VI. ABKS-PBM ANALYSIS
A. CORRECTNESS ANALYSIS
We first analyze the correctness of access authorization of DU’s searching with respect to the DO outsourced index keyword $w$. CS needs to find out whether $S$ satisfies $T$. The CS computes $\prod_{i \in S} (E_i)^{D_i}$, to check out whether it constitutes the same access verification $A_{verf}$ as set by the data owner in ciphertext

$$A_{verf} = \prod_{i \in S} (E_i)^{P_i}$$
We now analyze the matching conformation between an encrypted index $T_w$ and its corresponding token $tok_w$.

Algorithm 5 tells us that after the access authorization, the CS needs to find out whether $w$ is equal to $q$ or not by check out whether equation 2 is true or not.

$$
RHS = \prod_{i \in S} (g^{h_{x_i}^{\nu}})^{E_i} = g^{\sum_{i \in S} (h_{x_i}^{nu} \times E_i)} = g^{\sum_{i \in S} \times E_i} = g^{\sum_{i \in S} \times \nu} = g^{\nu x y} \tag{1}
$$

$$
A_{vrf} = LHS
$$

We present the features of our proposed ABKS-PBM scheme, we theoretically analyze and compare it with Hui Yin et al. CP-ABSE scheme [34]. The reason for selection is its convincing performance along with suitability for resource constraint devices. The necessary notations we used in this comparison are defined in Table 1.

For clarity, we tabulate each algorithm construction cost and output size for ABKS-PBM and CP-ABSE in Table 2 and Table 3, individually. It is important to note that we are not considering the less time consuming functions such as hash function, basic arithmetic operation; multiplication, addition and subtraction in $\mathbb{Z}^n$. In case of successful search query, its transmission cost is also ignored, thus in both cases the output size of search algorithm is set to zero.

Table 3 shows that CP-ABSE costs the least in the setup phase as its computation but also communication overhead of the secret channel. More precisely suppose the central authority has $K$ number of users, where each user having $t$ number of attributes, then TAA needs to transmit $(K \times t)$ elements through secure channel and the computation time $(K \times t \times E_1)ms$.

On the other hand the TAA of our proposed scheme requires to transmit only $K$ random group elements through secure channel and the computation time is $(K \times Z_q^*)ms$. Thus, significantly minimizes the communication overhead.

### C. COMMUNICATION OVERHEAD ANALYSIS

In our scheme, the TAA needs to send a constant-size secret key for each DU through secure secret channel rather than one for each attribute, which significantly reduces not only its computation but also communication overhead of the secret channel. More precisely suppose the central authority has $K$ number of users, where each user having $t$ number of attributes, then TAA needs to transmit $(K \times t)$ elements through secure channel and the computation time $(K \times t \times E_1)ms$.

### D. SECURITY ANALYSIS

1) **INDISTINGUISHABILITY UNDER CHOSEN-PLAINTEXT ATTACK MODEL (IND-CPA)**

Here in this section, we provide the security proofs of ABKS-PBM in chosen plaintext model reduces to the hardness problem of the Decisional Bilinear Diffie-Hellman (DBDH) assumption.

**Theorem 1:** If Decisional BDH is an intractable problem, then our constructed keyword encryption algorithm provide security against chosen-keyword attack (CKA) in chosen plaintext attack model (CKA-CPA).

**Proof:** Let there is an adversary $A$, who can attack ABKS-PBM index keyword encryption with a non-negligible advantage $\epsilon$, then there exist a simulator $B$ for solving the decisional BDH problem with a non-negligible advantage $\epsilon / 2$. The simulation proceed as follows:

We let the challenger $C$ generate a group $\mathbb{G}$ with generator $g$. The challenger $C$ then flips a fair binary coin $\mu \in \{0, 1\}$
TABLE 2. Output and computation complexity of ABKS-PBM.

