Privacy-Preserving System for Enriched-Integrated Service

Kaisei KAJITA^{†a)}, Nonmember, Go OHTAKE[†], and Kazuto OGAWA^{†,††}, Members

SUMMARY In this study, we propose a secure data-providing system by using a verifiable attribute-based keyword search (VABKS), which also has the functions of privacy preservation and feedback to providers with IP anonymous server. We give both theoretic and experimental result, which show that our proposed system is a secure system with real-time property. One potential application of the system is to Integrated Broadcast-Broadband (IBB) services, which acquire information related to broadcast programs via broadband networks. One such service is a recommendation service that delivers recommendations matching user preferences (such as to TV programs) determined from the user's viewing history. We have developed a real-time system outsourcing data to the cloud and performing keyword searches on it by dividing the search process into two stages and performing heavy processing on the cloud side.

key words: privacy preserving, attribute-based keyword search, integrated broadcast-broadband service

1. Introduction

1.1 Background

The environment surrounding electrical media has changed significantly in recent years, and the Internet and mobile terminals have come into everyday use. Providing users (viewers) with valuable experiences through communication devices is an important issue concerning service providers. Many integrated services, such as Hulu [6], Netflix [9], and Amazon Prime Video [1], have been developed. For example, by referencing a user's *purchasing, sightseeing, and viewing history*, a service provider can provide recommendations of goods, places, and programs that have been personalized according to the individual user's preference.

Access to traditional broadcasting used to be mainly limited to inside buildings. In light of the current circumstances, the broadcast industry has developed the *Integrated Broadcast-Broadband (IBB)* system, and the International Telecommunication Union (ITU) has approved several recommendations related to IBB [8]. Many IBB systems have been developed on the basis of these recommendations, such as Hybridcast in Japan [7], Freeview Play in the UK [3], Hybrid broadcast broadband (Hbb) TV in Europe [4], and

Ginga in Brazil [5], which enable information related to a broadcast program to be transmitted via a broadband network.

The basic architecture of a mobile IBB system is shown in Fig. 1. As shown, a broadcast reaches a TV set through the broadcast channel and a broadband channel. A companion screen device (e.g., a smart phone or a tablet terminal) is connected to the TV set via a home LAN and can be taken out of the building. As a result, services on the IBB system are able to recommend and provide broadcast programs that match the user's preferences according to the viewing history of the broadcast programs he or she has viewed in the past. In addition, other services can be provided to the companion screen, and the user can obtain these services outside.

Most mobile IBB systems will likely use a cloud server, primarily because it reduces the space needed for hardware, and all of the data can be managed centrally. Here, a cloud server is considered an untrusted environment when its resources are publicly used and communication is performed over an untrusted network. A report from the Cloud Security Alliance [2] describes security problems that are threats to clouds; these threats along with their solutions have been summarized by J. Tang et al. [21]. When using a cloud server, one of the most crucial issues is how to protect personal information. One way to protect it is by encryption when a data user puts their personal information on the cloud. However, using a general encryption scheme, in which an encryption key and a decryption key are in a one-to-one relationship, is not an efficient choice for an IBB system. In the case of transmitting personal information to multiple service providers, it is necessary for users to use a different encryption key each time, as each service provider uses a different key. This becomes inefficient when the number of users becomes large.

As a solution to the above problem, we decided to fo-

Fig. 1 Basic architecture of a mobile IBB system.

Manuscript received August 21, 2020.

Manuscript revised December 10, 2020.

Manuscript publicized February 24, 2021.

[†]The authors are with the Science and Technology Research Laboratories, Japan Broadcasting Corporation, Tokyo, 157-8510 Japan.

