Lightweight Searchable Encryption Protocol for Industrial Internet of Things

Ke Zhang, *Member, IEEE,* Jiahuan Long, Xiaofen Wang, Hong-Ning Dai, *Senior Member, IEEE*, Kaitai Liang, *Member, IEEE,* and Muhammad Imran, *Senior Member, IEEE*

***Abstract*—Industrial Internet of Things (IoT) has suffered from insufficient identity authentication and dynamic network topology, thereby resulting in vulnerabilities to data confidentiality. Recently, the attribute based encryption (ABE) schemes have been regarded as a solution to ensure data transmission security and the fine-grained sharing of encrypted IoT data. However, most of existing ABE schemes that bring tremendous computational cost are not suitable for resources-constraint IoT devices. Therefore, lightweight and efficient data sharing and searching schemes suitable for IoT applications are of great importance. To this end, we propose a light searchable attribute based encryption scheme (namely LSABE). Our scheme can significantly reduce the computing cost of IoT devices with the provision of multiple-keyword searching for data users. Meanwhile, we extend the LSABE scheme to multi-authority scenarios so as to effectively generate and manage the public/secret keys in the distributed IoT environment. Finally, the experimental results demonstrate that our schemes can significantly maintain computational efficiency and save the computational cost at IoT devices, compared to other existing schemes.**

I. Introduction

INDUSTRIAL Internet of things (IoT) is a new paradigm reshaping modern industries via connecting various devices with IoT gateways, access points and base stations, consequently achieving ubiquitous data acquisition and sharing across the diverse industrial sectors. The widespread employment of IoT nodes bring an enormous amount of data. The analytics on massive IoT data can further drive the developments of fault detection, disaster forecasting, service improvement and intelligent decision making.

When collecting enormous data, IoT devices suffer from the limited storage capacity and computational capability, thus needing an intermediary agent to manage and store data effectively. Fortunately, the emergence of industrial cloud and edge computing technologies compensates for these deficiencies of IoT devices. Particularly, local edge computing nodes can collect massive data from various IoT devices, then preprocess and transmit the data to remote cloud servers which can then offer data analytics services for data users, such as assembly line operators, technicians and enterprise managers [1].

However, industrial IoT-cloud systems improve the efficiency of data analytics while bringing a number of challenges especially in the security aspect. The first challenge is privacy preservation. Sensitive IoT data can be hacked or leaked by malicious users due to the wide proliferation of IoT devices, which are exposed to the unprotected environment. In order to protect data confidentiality, sensitive data needs to be encrypted in the end-to-end manner [2, 3]. Second, the flexible access control plays a vital role in data sharing since IoT-cloud systems are no longer confined to one-to-one authorization but one-to-many [4, 5]. Therefore, how to efficiently authorize devices in a group of users is another challenge. To solve the above problems in industrial IoT-cloud systems, attribute based encryption (ABE) was introduced in industrial IoT-cloud system.

A. Motivation

The ciphertext-policy attribute based encryption (CP-ABE) schemes have recently been used in industrial IoT-cloud system to secure the data privacy while meanwhile providing a flexible access control mechanism. In a CP-ABE, the plaintext is encrypted by a set of predefined attributes associated the public key, and the ciphertext can be decrypted only when the attributes in the secret key match the access strategy in the ciphertext. Such fine-grained access control is also applicable to IoT-cloud systems. In summary, a CP-ABE scheme, as a one-to-many cryptographic system, is well-suited for distributed IoT nodes in an unprotected environment, consequently securing data confidentiality and achieving the flexible access control in IoT-cloud systems.

Although the attribute based encryption (ABE) schemes have been regarded as a solution to ensure the data transmission security and the fine-grained sharing of encrypted industrial IoT data, they also need to overcome new application obstacles in IoT-cloud systems. 1) *Resource-limited IoT devices.* IoT devices suffer from the limited storage capacity and computational capability. Computationally-complex encryption algorithms cannot be adopted at IoT devices due to

This work was supported in part by the Sichuan Science and Technology Program under Grant 2019YFG0405, in part by the Natural Science Foundation of China under Grant U1833122 and in part by the Macao Science and Technology Development Fund under Grant 0026/2018/A1. Imran's work is supported by the Deanship of Scientific Research at King Saud University through research group project number RG-1435-051.

K. Zhang and X. Wang are with School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (email: kezhang@uestc.edu.cn; xfwang@uestc.edu.cn).

J. Long is with School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (email: jhlong@std.uestc.edu.cn).

H.-N. Dai is Faculty of Information Technology, Macau University of Science and Technology, Macau SAR (email: hndai@ieee.org).

K. Liang is is with Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Netherlands (email: k.liang-3@tudelft.nl).

M. Imran is is College of Applied Computer Science, King Saud University, Riyadh, Saudi Arabia (email: dr.m.imran@ieee.org).

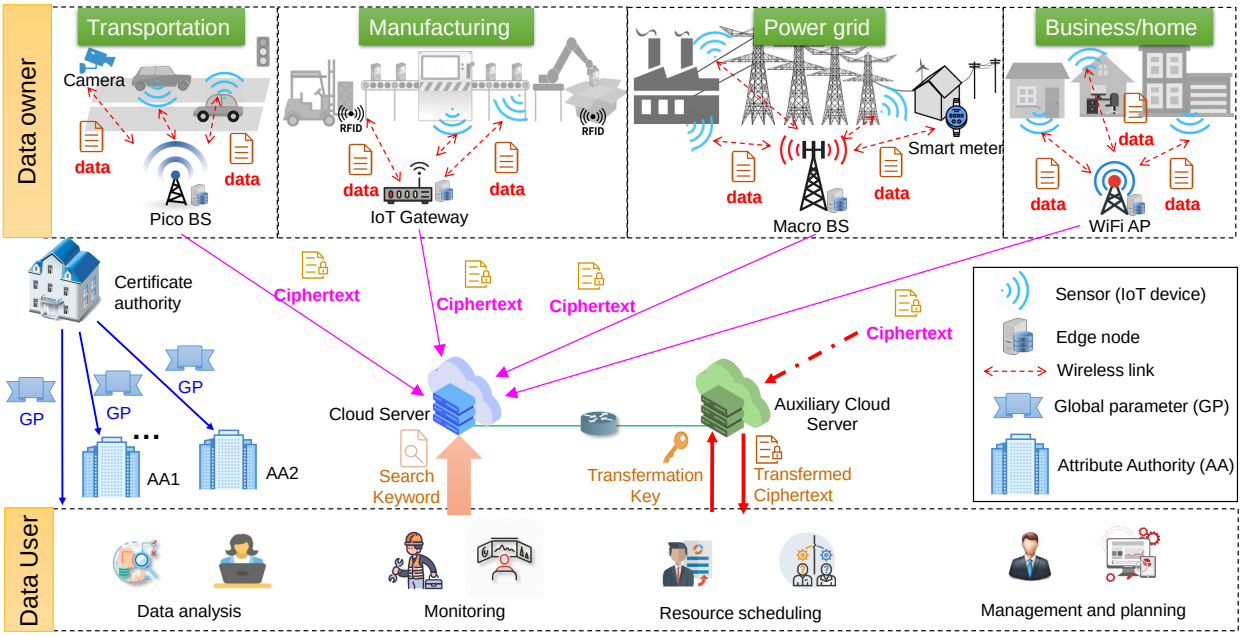


Fig. 1. Overview of our LSABE-MA scheme, which consists of six entities: data owner (IoT devices), certificate authority, attribute authority (AA), cloud server, auxiliary cloud Server and data user. LSABE-MA can be applied to diverse industrial applications such as power grid, transportation, manufacturing and business.

the poor computational capability while simplified encryption algorithms may be easily cracked by malicious users. 2) *Difficulty in encrypted-data retrieval at cloud servers.* Diverse types of ciphertexts that are stored at cloud servers need to be searched and located efficiently. However, the *unreadability* of encrypted files restrains the flexibility and precision of data retrieval, resulting in irrelevant or wrong searching results. 3) *Absence of effective key management.* In traditional ABE algorithms, authorization of all attributes is controlled by a central authority (CA). The single attribute authority cannot effectively generate and manage the public/secret keys for enormous number of IoT devices. In addition, once CA is compromised, all previously encrypted files can be leaked.

To address the above challenges, a novel lightweight searchable encryption scheme is necessary to be developed for industrial IoT-cloud systems.

B. Main contributions

Motivated by Green's key-blinding ABE scheme [6] and Yang's scheme [7], we propose a lightweight searchable ABE (namely LSABE) scheme. Meanwhile, in order to adopt the multi-keyword search in a distributed IoT environment with the provision of the decentralized key management, we devise a lightweight searchable ABE with multi-authority (namely LSABE-MA) for industrial IoT-cloud systems. Fig. 1 shows an overview of our LSABE-MA scheme (details will be illustrated in Section IV). In contrast to existing encryption schemes in industrial IoT-cloud systems, both LSABE and LSABE-MA have the following advantages.