| Algorithm  | Computation Complexity | Output Complexity |
|------------|------------------------|-------------------|
| Setup      | $P + |U|E_{G_1} + 3Z_p^*$  | $2 |U || G_1$       |
| KeyGen     | $2 |S|E_{G_1} + Z_q^*$    | $2 |G_1 | + | Z_q^*$ |
| EncInd     | $(|T|+5)E_{G_1} + P + 4M_{G_1} + H + Z_q^*$ | $(2 |T|+2) |G_1 | + | G_2$ |
| TokenGen   | $2E_{G_1} + H$         | $(2 + |X|) |G_1$       |
| Search     | $|N|G_1 + 2P$          | 0                 |

TABLE 3. Output and computation complexity of CP-ABSE in [34].

| Algorithm  | Computation Complexity | Output complexity |
|------------|------------------------|-------------------|
| Setup      | $P + 2E_{G_1} + 2Z_q^*$ | $Z_q^* + 3 || G_1 || G_2$ |
| KeyGen     | $|S|H' + (2 |S|+4)E_{G_1} + (|S|+1)M_{G_1} + (|S|+1)Z_q^* + V_{Z_q}^* + M_{Z_q}^*$ | $(2 |S|+2) || G_1$ |
| EncInd     | $Z_q^* + M_{Z_q}^* + (2 |T|+2)E_{G_1} + P + H + E_{G_2} + M_{G_2} + |T|H'$ | $(2 |T|+1) || G_1 || G_2$ |
| TokenGen   | $|S|E_{G_1} + |S|M_{G_1} + H$ | $(2 |S|+1) || G_1$ |
| Search     | $(2 |N|+1)P + |N|E_{G_2} + (|N|+1)M_{G_2} + (|N|+1)V_{G_2}$ | 0                 |

outside of $B$’s view, send $t_\mu$ to the simulator $B$. The selection of random elements $(a, b, c, z) \in Z^*$ for the outcome of $\mu$ is given below.

- Case 1: if $\mu = 0$, then $C$ sets input $t_\mu$ of algorithm $B$ as $(a, b, c, z) = (g^a, g^b, g^c, e(g, g)^{abc})$.
- Case 2: if $\mu = 1$, then $C$ sets input $t_\mu$ of algorithm $B$ as $(a, b, c, z) = (g^a, g^b, g^c, e(g, g)^z)$.

**Init:** adversary $A$ chooses an attributes set $W^*$ that it wishes to be challenged upon.

**Setup:** $B$ sets the public parameter $Y = e(A, B) = e(g, g)^{ab}$ and gives it to $A$.

**Phase 1:** $A$ can repeatedly asks for a number of key-extraction queries $k_{A_1}^w, k_{A_2}^w, \ldots, k_{A_n}^w$ of attribute sets $w_1, w_2, \ldots, w_n$ and the ciphertext $r_{w_1}, r_{w_2}, \ldots, r_{w_m}$ of index keyword $w_1, w_2, \ldots, w_m$. These key-extraction and ciphertext returned by $B$ need to satisfy the following conditions:

- All the extracted private keys from the set of attributes $w_1, w_2, \ldots, w_n$ do not satisfies the challenge set of attributes $W^n; |S| = (|W| \cap W^n) < d$
- All the extracted private keys can be used to valid trapdoor.

**Challenge:** $A$ sends two challenge index keywords $w_0$ and $w_1$ to $B$. The simulator $B$ flips a fair binary coin $v \in \{0, 1\}$ and computes the ciphertext is output as:

$$T_{w_b} = (W^*, (T_{w_b}^*)^e = e(B^{H(w_0)}A^z), (T_{w_b}^*)^e = e(g^{bH(w_0)}, g^a e(g, g)^{abc}, (T_{w_b}^*)^e = g^a, (E_{i}^*) = (E_{i})_{v \in W^*})$$

if $\mu = 0$, then $Z = e(g, g)^{abc}$ and $t_0$ is sent as input to $B$ and $T_{w_b}^* = (W^*, (T_{w_b}^*)^e = e(g^{bH(w_0)}, g^a e(g, g)^{abc}, (T_{w_b}^*)^e = g^a, (E_{i}^*) = (E_{i})_{v \in W^*})$ Since $r$ and $\alpha$ are randomly chosen from $Z_p$ in the index keyword generation, we let $\alpha = \alpha$ and $b = r$, the ciphertext can be denoted as