^{††}The author was with the National Institute of Information and Communications Technology, Tokyo, 184-8795 Japan.

a) E-mail: kajita.k-bu@nhk.or.jp

DOI: 10.1587/transinf.2020NTP0009

cus on a scheme called *verifiable attribute-based keywords search (VABKS)* [24], which associates secret keys or ciphertexts with a set of attributes and enables access to be controlled in accordance with the decryption condition (policy). Furthermore, since the VABKS scheme has functions to search for keywords while encrypting them and to verify the search results, it can fulfill the requirements of our IBB service model (described in Sect. 2). Although VABKS is highly functional, one disadvantage is that the search time is lengthy and it is not possible to construct a system with a real-time property. These mean that it is still hard to use VABKS to ensure privacy.

1.2 Related Work

There are many related proposals that can be applied to IBB services. Among them, we decided to focus on VABKS because of its high affinity with IBB services. VABKS can trace its roots to *attribute-based encryption (ABE)* [17], which does not need to generate a distinct encryption key for each entity, and *searchable encryption* [19], whose ciphertexts can be searched while they are being encrypted.

In the case of an ABE scheme, only entities having an attribute that satisfies an access-control policy can decrypt a ciphertext, where the decryption keys or ciphertexts are generated based on attributes (residential place, sex, age, membership type, industry type, reputation, etc.). Therefore, one of the countermeasures against illegal access is to use an ABE. The first ABE scheme was proposed by Sahai et al. [17] as an extension of identity-based encryption (IBE). By using ABE, data such as personal information is encrypted only once. That is, because only the attributes of the service provider that can decrypt the encrypted personal information are specified at the time of encryption, it is possible to efficiently distribute personal information to multiple service providers. ABE schemes are classified as either key-policy ABE (KP-ABE) [12] or ciphertext-policy ABE (CP-ABE) [10]. In the case of the KP-ABE scheme, a ciphertext is associated with a set of attributes, and a private key is associated with a policy. In the case of the CP-ABE scheme, a ciphertext is associated with a policy, a private key is associated with a set of attributes, and a ciphertext can only be decrypted by a user whose attributes satisfy the policy that is attached to the ciphertext. Ohtake, Ogawa, and Safavi-Naini proposed a privacy preserving system for IBB services using CP-ABE [15].

One of the solutions for dealing with security threats in a cloud is to use searchable encryption. As for the notion of searchable encryption, in the method proposed by Song, Wagner, and Perring [19], a client stores a set of encrypted files, as well as an encrypted list of keywords for each file, on an untrusted server, and later, in the search phase, the client can efficiently retrieve some of the encrypted files containing specific keywords while keeping the keywords and the files secret. Searchable encryption schemes can be classified as either searchable symmetric encryption (SSE) [19] or public-key encryption with keyword search

(PEKS) [11]. PEKS was integrated by Wang et al. [22] into CP-ABE to create a new cryptographic primitive called “ciphertext-policy attribute-based encryption with keyword search function” (KSF-CP-ABE) based on Waters’ CP-ABE [23]. However, this scheme is inefficient because it requires composite-order bilinear groups. *Attribute-based keyword search (ABKS)* [18], [20], [22], which is a generalized notion of KSF-CP-ABE, has the properties of both ABE and searchable encryption. An ABKS scheme with efficient user revocation was proposed by Sun et al. [20]. A novel cryptographic primitive *verifiable ABKS*, called VABKS, was proposed by Zheng et al. [24], who gave a generic construction that enables anyone to verify the result of a keyword search.

1.3 Contributions

In this work, we propose a privacy-preserving and efficient IBB system by applying VABKS. There are four main contributions: (i) a fully privacy-preserving IBB system with VABKS and IP anonymity, (ii) an efficient keyword-search system, and (iii) an interactive system with feedback to providers, and (iv) IP anonymity of service providers. Kajita et al. [13] have overcome inefficient problem of VABKS and had feedback to service providers. By giving feedback to service providers, all entities in our system have merits to join the service. As far as we know, only our preliminary version [13] simultaneously considers access-control to the DB and privacy preservation. However, our previous version [13] has been pointed out a risk leaking users’ access histories because a cloud server can know which services used by users from service provider’s IP address.