1) Flexible access control: our schemes can ensure the flexible access control for authorized users to access data. In our LSABE-MA, any entity can be abstracted into one or more attributes (e.g., region, institution, professional title, function). Through these different attributes, we can define a variety of access control (AC) policies in an industrial IoT-cloud system, such as sending information to a specific type of sensors in a specific area, or providing sensitive data access to data analysts in a specific department.

2) Data Confidentiality: Our encryption algorithm can protect sensitive data against both curious operators and malicious users with the provision of the secure transmission of control information in an industrial IoT-cloud system.

3) Lightweight decryption algorithm for IoT devices: Our schemes relieve the tremendous computing burdens at resource-constrained IoT nodes. Specifically, we outsource the main computing tasks in the decryption process to the clouds without losing data confidentiality. IoT devices only need to compute one exponentiation operation to recover the message without conducting complex bilinear pairs operations.

4) Accurate data retrieval for IoT: Our schemes leverage a trapdoor-match mechanism, under which both plaintexts and keywords representing key features of plaintexts are encrypted and transmitted to cloud servers. IoT terminals can then match these encrypted files using the trapdoors generated by user-defined keywords. Our schemes also support multi-keyword search on the clouds, thereby achieving more precise and accurate searching.

5) Decentralized and scalable key management: Our LSABE-MA can separately generate and manage the public/secret keys for IoT devices, thereby avoiding delegating absolute trust to a CA, which may be corrupted in an unprotected IoT environment. In addition, LSABE-MA also breaks the conventional predefined attribute threshold in CA. Once a new attribute authority (AA) distributes secret/public keys to IoT devices, it does not invalidate the keys as in the past since there is no cooperation between AAs. This dedicated design can realize the secure and dynamic key management in IoT-cloud systems. Therefore, it is quite feasible to industrial IoT.

TABLE I. Comparison with other representative schemes

Schemes	Fine-grained AC	Multi-keyword search	Lightweight decryption	MA environment	Test platform
[8]	√	×	√	×	Mobile phone
[9]	√	×	√	×	PC
[10]	√	√	×	×	PC
[11]	√	√	×	×	PC
[12]	√	×	√	×	Mobile phone, PC
[13]	√	×	×	√	PC
[14]	√	×	√	×	PC
[15]	√	×	×	×	PC
[16]	×	×	√	√	PC
Our schemes	√	√	√	√	PC, Mobile phone, Raspberry Pi models

6) Real-world evaluation: Our schemes can be applied to large-scale industrial systems with mobile micro-controllers. We conduct extensive experiments on a real industrial environment (i.e., a production line of a workroom). Moreover, we consider different types of devices, such as two types of Raspberry Pi models and mobile phones. Compared with other existing ABE algorithms, our schemes demonstrate outstanding performance in terms of running time and storage costs, thereby revealing the practicability and efficiency in realistic industrial IoT environment.

C. Related Work

We next briefly review the recent advances in data security schemes in IoT. Bethencourt's scheme [17] is a popular CP-ABE scheme to achieve the fine-grained AC, in which the ciphertext can be decrypted when the attributes of the secret key match the access strategy in ciphertext. However, the highly complex CP-ABE algorithms like Bethencourt's scheme cannot be directly adopted for resource-limited IoT devices. In particular, during the decryption phase of encryption/decryption schemes, a substantial number of bilinear pairing operations need to be conducted, consequently overloading the IoT devices. To decrease the bilinear pairing operations of authorized users, Green et al. [6] proposed a lightweight CP-ABE scheme, in which the main computing task is done at the remote clouds and the remainder is completed at the user end. Consequently, the computing overhead of decryption can be greatly decreased while the data confidentiality can still be well protected. Since Green's seminal work, substantial efforts have been conducted to design more efficient lightweight ABE schemes such as [8, 9, 14, 16, 18]. Yao et al. [18] proposed a lightweight AC Elliptic Curve Cryptosystem (ECC) for IoT systems. Later, Amin and Kumar [16] developed a lightweight authentication protocol, which has a low complexity scheme while ensuring data security.

Although the lightweight IoT schemes can preserve privacy and security of IoT data, they also pose another problem, i.e., how to conduct data retrieval over a large number of ciphertexts. Therefore, several searchable encryption (SE) proposals were proposed to support encrypted data retrieval. For example, Boneh et al. [19] constructed the groundbreaking public key encryption with the provision of keyword searching (PEKS), which enables users to securely retrieve the desired files over encrypted data using user-defined keywords. Recently, studies [15, 20, 21] investigated the integration of ABE schemes with SE schemes. However, these methods can only be used to search for a single keyword; it restrains the flexibility and precision of data retrieval. In order to achieve more precise data retrieval, Miao et al. [10] proposed an improved ABE scheme with multi-keyword search so as to support simultaneous numeric attribute comparison, thereby greatly enhancing the flexibility of ABE encryption in dynamic IoT environment. Moreover, Yang et al. [7] proposed a traceable and lightweight ABE scheme to solve the problem of abusing secret key. Furthermore, attribute based multi-keyword search schemes have also investigated in [11, 12, 22]. Nevertheless, these CP-ABE schemes inevitably concentrate on the single authority environment, in which the authorization of all attributes is essentially controlled by a CA. The single authorization cannot effectively generate and manage the public/secret keys in industrial IoT.

In contrast, our LSABE and LSABE-MA schemes can support both single keyword and multi-keyword searching while maintaining the lightweight decryption. Moreover, our schemes can decentralize the key management and overcome the difficulty of distributed access control. Therefore, our schemes are quite feasible to practical industrial IoT environment. Table I compares our schemes with other representative schemes in perspectives of fine-grained AC, multi-keyword search, lightweight decryption, multi-authority (MA) and practical testing platforms.

The rest of this paper is organized as follows. Sections II and III give the definition and construction of LSABE scheme, respectively. Sections IV and V then introduce the overall design and construction of LSABE-MA scheme, respectively. Section VII next presents the experimental results. Finally, Section VIII concludes this paper.

II. Definition of LSABE Scheme

We present the definitions of LSABE scheme as follows.

Setup $(\kappa) \rightarrow (MSK, PP)$. Given the security parameter κ, Setup algorithm outputs the master secret key denoted by MSK and public parameters denoted by PP.

SecretKeyGen $(MSK, S, PP) \rightarrow SK$. Given the data owner's attribute set s, key generation center (KGC) conducts the SecrekeyGen algorithm and outputs the secret key SK.

Encrypt $(M, KW, (A, \rho), PP) \rightarrow CT$. Given keyword set KW extracted from file M and the access policy (A, ρ), data owner outputs ciphertext CT, which contains the secure index I and the encrypted file C_M.

Trapdoor $(SK, KW', PP) \rightarrow T_{KW'}$. Given the secret key SK, a query keyword set KW', data user runs the Trapdoor algorithm and outputs the trapdoor $T_{KW'}$.

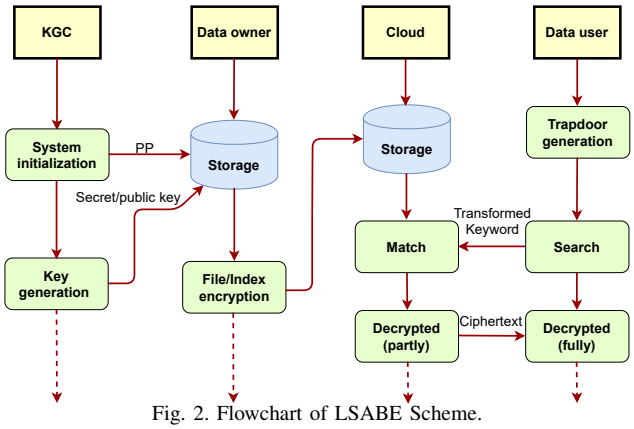


Fig. 2. Flowchart of LSABE Scheme.

TransKeyGen $(SK, z) \rightarrow TK$. Data user conducts the TransKeyGen algorithm, which takes the secret key SK as input and a blind value z and output the transformation key TK. Then, the $(T_{KW'}, TK)$ is be uploaded to the cloud server.

Search $(CT, T_{KW'}) \rightarrow 0/1$. The cloud server takes the trapdoor $T_{KW'}$ and the ciphertext CT as input, and executes the search algorithm. If the output is "0", the query fails. If the output is "1" , the query is successful and the cloud servers continue to run the transform algorithm.

Transform $(CT, TK) \rightarrow CT_{out}/\perp$. Given the transformation key TK, the cloud server can transform the ciphertext into a partially decrypted ciphertext. This Transform algorithm is executed if and only if the search algorithm outputs "1" and the attributes embedded in the transformation key satisfy the access structure of the ciphertext CT.