$$T_{w_b} = (W^*, (T_{w_b}^*)^e = e(g^{bH(w_0)}, g^a e(g, g)^{abc}, (T_{w_b}^*)^e = g^a, (E_{i}^*) = (E_{i})_{v \in W^*})$$

if $\mu = 1$, then $Z = e(g, g)^z$ and $t_1$ is sent to $B$ and the ciphertext output as

$$T_{w_b} = (W^*, (T_{w_b}^*)^e = e(g^{bH(w_0)}, g^a e(g, g)^{abc}, (T_{w_b}^*)^e = g^a, (E_{i}^*) = (E_{i})_{v \in W^*})$$

Since the $Z$ is random, the ciphertext contain no information about index keyword, $T_{w_b}$ will be a random element of target group from the view of $A$.

$A$ outputs its guess $b'$ of $b$, if $b' = b$ then the simulator will output $\mu's$ guess $\mu' = 0$, to indicate that the
challenger C sends a valid BDH-tuple, to B. The adversary’s advantage to recover \( H(\cdot) \) from \( T^*_{\text{nb}} \) is \( \epsilon \) by definition. Therefore the probability in this situation is \( \frac{1}{2} + \epsilon \). If \( b' \neq b \) the simulator will output \( \mu' \)’s guess \( \mu' = 1 \), to indicate that the challenger C sends a random \( 4-\text{tuple} \) to B and the possibility in this case is \( \frac{1}{2} \). For solving the decisional BDH problem the advantage of C as follows:

\[
\frac{1}{2} \Pr[\mu' = \mu | \mu = 0] + \frac{1}{2} \Pr[\mu' = \mu | \mu = 1] - \frac{1}{2} = \left( \frac{1}{2} \right)^2 + \epsilon + \frac{1}{2} \left( \frac{1}{2} \right)^2 - \frac{1}{2} = \epsilon
\]

**Theorem 2:** If discrete logarithm DL problem is hard to compute, our constructed token generation algorithm is secure against trapdoor recoverable attack (TRA) in eavesdropping attack model.

**Proof:** With the help of security game between the challenger C and adversary A, the proof of the above theorem is as follows:

**Phase 1:** A query the challenger C for multiple keywords \( q_1, q_2, \ldots, q_n \), where \( 1 \leq i \leq n \). In response to A, the challenger C sends the following ciphertext:

\[
T_A(q_i) = (T = g^{bH(w_i)}, \{D_i\}_i \in S), \quad \text{where S represents BFFs A’s attribute set.}
\]

**Challenge:** A sends two challenge keywords \( q_0 \) and \( q_1 \) to C, which have never been submitted earlier. C flips a fair binary coin \( b \in \{0, 1\} \) and compute

\[
T_A(q_b) = (T = g^{bH(w)}, \{D_i\}_i \in S)
\]

and sends it to adversary A.

**Phase 2:** A acts the same as it did in Phase 1, except for keyword \( q_0 \) and \( q_1 \)

**Guess:** A outputs its guess \( b' \) of \( b \). By definition of security game, A can’t access the encryption oracle, it cannot efficiently compute \( T_A(q_1) \) and \( T_A(q_0) \) without knowing \( b \). Given that the discrete log problem (DL) is intractable, the correct guess \( b' = b \) probability for A, is at most \( \frac{1}{2} \).

**Theorem 3:** If Computational Diffe-Hellman (CDH) problem is intractable, our proposed scheme is secure against collusion attack. That is, even if one or more than one malicious DU collude to decrypt the ciphertext by integrating more than one secret key or attributes, they are unable to succeed if none of them can independently succeed in decryption.

**Proof:** For decryption of ciphertext, malicious DU must obtain the attribute secret keys for the corresponding ciphertext. Apparently, the most probable attack scenario, is the collusion attack (i.e., an integration of attributes or secret keys) as there may be some overlapping set of attributes among the malicious DUs’.