1.3.1 Secure IBB System

The assumed service uses a database (DB) on a public cloud server, and it controls the user’s access to the DB. To preserve the user’s privacy, entities other than the user cannot know any information about the user. We construct a secure IBB system by applying VABKS, which has many functions. We control user access to the DB. The users have attributes and they can access the DB only if their attributes satisfy an access-control policy. We also give our system functions to search for keywords, while keeping the viewing history and location data encrypted, and to verify the result of the keyword search. The user’s personal information is encrypted as search tokens. Queries from the user and search results sent to the user should not be disclosed to any entity. When a data provider outsources its data to a cloud server, our system protects the data by means of the symmetric encryption algorithm included in the VABKS scheme.

1.3.2 Efficient Keyword-Search System

Although VABKS is highly functional, its heavy computational and search loads mean that we cannot obtain a real-

time property just by applying it to a system without any ingenuity. Here, we decided to construct an efficient *two-stage* system that deals with the problems of the VABKS scheme so that it can have the real-time property. It is assumed that a user has two kinds of personal information: viewing history and location data. The viewing history is used for narrowing down the keywords and data files in the DB to make the user's unique DB, and the location data is used for queries to the user's unique DB. We implemented this two-stage system and compared it with a one-stage (normal) system with VABKS. In the one-stage system, the time required for the search increases linearly. This seems to be a fatal defect for IBB services, since virtually all of them require quick responses in real-time. By contrast, our two-stage keyword-search system narrows down the data relevant to the user in advance. In so doing, it takes only about one second for it to search about fifty keywords. This means the proposed system is more practical than the system that conducts keyword searches in one stage.

1.3.3 Feedback to Providers

We consider a valuable system not only for user but also for the service provider. By giving feedback, the provider can get information about users, such as which users used their services. Concretely, in a service model without feedback, a business operator only provides data and can not know who or how many users access it. If the business operator knew that, it could provide an individualized service or new services utilizing statistical analysis. This would also enable interactive communication between users and service providers.

1.3.4 IP Anonymity of Providers

We make providers' IP addresses be anonymized when the provider outsources its data to the cloud server. We assume there are multiple service providers. Unless providers' IP addresses are hidden, a cloud server can store each provider's outsourced data to identify names of all providers. In the narrow-down and query phase, the cloud server knows which provider's data is used by users although all data in the cloud server is encrypted. This contribution is the main difference from [13]. By adding this function, we achieve a secure system that can make use of personal information while preserving its privacy.

1.4 Organization

The rest of the paper is organized as follows. Section 2 describes the IBB service and its requirements for our privacy-preserving system. Section 3 describes the cryptographic tools used in the system and defines the entities and securities of the system. In Sect. 4, we propose a system that preserves privacy of users, has an efficient keyword-search function, and gives feedback to the provider. In Sect. 5, we

prove the security of the system and demonstrate the practicality and efficiency of the system. We conclude our discussion in Sect. 6.

2. Envisioned IBB Service

We suppose a case where an IBB service is provided in a mobile environment and a viewing history is stored in a mobile communication device. The viewing history of the broadcast program and location information of the user are used in the service.

2.1 Service Model

We suppose that users obtain services in a mobile environment; here, we extend the existing services into integrated ones for mobile environments (see Fig. 2). For example, when a user drives somewhere related to the viewed broadcast program, a service can provide him or her with on-the-spot information about that place. In the service, providers provide data to the cloud, and each user uses the data. The users send their viewing histories and location data to the cloud and in accordance with this information, the cloud chooses data to recommend to them from among all the data provided by the providers. As the provided data increases, better services can be offered.

Such personal navigation-guide services provide on-the-spot information. There are other cases in which the service provider recommends TV programs related to the driving route. As such, when the user finishes driving, the programs can be transmitted from his or her mobile terminal to a TV set back at home, and the user can watch it on TV later. Two systems—a TV set at home and a mobile terminal in the car—are combined in this service model. We consider this personal navigation guide service to be a form of IBB service.