Decrypt$(z, CT_{out}) \rightarrow M$. The data user runs the Decrypt algorithm with its blind value z and the partially decrypted ciphertext CT_{out} as input, and then the user can recover the message M with lightweight decryption.

III. Construction of LSABE Scheme

This section presents the entire construction of LSABE scheme as shown in Fig. 2.

A. System Initialization

Let g be a generator of bilinear group $\mathbb{G}$. Let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be the bilinear map, and $H : \{0,1\}^* \rightarrow G$, $H' : \{0,1\}^* \rightarrow Z_p^*$, $h : \{0,1\}^* \rightarrow \kappa$ are three hash functions and κ is the security parameter. KGC chooses the random element $\alpha, \beta, \lambda \in \mathbb{Z}_p^*, f \in \mathbb{G}$. The public parameter and master secret key of the system are $PP = (f, g, g^\beta, g^\lambda, e(g,g)^\alpha)$, $MSK = (\alpha, \beta, \lambda)$, respectively.

B. Key Generation

Key generation mainly consists of secret key generation and transformation key generation.

1) Secret key generation (SecrekeyGen)*:* KGC selects random elements $t, \delta \in \mathbb{Z}_p^*$ and calculates $K_1 = g^{\frac{\alpha}{\lambda+\delta}}$, $K_2 = \delta$, $K_3 = g^t$, $K_x = H(x)^t (\forall x \in S)$, $K_5 = g^\alpha g^{\beta t}$. The secret keys of the data user are $SK = (K_1, K_2, K_3, \{K_x\}_{x \in S}, K_5)$.

2) Transformation key generation (TransKeyGen)*:* The data user chooses a random value $z \in \mathbb{Z}_p^*$, and computes $K_3' = g^{zt}$, $K_x' = H(x)^{zt} (\forall x \in S)$, $K_5' = g^{z\alpha} g^{z\beta t}$. The transformation keys are $TK = (K_3', \{K_x'\}_{x \in S}, K_5')$.

C. Encryption

Encryption consists of file encryption and index generation.

1) File Encryption: The data owner randomly selects Υ from $\mathbb{G}_T^*$, and encrypts file M with his/her secret key $k_{SE} = h(\Upsilon)$ as follows: $C_M = \mathsf{SEnc}'_{k_{SE}}(M)$.

2) Index Generation: Data owner selects keywords from the message M to form a keyword set KW , where $KW = \{kw_1, kw_2 ..., kw_{l_1}\}$. Let A be an $n \times l$ matrix and ρ be the function that associates rows of A with the attributes. The access policy is denoted by (A, ρ). The secure index generation algorithm works as the following procedure.

(1) Choose a random $s \in \mathbb{Z}_p$ and a random vector $v \in Z_p^n$ with s as its first entry. For each $i \in [l]$, we let λ_i denote $A_i \cdot v$, where A_i is row i of A.

(2) Construct an l_1 degree polynomial $r(x) = \eta_{l_1} x^{l_1} + \eta_{l_1-1} x^{l_1-1} + ... + \eta_0$, such that $H'(kw_1), ..., H'(kw_{l_1})$ are the l_1 roots of the equation $r(x) = 1$.

(3) It gives the random numbers $\varrho_1, b \in \mathbb{Z}_p^*$ and constructs the secure index $I = \Upsilon \cdot e(g,g)^{\alpha s}$, $I_1 = g^b$, $I_2 = g^{\lambda b}$, $I_3 = g^s$, $I_4 = g^{\varrho_1}$, $I_i = g^{\beta \lambda i} H(\rho(i))^{-\varrho_1}$, $\hat{I}_j = \varrho_1^{-1} \cdot \eta_j$, $E = e(g,f)^{\varrho_1} e(g,g)^{\alpha b \varrho_1}$.

(4) Outsource the ciphertext CT to the cloud $CT = (I, I_1, I_2, I_3, I_4, \{I_i\}_{i \in [l]}, \{\hat{I}_j\}_{j \in \{0,1,...,l_1\}}, E, C_M)$.

D. Trapdoor Generation

If the user wants to match a file containing a certain keyword set $KW' = \{kw_1, kw_2, ..., kw_{l_2}\}$, one can create a trapdoor T'_{KW} using his/her keyword set and the secret key. The data user randomly chooses $u, \varrho_2 \in \mathbb{Z}_p^*$ and computes $T_1 = K_1^u$, $T_2 = K_2$, $T_3 = u\varrho_2 l_2^{-1}$, $T_4 = e(g,f)^u$, $T_{5,j} = \varrho_2^{-1} \Sigma_{i=1}^{l_2} H'(kw_i)^j$. The keyword trapdoor $T_{KW'}$ is $T_{KW'} = (T_1, T_2, T_3, T_4, \{T_{5,j}\}_{j \in \{0,1,...,l_1\}})$.

E. Search and Transform

When the cloud server receives the transformation key and the trapdoor from a data user, it then conducts the following two operations: Search and Transform.

1) Search *works as follows:*

i. The cloud calculates and determines whether the equation $T_4 \cdot e(T_1, I_1^{T_2} I_2) = E^{T_3 \cdot (\Sigma_{i=1}^{l_1} \hat{I}_j T_{5,j})}$ is true.

ii. If the equation is true, the cloud outputs 1 meaning that $KW' \subset KW$. Otherwise, the cloud outputs 0.

2) Transform *works as follows:*

i. If Search outputs 0 or S is associated with TK that does not satisfy the access structure in CT, Transform will not be executed.

ii. If Search outputs 1 and S is associated with TK satisfying the access structure in CT, Transform will be executed. Let $N \subset [l]$ be $N = \{i : \rho(i) \in S\}$ and a set of constants $\{w_i \in Z_p\}_{i \in N}$ with $\Sigma_{i \in N} w_i A_i = (1, 0, ..., 0)$. The cloud computes the transformed ciphertext as $I' = e(K_5', I_3) / e(\Pi_{i \in N} I_i^{w_i}, K_3') \cdot e(\Pi_{i \in N} K'^{w_i}_{\rho(i)}, I_4) = e(g,g)^{s\alpha z} e(g,g)^{s\beta t z} / (\Pi_{i \in N} e(g,g)^{\beta t z \lambda_i w_i}) = e(g,g)^{s\alpha z}$.

The transform algorithm outputs the transformed ciphertext $CT_{out} = (I, C_M, I')$, and sends it to the data user.

F. Decryption

In this algorithm, the data user takes the blind value z and the transformed ciphertext CT_{out} as input, and computes the ciphertext as $I/(I')^{1/z} = \Upsilon$, $k_{SE} = h(\Upsilon)$, $M = \mathsf{SDec}'_{\mathsf{k_{SE}}}(\mathsf{C_M})$. As mentioned before, the cloud server has partially decrypted the ciphertext at first, and then the user only needs to perform simple and fundamental operations to recover the message.

IV. Extension to LSABE-MA

A. System Model Of LSABE-MA

LSABE-MA scheme consists of the following entities.

1) Data Owner: IoT has been used in a diversity of industrial sectors, e.g., smart grids and smart manufacturing. Intelligent devices collecting ambient data in these systems communicate with each other. For example, a smart grid equipped with various sensors and smart meters can monitor the real-time status of microgrids [23]. In the healthcare monitoring scenario, using embedded sensors, the physical status parameters of the patient such as blood pressure, heart rate and body temperature, can be monitored in real time [24]. Some privacy-sensitive data are saved in various files. In order to support keyword searching, keywords are extracted from each file. Both keywords and files have been uploaded to the remote cloud after encryption at IoT nodes, as shown in Fig. 1.

2) Certificate authority (CA): CA is only responsible for generating the initial public parameters and allocate the global identity (GID) to the authorized users.

3) Attribute Authority (AA): There are a number of attribute authorities (AAs) in the IoT-cloud system. Every AA generates the attribute secret key (ASK) and the attribute public key (APK) for each attribute that it manages. Both APK and ASK will be consequently offered to the data owner.

4) Cloud server (CS): Cloud server that has extensive computing capability and the massive storage space can provide users with a number of computing and storage services. Moreover, the cloud server provides users with the keyword searching service. Users' privacy-sensitive data stored at the cloud server may be misused or disclosed.

5) Auxiliary Cloud Server (ACS): ACS is responsible for helping a user to complete a part of the decryption operation by using the user's transformation key as input.

6) Data User: Each data user has the user identity allocated by the CA and gets the AS from multiple authorities according to his/her attributes. Data users can search the ciphertexts with the set of keywords. Then, a trapdoor is computed from the user-predefined keywords and uploaded to the cloud. If the keywords in the trapdoor match the keywords in the encrypted index, the data user then converts his/her secret key into a random transformation key. The auxiliary cloud server next transforms the ciphertext into partially decrypted ciphertext, which is sent back to the data user.