Let us assume that some DU’s have already got their secret keys corresponding to their respective attribute set. Now, they want to collude against some ciphertext which is encrypted under some attribute \( S' \) that intersects with their respective attribute set. More specifically, that they possess some individual decryption keys which constitute the required decryption key corresponding to the common attribute in the form of \( D_i = g^{R_{ji}} \). Now, even though they themselves construct the secret key against the required set of attributes, still they are not capable of decryption as their individuals \( R_i \)’s are randomly chosen to meet the equation

\[
\sum_{i=1}^{\left| S \right|} (R_{ti}) = y \times \sum_{i=1}^{\left| S \right|} t_i.
\]

Given that the CDH problem is intractable, malicious DU’s will never construct \( y \times \sum_{i=1}^{\left| S \right|} t_i \) because of their individual random selection for \( R_i \). Hence, the proposed scheme is secure against the collusion attack under the CDH problem assumption.

2) **INDISTINGUISHABILITY UNDER CHOSEN-CIPTERTEXT ATTACK MODEL (IND-CCA)**

The security proof of the proposed scheme is given in chosen-plaintext attack in a selective model. The security of the proposed scheme can be further enhanced to IND-CCA by incorporating the one-time-signature scheme of [35] in its security proofs. This transformation from CPA-secure ABKS-PBM scheme to CCA-secure ABKS-PBM scheme requires two more algorithms, namely sign algorithm S and verifier algorithm V in our basic construction. We also need a sign key \( S_k \) and verification key \( V_k \) from TAA. The sign algorithm generates a signature \( \sigma \) while taking the message \( m \) and signing key \( S_k \) as input. The verifier algorithm V takes \( m \), \( V_k \), and \( \sigma \) as input, and outputs its guess \( b \in \{0, 1\} \).

Accordingly, the security game between an adversary A and challenger C proceeds as follows. First the challenger runs \( \text{keyGen} \) algorithm and returns to \( A \) the signing key \( S_k \) and the verifying key \( V_k \). Adversary A chooses \( m^* \) a signing query that it wishes to be challenged upon. The challenger computes \( \sigma^* = S(m^*, S_k) \) and gives it to \( A \).

A outputs its guess \( (\sigma, m) \). We say \( A \) sends the valid forge message \( m \) if \( V(m, \sigma, V_k) = 1 \), while \( (m^*, \sigma^*) \neq (m, \sigma) \). In other words, the adversary \( A \) will be given access to the decryption algorithm to decrypt randomly chosen ciphertext. However, the signature of the ciphertext will make the adversary \( A \) unable to temper with the ciphertext. For more insight the reader may refer to [35], [36].

**E. PERFORMANCE ANALYSIS**

This section briefly presents experimental results and performance evaluation of our proposed ABKS-PBM scheme. The hardware execution environment is a CPU: Intel Core I3 4005 with 4GB RAM, with Ubuntu 14. The software execution environment is a Charm-crypto version 0.42 and Spyder 2.2.5 is used to evaluate the performance of each algorithm such as Setup, KeyGen, EnclInd and Trapdoor generation.

1) **EVALUATION OF STORAGE COST**

To compare the storage cost of our proposed scheme, we implemented the existing CP-ABKS scheme proposed in [34]. The following figure 2 demonstrates the storage

---

**Figure 2**

Storage cost comparison between our scheme and existing CP-ABKS scheme.

---

For a detailed analysis of the storage cost, refer to [34].
cost. We reduced the storage size of KeyGen by generating a constant-size key for data owner, EncInd and Trapdor algorithm, used to generate keyword query by data user. The size of the Setup algorithm takes more space in ABKS-PBM scheme, however, in practice it is acceptable since Setup phase is a one time cost and runs on resource rich TAA.

2) EVALUATION OF KeyGen ALGORITHM
KeyGen algorithm associates the attribute list with the secret key for a particular data user by the TAA. Figure 3 shows the execution time of KeyGen algorithm. We can see that the execution time of both algorithms are linearly proportional to the associated attribute list for corresponding data users. We observe that our proposed scheme requires lesser execution time when we increase the number of attributes, since the proposed scheme consists of lesser exponentiation operation as compared to CP-ABSE scheme.