2.2 Requirements

Since the proposed IBB service involves a user driving a car, latency will significantly affect a system that provides “on-the-spot” information. This means the system needs to

Fig. 2 Service model.

have a low-latency property: a real-time property. Since the provided data should be large from the viewpoint of the service quality, CPU cost for processing the data may be large. The system that can process the large amount of data with a low-latency is required. If each entity does not have its merits, the service does not stand. From this viewpoint, the providers wish to obtain the information of the user that used their data. So the system that gives feedback to service providers is required. The service utilizes the user's viewing history and location data. Such data comprises critical personal data and naturally must be protected. Even if certain encryption schemes are used, additional heavy load is unacceptable for the service.

Data held by the service providers should not be leaked to unauthorized entities and the recommended information should be sent only to the target user. As for the recommended information, it should not be modified. Moreover, we should achieve the system which has more strong security by keeping the provider's IP address secret from cloud server. We thus define the requirements of the service as follows.

- **System requirements**

- Light load: The user can access the service in real-time without much latency.
- Efficient key management: The key management costs of each entity should be light.
- Secret searchability: The user can search for keyword with all data encrypted.
- Feedback possibility: The service provider knows which users access the data.
- Anonymity of providers: The IP address of the data provider should be anonymized.

- **Security requirements**

- Privacy preservation: There are four considerations here.
 - a. The user's personal data should not be leaked to any entity.
 - b. The source data held by the service providers should not be leaked to any unauthorized entity.
 - c. The individualized data sent to a certain user should not be leaked to any entity other than that user.
 - d. No one except authorized users know the provider of each data.
- Verifiability: The individualized data sent to a certain user should be verifiable by that user as to whether the information is correct.
- Unforgeability: The cloud server should not be able to forge any valid data.

3. Preliminaries

In this section, we first describe the cryptographic schemes

and algorithms used in our system. We then introduce the necessary entities and security requirements for our system.

3.1 Cryptographic Tools

Let $a \leftarrow S$ denote selecting an element a from a set S uniformly at random.

3.1.1 Symmetric Encryption

Let SE be a symmetric encryption scheme such that $SE = (\text{KeyGen}_{SE}, \text{Enc}_{SE}, \text{Dec}_{SE})$, where KeyGen_{SE} is an algorithm to generate a symmetric key, Enc_{SE} is an algorithm to encrypt a message, and Dec_{SE} is an algorithm to decrypt a ciphertext.

3.1.2 Digital Signature

Let Sig be a digital signature scheme such that $\text{Sig} = (\text{KeyGen}_{\text{Sig}}, \text{Sign}_{\text{Sig}}, \text{Verify}_{\text{Sig}})$, where $\text{KeyGen}_{\text{Sig}}$ is an algorithm to generate a pair of public and private keys, Sign_{Sig} is an algorithm to generate a signature for a message, and $\text{Verify}_{\text{Sig}}$ is an algorithm to verify whether the message matches the signature.

3.1.3 Attribute-Based Encryption

CP-ABE is defined as follows. Let ABE be an attribute-based encryption scheme such that $ABE = (\text{Setup}_{ABE}, \text{KGen}_{ABE}, \text{Enc}_{ABE}, \text{Dec}_{ABE})$, where Setup_{ABE} is an algorithm to generate a public parameter pm_{ABE} and a master key mk_{ABE} , KGen_{ABE} is an algorithm to generate a secret key sk_{ABE} taking pm_{ABE} , mk_{ABE} , and attribute α as inputs, Enc_{ABE} is an algorithm to generate a ciphertext c taking an access-control policy $\mathcal{P}$ and data d as inputs, and Dec_{ABE} is an algorithm to decrypt d from c by using sk_{ABE} .

In this study, we decided to use CP-ABE, since the ciphertext should be decrypted on the basis of an access-control policy in the proposed system.

3.1.4 Verifiable Attribute-Based Keyword Search

A verifiable attribute-based keyword search scheme (VABKS) has properties of ABE and can control users' access to the ciphertexts. In addition, it can search ciphertexts without decrypting them and verify the correctness of search results.