B. Framework of LSABE-MA

Global Setup $(\kappa) \rightarrow (PP, MSK)$. Given the security parameter κ, Global setup algorithm outputs public parameter PP for the system, global identity GID for the authorized users and the master secret key MSK for each authority.

Authority Setup $(PP) \rightarrow (APK_{i,j}, ASK_{i,j})$. Each authority A_j conducts the authority setup algorithm, which inputs public parameter PP and generates an attribute public key $APK_{i,j}$ and an attribute secret key $ASK_{i,j}$ for each attribute i that it manages.

SecretKeyGen $(MSK, i, PP, GID, ASK_{i,j}) \rightarrow SK_{i,GID}$. Given PP, GID, an attribute i belonging to a certain authority, and the attribute secret key $ASK_{i,j}$ for this authority. The SecretkeyGen algorithm generates a secret key $SK_{i,GID}$ for this attribute and sends it to the data user.

Encrypt $(M, (A, \rho), KW, PP, \{APK_{i,j}\}) \rightarrow CT$. Given file M, access policy (A, ρ), keyword set KW, PP and the set of attribute public keys $APK_{i,j}$ for relevant authorities, the Encrypt outputs the ciphertext CT, which contains the encrypted secure index I and the encrypted file C_M.

Trapdoor $(\{SK_{i,GID}\}, KW', PP) \rightarrow T_{KW'}$. Given the secret key set, query keyword set KW' and PP, data users run Trapdoor, which outputs the keyword trapdoor $T_{KW'}$.

TransKeyGen $(\{SK_{i,GID}\}, z) \rightarrow TK_{GID}$. Data user runs the TransKeyGen algorithm, which takes as input the secret key set and a blind value z, and outputs the transformation key TK_{GID}. Then, the data user submits $T_{KW'}$ to the cloud server.

Search $(CT, T_{KW'}) \rightarrow 0/1$. Cloud server performs the search algorithm with the trapdoor $T_{KW'}$ and the ciphertext CT as input. If the output is "0", the query task failed. If the output is "1" , the query is successful and the cloud server continues to run transform algorithm.

Transform $(CT, TK_{GID}) \rightarrow CT_{out}/\ \perp$. Given the transformation key TK_{GID}, the cloud server can transform the ciphertext into a transformed ciphertext and then returns the transformed ciphertext CT_{out} to the user end. Otherwise, it outputs $\perp$.

Decrypt $(z, CT_{out}) \rightarrow M$. The data user runs the Decrypt algorithm with transformed ciphertext CT_{out} and its blind value z as input. The user can easily recover the message M through simple calculations.

V. Construction of LSABE-MA

A. System Initialization

Let S_A denote the set of attribute authorities and $H : \{0,1\}^* \rightarrow G$ is the mapping from the global identity GID to element of $\mathbb{G}$. There are two initialization processes to be illustrated as follows.

1) Global Setup: The CA executes the algorithm GlobalSetup, which chooses the random element $\lambda \in Z_p^*, f \in G$. The master secret key MSK and public parameter PP of the system are $MSK = \lambda$ and $PP = (f, g, g^{\lambda})$, respectively.

2) Authority Setup: Each authority $A_j (j \in S_A)$ has a set of attributes L_j. The attributes are disjoint with each other, i.e., $(L_i \bigcap L_j = \varnothing$ for $i \neq j)$. For each $i \in L_j$, the authority A_j also chooses two random exponents $a_i, y_i \in \mathbb{Z}_N$ as its attribute secret key $ASK_{i,j} = \{a_i, y_i, \beta_i\}_{i \in L_j, j \in S_A}$, and the attribute public key as $APK_{i,j} = \{e(g,g)^{a_i}, g^{y_i}, g^{\beta_i}\}_{i \in L_j, j \in S_A}$.

TABLE II. Storage cost comparison

Schemes	Public parameter size	Ciphertext size	Secret key size	Trapdoor size
[25]	$(3\|U\|+1)\|G\|+\|G_T\|$	$(l+2)\|G\|$	$(2\|S\|+1)\|G\|+\|Z_p\|$	$(2\|S\|+2)\|G\|$
[7]	$4\|G\|+\|G_T\|+3\|Z_p\|$	$(l+3)\|G\|+2\|G_T\|+(l_1+1)\|Z_p\|$	$(\|S\|+3)\|G\|+3\|Z_p\|$	$(\|S\|+2)\|G\|+\|G_T\|+(l_1+3)\|Z_p\|$
[26]	$5\|G\|+\|G_T\|+5\|Z_p\|$	$(l+1)\|G_T\|+\|G\|$	$(2+2\|S\|)\|G\|$	$(2\|S\|+4)\|G\|$
[6]	$3\|G\|+\|G_T\|$	$(2l+1)\|G\|+\|G_T\|$	$(\|S\|+2)\|G\|$	$\perp$
[10]	$4\|G\|+3\|Z_p\|$	$2l\|G\|+\|G_T\|$	$\|S\|\|Z_p\|+2\|S\|\|G\|$	$(2\|S\|+3)\|G\|+\|Z_p\|$
[21]	$(2\|U\|+10)\|G\|+3\|G_T\|$	$(l+4)\|G\|+\|G_T\|$	$3\|S\|\|G\|$	$3\|S\|\|G\|$
[27]	$4\|G\|+4\|Z_p\|$	$(2l+3)\|G\|$	$(2\|S\|+1)\|G\|$	$\perp$
[28]	$\|Z_p\|+5\|G\|+\|G_T\|$	$(3l+1)\|Z_p\|+(3l+1)\|G\|$	$(2\|S\|+2)\|G\|$	$\perp$
[29]	$\|U\|+2\|G\|+\|G_T\|+2\|Z_p\|$	$(l_1+1)\|Z_p\|+2\|G_T\|+(l+3)\|G\|$	$(\|S\|+6)\|G\|+3\|Z_p\|$	$2\|G\|+\|Z_p\|$
LSABE	$4\|G\|+\|G_T\|$	$(l_1+1)\|Z_p\|+2\|G_T\|+(l+4)\|G\|+3\|Z_p\|$	$(\|S\|+3)\|G\|+\|Z_p\|$	$(l_1+2)\|Z_p\|+\|G_T\|+\|G\|$
LSABE-MA	$3\|G\|+\|Z_p\|$	$(2l+1)\|G_T\|+(l+4)\|G\|+(l_1+1)\|Z_p\|$	$3\|S\|\|G\|$	$(\|S\|+1)\|G\|+\|G_T\|+(l_1+1)\|Z_p\|$

B. Key Generation

1) SecretKeyGen: Data user U_{GID} receives a set of attribute $I[j,GID]$ from authority A_j, and corresponding secret key $SK_{i,GID}$. For each $i \in I[j,GID]$, authority A_j computes $K_{1,i} = g^{\frac{\alpha_i}{\lambda+\delta}}$, $K_{3,i} = H(GID)^{y_i}$, $K_{4,i} = g^{\alpha_i} H(GID)^{\beta_i}$. We then have $SK_{i,GID}$ as follows: $SK_{i,GID} = \{K_{1,i}, K_{3,i}, K_{4,i}\}_{i\in I[j,GID], j\in S_A}$.

2) TransKeyGen: User U_{GID} chooses random value $z \in Z_p^*$, with which we can get $K'_2 = H(GID)^z$, $K'_{3,i} = H(GID)^{zy_i}$, $K'_{4,i} = g^{z\alpha_i} H(GID)^{z\beta_i}$. Then we can obtain transformation key $TK_{GID} = (K'_2, \{K'_{3,i}, K'_{4,i}\}_{i\in I[j,GID], j\in A_{GID}})$, where A_{GID} denotes the set of authorities issuing the secret key to user U_{GID}.

C. Trapdoor Generation

If a user wants to query a file containing keywords specified by keyword set $KW' = \{kw_1, kw_2 ..., kw_{l_2}\}$, he/she generates a trapdoor T'_{KW} using his/her keyword set and the secret key. Data user randomly chooses $u, \varrho_2 \in Z_p^*$ and computes $T_{1,i} = K_{1,i}^u$, $T_2 = H(GID)$, $T_3 = u\varrho_2 l_2^{-1}$, $T_{4,x} = \varrho_2^{-1} \Sigma_{x=1}^{l_2} H'(kw_i)^x$, $T_5 = e(g,f)^u$. The keyword trapdoor $T_{KW'}$ is then given by $T_{KW'} = (\{T_{1,i}\}_{i\in I[j,GID], j\in A_{GID}}, T_2, T_3, T_5, \{T_{4,x}\}_{x\in\{0,1,...,l_1\}})$.