3) EVALUATION OF EncInd ALGORITHM
The EncInd algorithm generates the encrypted keyword index, which is related to the data user access control policy. This encrypted version of keyword makes the data user search operation possible on cloud server (CS). Figure 4 demonstrates the time cost taken by the CP-ABSE and our proposed algorithm to encrypt one index keyword. Both algorithms take linear time as the number of chosen set of attributes by the data user increases, however, we observe that because of lesser computation overload our ABKS-PBM performs efficiently.

4) EVALUATION OF TRAPDOR ALGORITHM
The Trapdor algorithm is executed by the authorized data user to generate a query trapdor for its interested keyword. The following figure 5 shows the time cost of running Trapdor algorithms for CP-ABSE and our proposed scheme. The time cost of CP-ABSE scheme is linearly proportional to the number of data user attributes while the proposed scheme has constant computation overhead, as it has nothing to do with the data user attribute list.

5) EVALUATION OF SEARCH ALGORITHM
The cloud server (CS) in response to the data user submitted query trapdor performs search operation on its stored index keywords to find its matched keyword. The average running time for search algorithm is depicted in figure 6. We run both algorithms for $|N| = 2$ and $|N| = 4$, where $N$ represents the minimum attribute that is associated with the given access control policy. We can see that the running time for both algorithms linearly expends with the minimum attribute set $N$ satisfying an access control policy and the number of index keywords. Free from costly bilinear pairing operation in access control conformation, which are utmost expensive operations, our scheme achieves better time cost for
search operation. Besides, searching is the most frequently performed operation in ABKS, hence its efficiency improves the overall system performance and makes it suitable for resource constrained devices.

VII. CONCLUSION

This paper proposed a ABKS-PBM scheme with pairing-free access verification and constant size users secret key under the hardness of Decisional Diffe-Hellman (DDH) assumption. The proposed scheme uses exponentiation operations and Fermats theorem for keyword index and trapdoor generation with fine grained access control searching capability. The security analysis proved our proposed scheme security and the experimental result indicate not only its low computation overhead but also its low communication overhead. As a future work, we would like to add functionality like verifiable search operation and make it more expressive by allowing support for OR and NOT gates.

REFERENCES

[1] X. Zheng, Y. Zhou, Y. Ye, and F. Li, “A cloud data deduplication scheme based on certificateless proxy re-encryption,” J. Syst. Archit., vol. 102, Jan. 2020, Art. no. 101666.

[2] P. Golle, J. Staddon, and B. Waters, “Secure conjunctive keyword search over encrypted data,” in Proc. Int. Conf. Appl. Cryptogr. Netw. Secur., Springer, 2004, pp. 31–45.

[3] 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.

[4] D. Xiaoding Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in Proc. IEEE Symp. Secur. Privacy. S&P, May 2000, pp. 44–55.

[5] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in Proc. Int. Conf. Theory Appl. Cryptograph. Techn., Springer, 2004, pp. 506–522.

[6] B. Balamurugan and P. V. Krishna, “Extensive survey on usage of attribute based encryption in cloud,” J. Emerg. Technol. Web Intell., vol. 6, no. 3, pp. 263–272, 2014.

[7] P. Wang, H. Wang, and J. Pieprzyk, “An efficient scheme of common secure indices for conjunctive keyword-based retrieval on encrypted data,” in Proc. Int. Workshop Inf. Secur. Appl., Springer, 2008, pp. 145–159.

[8] J. Shi, J. Lai, Y. Li, R. H. Deng, and J. Weng, “Authorized keyword search on encrypted data,” in Proc. Eur. Symp. Res. Comput. Secur., Springer, 2014, pp. 419–435.

[9] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, “Protecting your right: Verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud.” IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 4, pp. 1187–1198, Apr. 2016.

[10] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in Proc. 13th ACM Conf. Comput. Commun. Secur. CCS, 2006, pp. 89–98.

[11] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in Proc. IEEE Symp. Secur. Privacy (SP), May 2007, pp. 321–334.

[12] V. Goyal, A. Jain, O. Pandey, and A. Sahai, “Bounded ciphertext policy attribute based encryption,” in International Colloquium on Automata, Languages, and Programming, Springer, 2008, pp. 579–591.