Let the data collection (also referred to as "data sets") $D = (\text{KS}, \text{MP}, \text{FS})$ denote a set of keyword sets KS, indexes MP, and data files FS. $\text{KS} = \{\text{KS}_1, \dots, \text{KS}_n\}$ is a set of n keyword sets in which elements are encrypted with a same access-control policy. $\text{MP} = \{\text{MP}(w) | w \in \bigcup_{i=1}^n \text{KS}_i\}$ is the set of $\text{MP}(w)$ that consists of a set of identifiers for identifying data files associated with keyword $w \in \bigcup_{i=1}^n \text{KS}_i$. $\text{FS} = \{F_1, \dots, F_N\}$ is a set of N data files. VABKS consists of six algorithms ($\text{Setup}_{\text{VABKS}}, \text{KeyGen}_{\text{VABKS}}, \text{BuildIndex}$

$VABKS$, $TokenGen_{VABKS}$, $SearchIndex_{VABKS}$, $Vrfy_{VABKS}$) defined as follows.

- $(mk_{VABKS}, pm_{VABKS}) \leftarrow Setup_{VABKS}(1^l)$ takes a system parameter l as input and generates a public parameter pm_{VABKS} and a master key mk_{VABKS} .
- $sk_{VABKS} \leftarrow KGen_{VABKS}(mk_{VABKS}, \alpha)$ takes mk_{VABKS} and α as inputs and generates a secret key sk_{VABKS} .
- $(Au, Index, D_{cph}) \leftarrow BuildIndex_{VABKS}(D, \mathcal{P}, \mathcal{P}')$: Auxiliary information Au , $Index$ and data ciphertext D_{cph} are obtained by running this algorithm, where $D = (KS, MP, FS)$, $\mathcal{P}$ is a set of access-control policies to encrypt n keyword sets in KS , and $\mathcal{P}'$ is a set of access-control policies for encrypting N data files in FS .
- $tk_{VABKS} \leftarrow TokenGen_{VABKS}(sk_{VABKS}, w)$ issues a search token tk_{VABKS} with credential sk_{VABKS} and a keyword w .
- $(rslt, proof) \leftarrow SearchIndex_{VABKS}(Au, Index, D_{cph}, tk_{VABKS})$ searches $Index$ and outputs the search result $rslt$ and a proof $proof$.
- $bits \leftarrow Verify_{VABKS}(sk_{VABKS}, w, tk_{VABKS}, rslt, proof)$ verifies $(rslt, proof)$ with respect to the search token tk_{VABKS} .

Correctness of $VABKS$ requires that, given $(mk_{VABKS}, pm_{VABKS}) \leftarrow Setup(1^l)$, $sk_{VABKS} \leftarrow KGen_{VABKS}(mk_{VABKS}, \alpha)$, for any keyword-based data collection D and keyword w , $(Au, Index, D_{cph}) \leftarrow BuildIndex_{VABKS}(D, \mathcal{P}, \mathcal{P}')$, $tk_{VABKS} \leftarrow TokenGen_{VABKS}(sk_{VABKS}, w)$, and $(rslt, proof) \leftarrow SearchIndex_{VABKS}(Au, Index, D_{cph}, tk_{VABKS})$, $(rslt, proof)$ always yields $1 \leftarrow Verify_{VABKS}(sk_{VABKS}, w, tk_{VABKS}, rslt, proof)$.

3.2 Definitions for the System

Here, we define the necessary entities and security requirements for the system.

3.2.1 Entities

The service has five entities: a trusted authority (TA) that is a trusted third party such as a system administrator, a data provider (DP) such as a broadcasting station, a data user (DU) who is both a subscriber and a driver, a cloud server (CL) that stores all data, and an IP anonymous server (IS). We define the roles of these five entities as follows.

DU: DU obtains individualized data from DP. This data is based on DU's personal information. DU does not want his personal information leaked to anyone else. DU has his own attributes. If the attributes satisfy DP's access-control policy, DU can access the data provided by DP.

DP: DP has a large amount of data. A data collection consists of keyword sets, indexes, and data files. DP has an access-control policy and controls which users can access the data it provides. DP does not provide DU with the data directly; instead, it passes the data to DU

via CL. DP does not give the data to anyone except for the authorized DUs.