D. Encryption

1) File Encryption: Data owner encrypts file M with secret key $k_{SE} = h(\Upsilon)$ and selects a random element Υ from $\mathbb{G}_T^*$. The encrypted file is represented as $C_M = \mathsf{SEnc}'_{\mathsf{k_{SE}}}(\mathsf{M})$.

2) Index Encryption: Data owner selects keywords from message M to form a keyword set KW, where $KW = \{kw_1, kw_2, ..., kw_{l_1}\}$. Let A be an $n \times l$ matrix and ρ be the function that associated rows of A to attributes. The access policy is denoted by (A, ρ). The encryption works as follows.

1. It chooses a random $s \in Z_p$ and a random vector $v \in Z_p^n$ with s as its first entry. For each $i \in [l]$, we let λ_i denote $A_i \cdot v$, where A_i is row i of A.
2. We construct the l_1 degree polynomial $r(x) = \eta_{l_1} x^{l_1} + \eta_{l_1-1} x^{l_1-1} + ... + \eta_0$, such that $H'(kw_1), ..., H'(kw_{l_1})$ are the l_1 roots of the equation $r(x) = 1$.
3. We randomly choose numbers $\varrho_1, b \in Z_p^*$ and construct the secure index by calculating $I_i = \Upsilon \cdot e(g,g)^{s\alpha_{\rho(i)}}$, $I_0 = g^b$, $I_1 = g^{\lambda b}$, $I_2 = g^s$. $I_3 = g^{\varrho_1}$, $I_{4,i} = g^{\beta_{\rho(i)}\lambda_i} g^{-\varrho_1 y_{\rho(i)}}$ $I_{5,x} = \varrho_1^{-1} \cdot \eta_x$, $E_1 = e(g,f)^{\varrho_1}$, $E_{2,i} = e(g,g)^{\alpha_{\rho(i)} b \varrho_1}$.
4. We outsource the ciphertext CT to the cloud where CT is represented by $CT = (I_0, I_1, I_2, I_3, E_1, \{I_i, I_{4,i}, E_{2,i}\}_{i\in[l]}, \{I_{5,x}\}_{x\in\{0,1,...,l_1\}}, C_M)$.

E. Search and Partially Decryption

After receiving the transformation key and the trapdoor from a data user, the cloud executes the following two steps: Search and Transform.

1) Search: Suppose the attributes embedded in TK_{GID} satisfy the access strategy in CT. Let $N \subset [l]$ be denoted as $N = \{i : \rho(i) \in S\}$. There are a set of constants $\{w_i \in Z_p\}_{i\in N}$ with $\Sigma_{i\in N} w_i A_i = (1, 0, ..., 0)$.

The cloud server then calculates the following equation to determine whether it is true.

$$T_5 \cdot e(\Pi_{i\in N} T_{1,\rho(i)}, I_0^{T_2} I_1) = (E_1 \cdot \Pi_{i\in N} E_{2,i})^{T_3 \cdot (\Sigma_{x=1}^{l_1} I_{5,x} T_{4,x})}.$$

If the equation is true, the cloud outputs 1 meaning that $KW' \subset KW$. Otherwise, the cloud outputs 0.

2) Transform: If Search algorithm outputs 0 or attributes associated with TK_{GID} does not satisfy the access strategy of CT, Transform algorithm outputs $\perp$.

If the output of Search algorithm is 1 and attributes in TK_{GID} satisfy the access strategy of CT, the cloud server will have the following expressions: $I' = e(\Pi_{i\in N} K'_{4,\rho(i)}, I_2) / e(I_3, \Pi_{i\in N} K'^{w_i}_{3,\rho(i)}) \cdot e(\Pi_{i\in N} I_{4,i}^{w_i}, K'_2) = \frac{e(g,g)^{sz\Sigma_{i\in N}\alpha_{\rho(i)}} e(g,H(GID))^{sz\Sigma_{i\in N}\beta_{\rho(i)}}}{(\Pi_{i\in N} e(g,H(GID))^{\beta_{\rho(i)} z\lambda_i w_i})} = e(g,g)^{sz\Sigma_{i\in N}\alpha_{\rho(i)}}$, and $I'' = \Pi_{i\in N} I_i = \Pi_{i\in N}(\Upsilon \cdot e(g,g)^{s\alpha_{\rho(i)}}) = \Upsilon^{|N|} e(g,g)^{s\Sigma_{i\in N}\alpha_{\rho(i)}}$.

Transformthen outputs $CT_{out} = (C_M, I', I'')$, which is the transformed ciphertext sent to the data user.

F. Decryption

In this algorithm, the data user inputs a blind value z and a transformed ciphertext CT_{out}, which is computed as follows: $(I''/(I')^{1/z})^{1/|N|} = (\Upsilon^{|N|})^{1/|N|} = \Upsilon$, $k_{SE} = h(\Upsilon)$, $M = \mathsf{SDec}'_{\mathsf{k_{SE}}}(\mathsf{C_M})$. We observe that the cloud server has partially decrypted the ciphertext at first, and then the user only needs to perform simple operations to recover the plaintext.

VI. Theoretical Analysis

A. Storage cost

Industrial systems consist of IoT devices, which are often limited in storage capacity. Thus, the storage cost of ABE schemes affects the realistic deployment in practical industrial IoT. We compare our proposed LSABE and LSABE-MA schemes with other most representative schemes such as [6, 10, 21, 25, 27–29]. We let $|S|$ be the size of attribute set S, $|U|$ be the size of the universe attribute set U, l_1 be the size of the keyword set KW and l be the number of rows in the matrix of access structure. The terms of $|G|$, $|G_T|$ and $|Z_n|$ represent the element lengths of groups G, G_T and

TABLE III. Computation cost comparison

Schemes	Key Generation	Encryption	Trapdoor	Search	Decryption
[25]	$(2\|S\|+2)E$	$(2l+2)E$	$(2\|S\|+2)E$	$(\|S\|+1)P+E_T$	$\perp$
[7]	$(\|S\|+3)E+E_T$	$(3+2l)E+3E_T+3P$	$P+E_T+(\|S\|+3)E$	$3P+3E_T+(2l+2)E$	E_T
[26]	$(2\|S\|+1)E$	$(l+1)E+E_T$	$(2\|S\|+1)E$	$(l+1)P+E_T$	$\perp$
[6]	$(\|S\|+2)E$	$P+(3l+1)E+E_T$	$\perp$	$\perp$	E_T
[10]	$(2\|S\|+2)E$	$(m+3+2l)E$	$(2\|S\|+3)E$	$(2\|S\|+4)P+E_T$	$\perp$
[30]	$(\|S\|+4)E$	$P+E_T+(5l+2)E$	$\perp$	$\perp$	$(3\|S\|+1)P+(\|S\|+1)E$
[21]	$4\|S\|E$	$2P+(l+6)E+E_T$	$8\|S\|E$	$2P+2lE$	$4P+E_T+(3\|S\|+5)E$
[27]	$(2\|S\|+2)E$	$(2l+4)E$	$(2\|S\|+4)E$	$(2l+3)P+lE_T$	$\perp$
[28]	$(4\|S\|+3)E$	$(5l+1)E+E_T$	$\perp$	$\perp$	$(4\|S\|+1)P$
[29]	$(\|S\|+7)E+2E_T$	$(l_1+2)E_T+2P+(l_1+l+4)E$	$3E$	$(l+3)P+3E$	P
LSABE	$(\|S\|+3)E$	$3P+(2l+4)E+3E_T$	$E+P+l_1E_T$	$4P+(2l+2)E+2E_T$	E_T
LSABE-MA	$4\|S\|E$	$3E_T+(2l+1)P+(2l+4)E$	$(\|S\|+1)E+P$	$3P+3E_T+(2l+1)E$	$2E_T$

Z_n, respectively. Table II shows the storage cost comparison between LSABE-MA and other schemes.

One of our LSABE-MA scheme's strengths is the preservation of the constant size of public parameters. In particular, there are 3 elements in group $\mathbb{G}$ and one element in $|\mathbb{Z}_n|$. However, the public parameter size in existing studies such as [21, 25, 29] grows linearly with the number of universe attributes U. Moreover, schemes such as [6, 10, 27, 28] contain the larger size of public parameters than our LSABE-MA. In addition, our LSABE scheme also maintains a constant public parameter size and does not vary with the number of attributes. Thus, our schemes have a much smaller number of initial parameters than the other existing schemes. This feature may make our schemes be more suitable to industrial IoT. Although the ciphertext size of both LSABE-MA and LSABE is larger than that of most existing schemes, it does not significantly affect the user experience because ciphertext uploaded to the cloud does not require extra storage at IoT devices.