[13] Y.-L. Tan, B.-M. Goh, R. Komiya, and S.-Y. Tan, “A study of attribute-based encryption for body sensor networks,” in Proc. Int. Conf. Informat. Eng. Inf. Sci., Springer, 2011, pp. 238–247.

[14] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in Proc. Annu. Conf. Theory Appl. Cryptograph. Techn., Springer, 2005, pp. 457–473.

[15] F. Han, J. Qin, H. Zhao, and J. Hu, “A general transformation from KP-ABE to searchable encryption,” Future Gener. Comput. Syst., vol. 30, pp. 107–115, Jan. 2014.

[16] P. Liu, J. Wang, H. Ma, and H. Nie, “Efficient verifiable public key encryption with keyword search based on KP-ABE,” in Proc. 9th Int. Conf. Broadband Wireless Comput., Commun. Appl., Nov. 2014, pp. 584–589.

[17] J. Har and D. K. Noh, “Attribute-based access control with efficient revocation in data outsourcing systems.” IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 7, pp. 1214–1221, Jul. 2011.

[18] Q. Zheng, S. Xu, and G. Atieniese, “VABKS: Verifiable attribute-based keyword search over outsourced encrypted data,” in Proc. IEEE INFOCOM Conf. Comput. Commun., Apr. 2014, pp. 522–530.

[19] K. Liang and W. Susilo, “Searchable attribute-based mechanism with efficient data sharing for secure cloud storage,” IEEE Trans. Inf. Forensics Security, vol. 10, no. 9, pp. 1981–1992, Sep. 2015.

[20] Y. Miao, J. Ma, Q. Jiang, X. Li, and A. K. Sangaiah, “Verifiable keyword search over encrypted cloud data in smart city.” Comput. Electr. Eng., vol. 65, pp. 90–101, Jan. 2018.

[21] Y. Miao, X. Liu, K.-K.-R. Choo, R. H. Deng, J. Li, H. Li, and J. Ma, “Privacy-preserving attribute-based keyword search in shared multi-owner setting,” IEEE Trans. Depend. Secur. Comput., early access, Feb. 9, 2019, doi: 10.1109/TDSC.2019.2967575.

[22] Y. Miao, J. Ma, X. Liu, X. Li, Q. Jiang, and J. Zhang, “Attribute-based keyword search over hierarchical data in cloud computing,” IEEE Trans. Services Comput., vol. 13, no. 6, pp. 985–998, Nov./Dec. 2017.

[23] H. Zhu, L. Wang, H. Ahmad, and X. Niu, “Key-policy attribute-based encryption with equality test in cloud computing.” IEEE Access, vol. 5, pp. 20428–20439, 2017.

[24] Mamta and B. B. Gupta, “An efficient KP design framework of attribute-based searchable encryption for user level revocation in cloud.”Concurrency Comput., Pract. Exper., vol. 32, no. 18, Sep. 2020, Art. no. e5329.

[25] Z. Wan and R. H. Deng, “VPSearch: Achieving verifiability for privacy-preserving multi-keyword search over encrypted cloud data,” IEEE Trans. Depend. Secur. Comput., vol. 15, no. 6, pp. 1083–1095, Nov. 2018.

[26] A. Joux, “A one round protocol for tripartite Diffie–Hellman,” J. Cryptol., vol. 17, no. 4, pp. 263–276, Sep. 2004.

[27] D. Boneh, B. Lynn, and H. Shacham, “Short signatures from the Weil pairing,” J. Cryptol., vol. 17, no. 4, pp. 297–319, Sep. 2004.

[28] D. Boneh and M. Franklin, “Identity-based encryption from the Weil pairing.” IEEE Trans. Inf. Forensics Security, vol. 10, no. 9, pp. 1981–1992, Sep. 2015.

[29] J.-L. Beuchat, J. E. González-Díaz, S. Mitsunari, E. Okamoto, F. Rodríguez-Henríquez, and T. Teruya, “High-speed software implementation of the optimal ate pairing over barreto–naehrig curves,” in Proc. Annu. Int. Cryptol. Conf. Pairing, Springer, 2014, pp. 549–565.