CL: CL makes a bridge between DU and DP. CL receives DU's personal information and DP's data. All data sent from DU and DP are encrypted.

IS: IS makes a bridge between DP and CL. IS changes the DP's IP address when it receives DP's data and sends it to CL.

TA: TA authorizes all DUs and DPs.

3.2.2 Security

We define the security and attack models of the system on the basis of the service requirements described in Sect. 2.2. The definitions are specified for the proposed system.

We need to consider the security requirements under the assumption of the worst case. Thus, we will consider CL is to be malicious; it might not follow the protocol and might illegally modify the data. Moreover, it might try to analyze the data stored on the cloud server in order to learn personal information about DU or obtain data files belonging to DP.

The DUs have several types of personal information that is used as search keywords, and they want to obtain data files possessed by DP according to their personal information. We will assume that the DUs follow the protocols but still try to access the data in CL. If the DUs do not follow the protocols, they cannot receive the service, but they might illegally obtain another user's data in CL. From the viewpoint of privacy preservation, the DU's personal information should not be leaked anywhere.

In the proposed system, DP plays the role of service provider. If DP is not an honest entity, the system will fail. If DP provides incorrect data files, incorrect information will be sent to the DUs, and the service will not work. Thus, DP should therefore follow an honest model.

We assume that IS does not collude with the other entities and follows the protocol. Moreover, if TA is not trustworthy, the service will not work. Thus, it should follow an honest model as well.

(1) Security definitions.

The following properties are defined as the security concerning the system.

Data secrecy: CL should not deduce any keyword from the encrypted data files or search tokens. Data secrecy requires that plaintexts of data files and keywords cannot be recovered from encrypted data.

Access anonymity: Access anonymity requires the DP's anonymity, and the DP's IP address must be securely anonymized. CL cannot link any search result to its DP.

Unlinkability of search tokens: CL cannot link one search token to another even if they are for identical keywords. Unlinkability of search tokens requires a non-deterministic search-token-generation function, and

queries must be properly represented and securely encrypted.

Data unforgeability: CL cannot forge any correctly encrypted data files.

Verifiability of search results: If CL returns an incorrect search result, the DU can detect the cheating behavior.

Collusion resistance: DU may try to collude with other DUs in order to obtain another DU's secret key or access a personal DB in CL. Collusion resistance requires that no DU be able to decrypt another DU's ciphertexts in an encrypted personal DB even if all DUs except for the authorized ones collude.

(2) Attack models.

In terms of level of privacy preservation, we consider three threat models depending on the information available to CL and DU.

Known ciphertext model CL can access the encrypted data and the submitted search tokens, which are encrypted as ciphertexts. It can also receive and record the search result. The semantic meanings of this threat scenario are captured by the non-adaptive known ciphertext attack model, in which CL attacks with intent to forge encrypted data, or to obtain information concerning DU's keywords or DP's data files.

Known IP packet model CL can access the IP packet sent through IS and search results sent to DU. It can also record all the IP packets sent through IS and search results sent to all DU. The semantic meaning of this threat scenario is that CL attacks with the intent of learning the relationships between DU's queries and DP.

Collusion attack model DU colludes with another DU. They collude with the aim of obtaining another user's data possessed by DP or another user's secret key.

The following section shows a concrete construction of our system. It meets the above security requirements against the above attack models.

4. Proposed IBB System

We propose a secure and efficient system for the above service. Figure 3 shows the system.

4.1 System Overview

The system utilizes the VABKS scheme. The key generation, encryption, decryption, and keyword search algorithms outlined below are those of VABKS, if there is no mention to the contrary.