The size of secret key in LSABE-MA is $3|S|$ in group G_T, which is at the same level as [21]. If the number of user attributes $|s|$ is no more than 3, our secret key size is smaller than that in [29]. The schemes in [6, 27, 28] have smaller secret key size than LSABE-MA while they do not support the keyword searching. Moreover, the secret key size of our LSABE is smaller than all other schemes except [6]. Thus, the secret key size in our schemes is smaller than those of most fully-functional schemes. The trapdoor size of LSABE-MA is at the same level as [7] and is smaller than those of schemes [10, 21, 25, 26]. Similarly, our LSABE has a smaller trapdoor size than all other schemes except [29], while the scheme of [29] does not work for IoT.

B. Computation cost

In addition to storage limitation, IoT devices often have computing capability constraint. We then evaluate the computation cost of our schemes. Table III compares our schemes with other schemes in terms of the computational costs. The element lengths in group G, G_T and bilinear pairing are denoted by $|E|$, $|E_T|$ and P, respectively.

During the key generation phase, our LSABE-MA seems to have a higher computational overhead than other schemes. This is because our LSABE-MA works for the multi-authority, which needs several authorities working together to generate the user's key. However, most of the other schemes are single-authority, which may result in the disclosure of the private key. Furthermore, our LSABE scheme with single-authority has smaller computing costs than other single-authority schemes.

With respect to data encryption, the data encryption is done by resource-limited sensor nodes in LSABE-MA. In the real IoT environment, encrypted privacy-sensitive files need to uploaded to the cloud immediately. So, the latency is crucial in the encryption phase. Our LSABE-MA and LSABE schemes only require $2l+4$ exponentiations on group G, which are much smaller than other existing schemes in [6, 7, 10, 28, 30]. In addition, schemes such as [6, 28, 30] without keyword query function do not have trapdoors and search algorithms. Meanwhile, other searchable ABE schemes in [10, 25–27] do not have the decryption phase.

During the trapdoor phase, users need to encrypted the query keywords. Our LSABE-MA only $|S|+1$ requires exponentiations on group G and no bilinear operations; this requirement is much lower than schemes in [7, 10, 21, 25–27]. Although the scheme in [29] has the smaller secret key size than LSABE-MA, it only supports one keyword. Our trapdoor algorithm adopted in LSABE-MA causes some computational overhead due to multiple keyword queries.

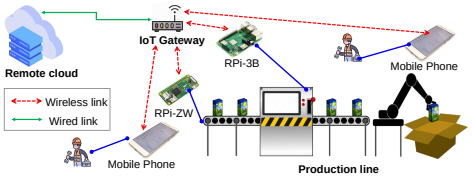
Among keyword-searching algorithms, our LSABE-MA requires $2l+1$ exponentiations on group G and three bilinear operations. This overhead is not significant compared with other schemes. In fact, the search is done by the cloud server so that this overhead can be negligible for cloud servers with strong computing capability.

Among decryption algorithms, our LSABE-MA requires only two exponentiations on group G_T and LSABE requires only one. Both LSABE-MA and LSABE require much fewers exponentiations than other schemes in [21, 28–30]. Although other lightweight schemes like [6, 7] have smaller exponentiations than our schemes, they are not suitable for IoT scenarios.

VII. Experimental results

A. Experimental settings and lightweight requirements

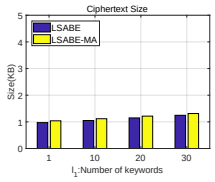
We conducted extensive experiments on a real industrial system, i.e., a sweet production line of an experimental workroom. Fig. 3(a) illustrates the experimental settings. We did not change the equipment of the exiting production line while we just installed a number of sensor devices around the production line to collect the ambient sound (e.g., noise) data. We adopted the off-the-shelf IoT devices such as mobile phones and Raspberry Pi models, which have been widely used in industrial prototypes or test-beds [23, 31–33]. Specifically, we considered three types of devices: i) Raspberry Pi model 3B (RPi-3B), ii) Raspberry Pi Zero W (RPi-ZW), iii) Mobile Phones. Fig 3(b) compares the key features of these devices. The reason for choosing the different devices is to evaluate the performance of our schemes at different IoT devices with diverse processing capabilities. In particular, Raspberry Pi models (i.e., RPi-3B and RPi-ZW) have much lower prices



(a) Industrial prototype to evaluate the proposed schemes

	RPi-ZW	RPi-3B	Smartphone
CPU	ARM11 1GHz Single core	ARM cortex-A53 1.2GHz 4-core CPU	Kirin960 4x2.36GHz 4x1.84GHz 8-core CPU
Memory	512MB RAM	1GB RAM	6GB RAM
Price ($)	10	35	400

(b) Test Platform Comparison



(c) Impacts of no. of keywords

Fig. 3. Experimental settings

than mobile phones though they have inferior computing capability to mobile phones. We leveraged the single-core RPi-ZW model because it can be used in less stringent industrial environment. Mobile phones can be flexibly leveraged in participatory sensing [31] and crowdsensing [34].

In experimental area (about 30 m^2), we placed one RPi-3B, one RPi-ZW and two mobile phones at different locations, as shown in Fig. 3(a). We mounted each Raspberry Pi model (i.e., RPi-3B and RPi-ZW) with a high sensitivity sound sensor (i.e., DAOKI 5PCS) to be used to collect ambient sound sensory data, while mobile phones used their built-in sensors to collect the sound sensory data. Java Pairing Basic Cryptography library (JPBC) [35] was installed in each device. Furthermore, we chose Type A pairings, which construct on the curve $y^2 = x^3 + x$ over the field F_q for some prime $q = 3 \mod 4$. Both G_1 and G_2 are the group of points EF_q, where $|Z_p| = 160$ bits, $|G| = |G_T| = 1024$ bits. The number of the keywords (denoted by l_1) is set to be 5 because extensive experimental results with varied number of keywords show the less influence of the number of keywords as shown in Fig. 3(c). Meanwhile, users may prefer some simple searching keywords (i.e., fewer keywords).

The practical industrial IoT systems have stringent requirements on both the execution time and storage cost of searchable encryption algorithms at IoT devices because the end-to-end delay heavily depends on the execution time while the storage cost poses resource concerns on the resource-limited IoT devices [32]. Regarding the execution time, the decryption time plays a more crucial role than the encryption time at IoT devices due to the asymmetric data transmission in industrial IoT [24]. Encrypted control messages (from control centers to IoT devices) often have a higher priority than encrypted sensory data (from IoT devices to clouds) while IoT devices require frequent decryption operations to decrypt control messages. Differently, encrypting sensory data has no such urgent requirement as decrypting control messages. Moreover, the storage consumption of public parameters plays a more important role than the ciphertexts to IoT devices since public parameters of searchable encryption algorithms require to be stored at IoT devices while ciphertexts can be stored at remote clouds.

B. Experimental Analysis

1) Execution time of LSABE-MA on different platforms: Fig. 4 presents the execution time of our LSABE-MA on different platforms. We define a threshold $T = 1$s that is the maximum latency for cryptograhic operations and schemes tolerated by real-time IIoT applications. The detailed experimental analysis is as follows.

Figs. 4(a) and 4(b) show encryption time and Trapdoor time of our LSABE-MA on different devices, respectively. The number of attributes varies from 1 to 20, being consistent with most of ABE schemes. The increased number of attributes leads to the increased execution time. We can also observe that mobile phones achieve the better performance than RPi-3B and RPi-ZW, whereas RPi-ZW shows the worst performance. For example, when the number of attributes $|S| = 5$, mobile phones, RPi-3B and RPi-ZW took 2.46s, 9.85s and 24.32s in encryption time, respectively. It may owe to the computing capability difference of diverse devices.

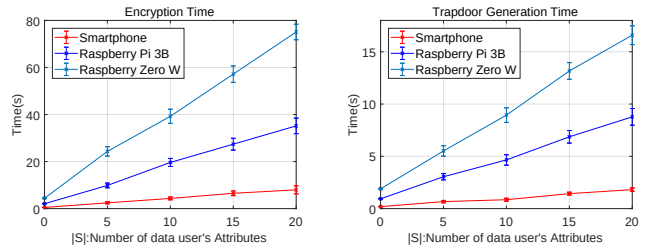
Fig. 4(c) shows that the decryption time does not significantly increase with the increased number of attributes. Even for RPi-ZW which has inferior computing capability to RPi-3B and mobile phones, it still can achieve millisecond-level decryption time (approximately 700 ms). It implies that our schemes are suitable for low-latency applications in IIoT.

Figs. 4(d), 4(e) 4(f) compare the full decryption and partial decryption time on mobile phones, RPi-3B and RPi-ZW. It took 77s in fully decrypting ciphertext with 20 attributes on RPi-ZW while it only takes 700ms by our LSABE-MA scheme. Meanwhile, it takes approximately 303ms and 46ms for decryption with 20 attributes ciphertext, on RPi-3B and mobile phones, respectively. In summary, our LSABE-MA can meet the low-latency requirement of IIoT applications.