[30] P. S. L. M. Barreto, S. D. Galbraith, C. O. Hêgeardtaj, and M. Scott, “Efficient pairing computation on supersingular abelian varieties,” Designs, Codes Cryptogr., vol. 42, no. 3, pp. 239–271, Mar. 2007.

[31] S. Canard, J. Devigne, and O. Sanders, “Delegating a pairing can be both secure and efficient,” in Proc. Int. Conf. Appl. Cryptogr. Netw. Secur., Springer, 2014, pp. 549–565.

[32] M. Guillevic and D. Vergnaud, “Algorithms for outsourcing pairing computation,” in Proc. Int. Conf. Smart Card Res. Adv. Appl., Springer, 2014, pp. 193–211.

[33] A. Karati, R. Amin, and G. P. Biswas, “Provably secure threshold-based ABE scheme without bilinear map,” Arabian J. Sci. Eng., vol. 41, no. 8, pp. 3201–3213, Aug. 2016.

[34] H. Yin, J. Zhang, Y. Xiong, L. Ou, F. Li, S. Liao, and K. Li, “CP-ABE: A ciphertext-policy attribute-based searchable encryption scheme,” IEEE Access, vol. 7, pp. 5682–5694, 2019.
SHAWAL KHAN received the bachelor’s degree in computer science from Shaheed Benazir Bhutto University, Peshawar, Pakistan. He is currently pursuing the M.S. degree in information security from the COMSATS Institute of Information Technology, Islamabad, Pakistan. His research interests include access control, cryptography, and network security.

SHAHZAD KHAN received the B.S. degree in computer science from Malakand University, and the M.S. degree in computer and communication security from the School of Electrical Engineering and Computer Science (SEECS), National University of Science and Technology (NUST), Islamabad, Pakistan. He is currently pursuing the Ph.D. degree with the Military College of Signals (MCS), NUST. He is also an Assistant Professor of computer science with Shaheed Benazir Bhutto University, Sheringal. His research interests include applied cryptography and information security.

MAHDI ZAREEI (Senior Member, IEEE) received the M.Sc. degree in computer network from the University of Science, Malaysia, in 2011, and the Ph.D. degree from the Communication Systems and Networks Research Group, Malaysia-Japan International Institute of Technology, University of Technology, Malaysia, in 2016. He joined the School of Engineering and Sciences, Tecnologico de Monterrey, as a Postdoctoral Fellow, in 2017, where he has been as a Research Professor, since 2019. His research interests include wireless sensor and ad-hoc networks, energy harvesting sensors, information security, and machine learning. He is a member of the Mexican National Researchers System (level I). He is also serving as an Associate Editor for IEEE Access and the Ad Hoc and Sensor Wireless Networks journals.

FAISAL ALANAZI received the B.Sc. degree in electrical engineering (electronics and communication) from KSU, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from The Ohio State University, in 2013 and 2018, respectively. He is currently an Assistant Professor with Prince Sattam Bin Abdulaziz University. His research interests include cryptography, vehicular ad-hoc networks, and delay tolerant networks. He is a member of the IEEE Communication Society.

NAZRI KAMA received the B.Sc. degree (Hons.) in management information system and the M.Sc. degree in realtime software engineering from the Universiti Teknologi Malaysia, in 1999 and 2001, respectively, and the Ph.D. degree from The University of Western Australia, Perth, WA, Australia, in 2010, specializing in software engineering in the areas of requirements engineering. He is an Active Researcher with the Software Engineering Laboratory. He is currently an Associate Professor and the Deputy Dean Research and Innovation of the Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia. He has been successfully leveraging his expertise in various software development projects.

MASOUM ALAM received the Ph.D. degree in computer sciences from the University of Innsbruck, Austria. He is currently an Associate Professor with the Department of Computer Sciences at COMSATS Institute of Information Technology, Islamabad, Pakistan. His research interests include access control systems, model driven architecture, and work flow management systems.

ADEEL ANJUM received the Ph.D. degree in computer sciences from the University of Innsbruck, Austria. He is currently an Associate Professor with the Department of Computer Sciences at COMSATS Institute of Information Technology, Islamabad, Pakistan. His research interests include access control systems, model driven architecture, and work flow management systems.