TA generates public information including an encryption key. Then it generates decryption keys by using the DU's attributes, and distributes each one to a distinct DU. The data is encrypted by DP, who also determines its access policy. DP generates public and secret keys that are not

Fig. 3 Proposed system.

identical to those of VABKS and obtains its certificate from a certain trusted third party. The data, public key, and certification are encrypted and outsourced to CL through IS. IS changes DP's IP address to a different address to anonymize the DP's data. When DU accesses the data, he or she generates search tokens from his or her history and sends them to CL. CL searches for encrypted data that matches the tokens and uses it to construct DU's personal DB. If DU's attributes satisfy the policy, the correct search results are obtained. We should point out here that the tokens are encrypted and the DU's attributes are embedded in the tokens, and further that the personal DB is typically small. When DU travels somewhere and uses the service, he or she makes another token from his location data and sends it to CL. CL searches the matched data from the personal DB and returns it to DU. DU decrypts the data and uses it. DU then encrypts his or her ID and the data's ID by using the public key attached to the data and sends it to the DP that is the owner of the used data. In this way, DP can know which DU accessed the data.

4.2 Keyword Setting

Before we give a concrete construction, we describe the keyword setting. We utilize viewing histories and location data as search keywords at the narrowing-down phase and the query phase in the proposed system, respectively. DU makes the viewing histories in advance before the system starts and prepares the location data in real time throughout the system. One viewing history contains several keywords such as program name, performers, and broadcast time. On the other hand, DP prepares keywords which contains the all kinds of items, such as program name, performers, broadcast time, and location data. The keyword setting of DU and DP are proceeded independently and these keywords are pre-defined by the system. Limiting the total number of search keywords can hasten the keyword search of the system. Consequently, the keyword search can be performed automatically without choosing keywords by DU itself and does not bother DU.

4.3 Construction

The basic ideas underlying this construction are a two-stage

keyword search, anonymous IP address, and feedback to DP. The keyword search has a preliminary stage to reduce the size of the original data collection to one more suitable for DU's personal DB. The second stage is a real-time stage that searches the personalized data, hence enabling high-speed retrieval from a smaller DB on the basis of DU's keywords (e.g., personal information). DP's access-control policy $\mathcal{P}$ for the data and keyword search can be specified by DP when it encrypts data. Any DU possessing attribute α satisfying $\mathcal{P}$ can search the encrypted data.

The proposed system, called "SYSV", consists of Sig, SE, ABE, and VABKS. Note that the item numbers from i to viii as shown below correspond to that in Fig. 3. Here, let $\{l_1, l_2, l_3\}$ be the security parameters.

i. Set-up phase: Given security parameters l_1, l_2 , TA executes $(mk, pm) \leftarrow \text{Setup}_{\text{VABKS}}(l_1)$ and $(mk_{\text{ABE}}, pm_{\text{ABE}}) \leftarrow \text{Setup}_{\text{ABE}}(l_2)$. TA then sets $mk = (mk, mk_{\text{ABE}})$, which it keeps secret, and $pm = (pm, pm_{\text{ABE}})$, which it makes public.

ii. Key-generation phase: TA obtains α from DU; then, it generates $sk = (sk, sk_{\text{ABE}})$ such that $sk = \text{KeyGen}_{\text{VABKS}}(mk, \alpha)$ and $sk_{\text{ABE}} = \text{KeyGen}_{\text{ABE}}(mk_{\text{ABE}}, \alpha)$. TA returns sk to DU.

iii. Data-encryption phase: DP executes the following five algorithms. Since one DP decides both policies $\mathcal{P}$ and $\mathcal{P}'$, $\mathcal{P} = \mathcal{P}'$ is realistic and in this phase, we let $\mathcal{P} = \mathcal{P}'$.