We observe from the above experimental results that our LSABE-MA scheme has the excellent performance in decryption time on different devices (i.e., mobile phones, RPi-3B and RPi-ZW models) even though the encryption time is higher than decryption time. This feature is promising in real industrial environment because the it is more stringent to decrypt control messages/instructions than encrypting the collected data [23, 24, 33] while our LSABE-MA scheme can fulfill this requirement.

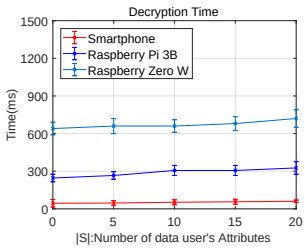
2) Comparison of LSABE-MA with other ABE scheme: Fig. 5 presents the performance comparison of our schemes with other baseline schemes (the representative schemes) including Liang's ABE scheme [21], Green's ABE scheme [6], Online/Offline ABE scheme [28], Miao's scheme [26], VABKS [27] and VFKSM [29].

Fig. 5(a) shows the comparison of our LSABE and LSABE-MA schemes with Liang's scheme and VFKSM in terms of public parameter size in system initialization. Apparently, our schemes keep the constant public parameter size. Particularly, the public parameter size of LSABE is 5,120 bits while that of LSABE-MA is 3,230 bits. However, the public parameter size in Liang's scheme and VFKSM increases as the growing number of all the attributes. If the number of global attributes is over 50, the public parameter size will be dozens of times

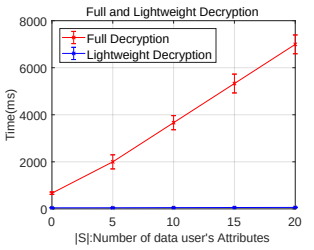


(a) Encryption Time on Different Devices

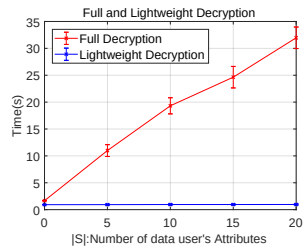
(b) Trapdoor Time on Different Devices



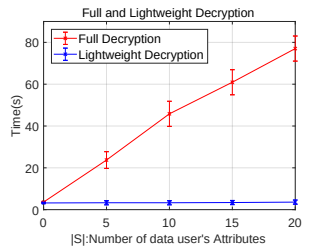
(c) Decryption Time on Different Devices



(d) Decryption on Mobile phones

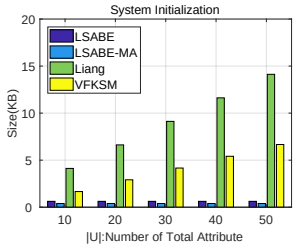


(e) Decryption on RPi-3B

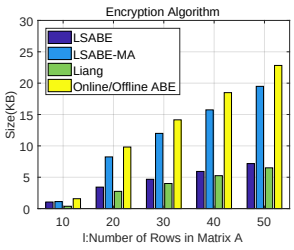


(f) Decryption on RPi-ZW

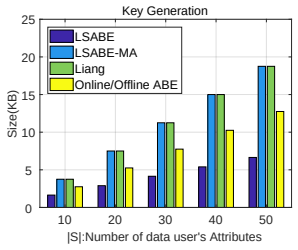
Fig. 4. Execution time for LSABE-MA scheme



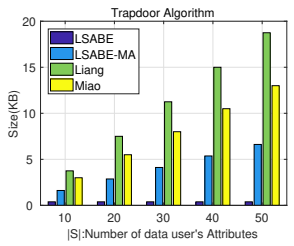
(a) Public Parameter Size



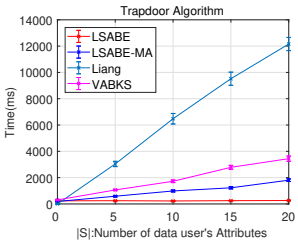
(b) Ciphertext Size



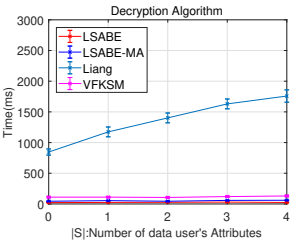
(c) Secret Key Size



(d) Trapdoor Size



(e) Trapdoor Generation Time



(f) Decryption Time

Fig. 5. Practical performance analysis in various ABE works

as large as LSABE-MA.

Fig. 5(b) compares our LSABE and LSABE-MA schemes with Liang's ABE scheme and Online/Offline ABE scheme. It is shown in Fig. 5(b) that both LSABE and LSABE-MA schemes have a smaller ciphertext size than Online/Offline ABE. Although the Liang's scheme has a smaller ciphertext size than our schemes, its ciphertext does not contain the various public key from different authorities.

Fig. 5(c) gives the comparison of our LSABE and LSABE-MA schemes with Liang's scheme and Online/Offline ABE scheme. As shown in Fig. 5(c), the secret key size of LSABE-MA is at the same level as Liang's ABE scheme. However, the secret key of LSABE-MA is generated by multiple authorities, which are not considered in Liang's scheme. Thus, the performance of LSABE-MA is better than Liang's scheme in terms of algorithm complexity.

Fig. 5(d) compares LSABE and LSABE-MA with Liang's scheme and Miao's scheme in terms of trapdoor size. It is shown in Fig. 5(d) that both LSABE and LSABE-MA schemes require much less storage and smaller transfer costs for the trapdoor than Liang's scheme and Miao's scheme. For example, when $|S| = 50$, the trapdoor size in Liang's scheme is 18.75 kbits nearly three times of our LSABE-MA scheme.

Fig. 5(e) compares our schemes with Liang's scheme and VABKS in trapdoor generation time. We observe that LSABE only has the constant time 237ms to complete a trapdoor generation, and the trapdoor size in LSABE-MA is also much smaller than other two schemes. When $|S| = 20$, LSABE-MA only requires 1,707ms, much smaller than Liang's scheme and VABKS, which need 12,160ms and 3,344ms, respectively.

Fig. 5(f) compares LSABE and LSABE-MA schemes with Liang's scheme and VFKSM in decryption time. We find that LSABE and LSABE-MA schemes just take 22ms and 44ms, respectively to decrypt the ciphertext while Liang's scheme and VFKSM take a much longer time than our schemes.

In summary, experimental results demonstrate that our proposed LSABE-MA and LSABE schemes have much better performance than other existing schemes in terms of parameter size, ciphertext size, secret key size, trapdoor size, trapdoor generation time and decryption time. Although the efficiency of LSABE-MA is slightly worse than LSABE (non-multi-authority version), it still outperforms other ABE schemes.

C. Extension of our schemes to smart city applications

In addition to industrial IoT applications, our proposed schemes can be extended to other smart city scenarios such as energy, transportation, healthcare. The proliferation of IoT data from every sector of cities brings not only opportunities but also challenges [36, 37]. On the one hand, data analytics on massive IoT data can be used to identify network performance bottlenecks or malicious behaviours so that corresponding countermeasures can be made to improve users' quality-of-service (QoS) or quality-of experience (QoE) [38, 39]. On the other hand, data analytics requires strong computing facilities

(e.g., cloud servers) to conduct computationally-complex tasks such as machine learning (ML) and deep learning (DL) algorithms. Thus, users' privacy-sensitive data has to be uploaded to remote cloud servers, which might be compromised to malicious attacks or intentionally (or unintentionally) disclose users' private data.

Our schemes can be adopted to address security and privacy challenges in smart cities. In particular, our proposed schemes can be integrated with ML/DL schemes to preserve data privacy while supporting data analytics. To this end, we may apply ML/DL models to conduct data analytics tasks such as traffic classification as implied by recent work [40]. Compared with conventional ML approaches, DL models require less feature-engineering efforts and obtain better performance. Therefore, DL approaches are more promising to the future data-driven applications. It is also shown in [40] that DL approaches can complete the traffic classification task on the weakly-encrypted traffic data. Therefore, our schemes may be more suitable for this application scenario. In the future, we will further investigate the integration of our schems with DL approaches to other smart city applications.

VIII. Conclusion

In this paper, we propose LSABE scheme, which can support multi-keyword search, fine-grained access control and lightweight decryption. We also define the security notions of LSABE and prove that our schemes can secure against the chosen-keyword attack and the chosen-plaintext attack. Furthermore, we propose an improved version of LSABE, namely LSABE-MA to support the multi-authority scenario. LSABE-MA is more applicable to industrial IoT environment. The functional analysis shows that both LSABE and LSABE-MA schemes require less storage and computing cost than most of the existing schemes. Experimental results on a real industrial system also demonstrate that our LSABE and LSABE-MA schemes outperform other representative schemes. The feasibility analysis in IoT devices proves that our LSABE and LSABE-MA schemes can be well adapted in industrial IoT environment.

References

[1] M. Usman, A. Jolfaei, and M. A. Jan, "RaSEC: An Intelligent Framework for Reliable and Secure Multi-Level Edge Computing in Industrial Environments," *IEEE Trans. on Industry Applications*, pp. 1–1, 2020.

[2] F. Farivar, M. S. Haghighi, A. Jolfaei, and M. Alazab, "Artificial Intelligence for Detection, Estimation, and Compensation of Malicious Attacks in Nonlinear Cyber-Physical Systems and Industrial IoT," *IEEE Trans. on Industrial Informatics*, vol. 16, no. 4, pp. 2716–2725, 2020.

[3] S. Ghane, A. Jolfaei, L. Kulik, K. Ramamohanarao, and D. Puthal, "Preserving privacy in the internet of connected vehicles," *IEEE Trans. on Intelligent Transportation Systems*, pp. 1–10, 2020.

[4] M. Usman, M. A. Jan, X. He, and J. Chen, "P2DCA: A Privacy-Preserving-Based Data Collection and Analysis Framework for IoMT Applications," *IEEE JSAC*, vol. 37, no. 6, pp. 1222–1230, June 2019.

[5] A. Jolfaei and K. Kant, "Privacy and security of connected vehicles in intelligent transportation system," in *2019 DSN*, June 2019, pp. 9–10.

[6] M. Green, S. Hohenberger, B. Waters *et al.*, "Outsourcing the decryption of ABE ciphertexts," in *USENIX Security*, vol. 2011, no. 3, 2011.

[7] Y. Yang, X. Liu, X. Zheng, C. Rong, and W. Guo, "Efficient Traceable Authorization Search System for Secure Cloud Storage," *IEEE Trans. on Cloud Computing*, pp. 1–1, 2018.

[8] S. Tan, K. Yeow, and S. O. Hwang, "Enhancement of a lightweight attribute-based encryption scheme for the internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6384–6395, 2019.

[9] J. Sun, H. Xiong, X. Liu, Y. Zhang, X. Nie, and R. H. Deng, "Lightweight and Privacy-Aware Fine-Grained Access Control for IoT-oriented Smart Health," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[10] Y. Miao, J. Ma, X. Liu, X. Li, Z. Liu, and H. Li, "Practical attribute-based multi-keyword search scheme in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3008–3018, Aug 2018.

[11] Y. Miao, X. Liu, R. H. Deng, H. Wu, H. Li, J. Li, and D. Wu, "Hybrid Keyword-Field Search with Efficient Key Management for Industrial Internet of Things," *IEEE Trans. on Industrial Informatics*, pp. 1–1, 2018.

[12] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, "Flexible and Fine-Grained Attribute-Based Data Storage in Cloud Computing," *IEEE Trans. on Services Computing*, vol. 10, no. 5, pp. 785–796, Sep. 2017.

[13] J. Wei, W. Liu, and X. Hu, "Secure and efficient attribute-based access control for multiauthority cloud storage," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1731–1742, 2018.

[14] K. Sowjanya, M. Dasgupta, S. Ray, and M. S. Obaidat, "An efficient elliptic curve cryptography-based without pairing kpabe for internet of things," *IEEE Systems Journal*, vol. 14, no. 2, pp. 2154–2163, 2020.

[15] J. Cui, H. Zhou, H. Zhong, and Y. Xu, "AKSER: Attribute-based keyword search with efficient revocation in cloud computing," *Information Sciences*, vol. 423, pp. 343–352, Jan. 2018.

[16] R. Amin, N. Kumar, G. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment," *Future Generation Computer Systems*, vol. 78, pp. 1005–1019, 2018.

[17] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07)*. Berkeley, CA: IEEE, May 2007, pp. 321–334.

[18] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the internet of things," *Future Generation Computer Systems*, vol. 49, pp. 104–112, 2015.

[19] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public Key Encryption with Keyword Search," in *Advances in Cryptology - EUROCRYPT*, 2004, pp. 506–522.

[20] J. Li and L. Zhang, "Attribute-based keyword search and data access control in cloud," in *2014 Tenth International Conference on Computational Intelligence and Security*. IEEE, 2014, pp. 382–386.

[21] K. Liang and W. Susilo, "Searchable Attribute-Based Mechanism With Efficient Data Sharing for Secure Cloud Storage," *IEEE Trans. on Information Forensics and Security*, vol. 10, no. 9, pp. 1981–1992, 2015.

[22] J. Sun, L. Ren, S. Wang, and X. Yao, "Multi-Keyword Searchable and Data Verifiable Attribute-Based Encryption Scheme for Cloud Storage," *IEEE Access*, vol. 7, pp. 66 655–66 667, 2019.

[23] Y. Wang, T. L. Nguyen, Y. Xu, Z. Li, Q. Tran, and R. Caire, "Cyber-Physical Design and Implementation of Distributed Event-Triggered Secondary Control in Islanded Microgrids," *IEEE Trans. on Industry Applications*, vol. 55, no. 6, pp. 5631–5642, 2019.

[24] P. Pace, G. Aloi, R. Gravina, G. Caliciuri, G. Fortino, and A. Liotta, "An Edge-Based Architecture to Support Efficient Applications for Healthcare Industry 4.0," *IEEE Trans. on Industrial Informatics*, vol. 15, no. 1, pp. 481–489, 2019.

[25] 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. on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 1187–1198, Apr. 2016.

[26] 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. on Services Computing*, 2017.

[27] Q. Zheng, S. Xu, and G. Ateniese, "VABKS: verifiable attribute-based keyword search over outsourced encrypted data," in *IEEE INFOCOM*, 2014, pp. 522–530.

[28] S. Hohenberger and B. Waters, "Online/Offline Attribute-Based Encryption," in *PKC 2014*, 2014, pp. 293–310.

[29] Y. Miao, R. Deng, K.-K. R. Choo, X. Liu, J. Ning, and H. Li, "Optimized Verifiable Fine-Grained Keyword Search in Dynamic Multi-owner Settings," *IEEE Trans. on Dependable and Secure Computing*, pp. 1–1, 2019.

[30] J. Ning, X. Dong, Z. Cao, L. Wei, and X. Lin, "White-Box Traceable Ciphertext-Policy Attribute-Based Encryption Supporting Flexible Attributes," *IEEE Trans. on Information Forensics and Security*, vol. 10, no. 6, pp. 1274–1288, Jun. 2015.

[31] Z. Wang, J. Wu, Y. Wu, S. Deng, and H. Huang, "Predictive location aware online admission and selection control in participatory sensing," *IEEE Trans. on Industrial Informatics*, vol. 15, no. 8, pp. 4494–4505, 2019.

[32] T. Hussain, K. Muhammad, J. D. Ser, S. W. Baik, and V. H. C. de Albuquerque, "Intelligent Embedded Vision for Summarization of Multiview Videos in IIoT," *IEEE Trans. on Industrial Informatics*, vol. 16, no. 4, pp. 2592–2602, 2020.

[33] R. A. G. Burbano, M. L. O. Gutierrez, J. A. Restrepo, and F. G. Guerrero, "IED Design for a Small-Scale Microgrid Using IEC 61850," *IEEE Trans. on Industry Applications*, vol. 55, no. 6, pp. 7113–7121, 2019.

[34] J. Xiong, R. Ma, L. Chen, Y. Tian, Q. Li, X. Liu, and Z. Yao, "A Personalized Privacy Protection Framework for Mobile Crowdsensing in IIoT," *IEEE Trans. on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.

[35] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proceedings of ISCC*. IEEE, 2011, pp. 850–855.

[36] M. H. Bhatti, J. Khan, M. U. G. Khan, R. Iqbal, M. Aloqaily, Y. Jararweh, and B. Gupta, "Soft Computing-Based EEG Classification by Optimal Feature Selection and Neural Networks," *IEEE Trans. on Industrial Informatics*, vol. 15, no. 10, pp. 5747–5754, 2019.

[37] I. A. Ridhawi, S. Otoum, M. Aloqaily, Y. Jararweh, and T. Baker, "Providing secure and reliable communication for next generation networks in smart cities," *Sustainable Cities and Society*, vol. 56, p. 102080, 2020.

[38] M. Aloqaily, I. A. Ridhawi, H. B. Salameh, and Y. Jararweh, "Data and service management in densely crowded environments: Challenges, opportunities, and recent developments," *IEEE Communications Magazine*, vol. 57, no. 4, pp. 81–87, 2019.

[39] S. Zafar, S. Jangsher, O. Bouachir, M. Aloqaily, and J. B. Othman, "QoS enhancement with deep learning-based interference prediction in mobile IoT," *Computer Communications*, vol. 148, pp. 86 – 97, 2019.

[40] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: An overview," *IEEE communications magazine*, vol. 57, no. 5, pp. 76–81, 2019.