- Generate a symmetric encryption key $sk_{\text{SE}} \leftarrow \text{KeyGen}_{\text{SE}}(l_3)$.
- Encrypt each data $d_i \in \text{FS} = \{d_1, \dots, d_N\}$ by $c_{d_i} \leftarrow \text{SEnc}(sk_{\text{SE}}, d_i)$ and generates an encrypted data set $\text{CD} = \{c_{d_1}, \dots, c_{d_N}\}$.
- Encrypt the symmetric key sk_{SE} by $c_{sk} \leftarrow \text{Enc}_{\text{ABE}}(sk_{\text{SE}}, \mathcal{P})$, where $\mathcal{P}$ is DP's policy.
- Set the index set MP, which represents the relation between CD and an encrypted keyword set CPH generated next.
- Build an encrypted index by $(\text{Au}, \text{Index}, \text{CPH}) \leftarrow \text{BuildIndex}_{\text{VABKS}}(\text{D}, \mathcal{P}, \mathcal{P})$ and encrypt each keyword $(w_{i,1}, \dots, w_{i,n})$ of the data d_i , which is in the keyword set KS. Then, generate encrypted keywords $\text{CPH} = \{\text{cph}_{i,j}\}_{i,j}$ and auxiliary information $\text{Au} = (\sigma, \text{BF}, \text{vk}_{\text{Sig}})$, where σ is a digital signature for each keyword set, BF is a Bloom filter and vk_{Sig} is a verification key for Sig.
- Send $c = (\text{CD}, c_{sk}, \text{CPH}), \text{MP}, \text{Au}$ to IS.

iv. IP address anonymizing phase: IS changes the IP address of the received IP packets and sends the packets to CL.

v. Narrowing-down phase: This is the preliminary stage. CL reduces the size of the encrypted DB as follows.

- After CL receives c , MP, and Au from DP through IS, it stores them in its DB. Note that all data in the DB, except MP and Au, are encrypted.
- DU takes keywords $\mathbf{v} = \{v_1, \dots, v_q\}$ and sk as inputs, generates $\text{tk}_{\mathbf{v}} \leftarrow \text{TokenGen}_{\text{VABKS}}(sk, \mathbf{v})$, and

sends $\text{tk}_{\mathbf{v}} = \{\text{tk}_{v_1}, \dots, \text{tk}_{v_q}\}$ to CL.

- CL executes the search algorithm of VABKS for $\text{cph}_{i,j}$ by using a search token $\text{tk}_{\mathbf{v}}$, and it outputs $(\text{rslt}', \text{proof}') \leftarrow \text{SearchIndex}_{\text{VABKS}}(\text{Au}, \text{cph}_{i,j}, \text{tk}_{\mathbf{v}})$, where rslt' denotes the result of the search, and proof' is a certification that includes σ .
- Let the encrypted data set $\text{CD}_{\text{DU}} = \{c_{d_1}, c_{d_2}, \dots\}$ be in the user's personal DB, which corresponds to $\text{cph}_{i,j}$ by using the index set MP. Let $\text{CPH}_{\text{DU}} = \{\text{cph}_{\text{DU},1}, \text{cph}_{\text{DU},2}, \dots\} \subseteq \text{CPH}$. Then, CL stores $c_{\text{DU}} = (\text{CD}_{\text{DU}}, \text{CPH}_{\text{DU}})$ in DU's personal DB.

vi. Query phase: This is the real-time stage. DU queries CL with DU's personal DB, which is much smaller than the original encrypted DB, as follows.

- DU takes a keyword g and sk as inputs and generates search tokens $\text{tk}_g \leftarrow \text{TokenGen}_{\text{VABKS}}(sk, g)$. DU then sends tk_g to CL.
- CL searches with DU's personal encrypted data c_{DU} by using tk_g and obtains $(\text{rslt}, \text{proof}) \leftarrow \text{SearchIndex}_{\text{VABKS}}(\text{Au}, \text{cph}_{\text{DU},i}, \text{tk}_g)$, where rslt denotes the result of the search, and proof denotes a certification.
- CL returns c_{sk} and c_{d_i} to DU corresponding to $\text{cph}_{\text{DU},i}$.

vii. Decryption phase: DU decrypts (c_{sk}, c_{d_i}) as follows.

- DU verifies whether the results of the search by CL have been forged; namely, $\text{bits} \leftarrow \text{Verify}_{\text{VABKS}}(sk, (\mathbf{v}, g), (\text{tk}_{\mathbf{v}}, \text{tk}_g), (\text{proof}, \text{proof}'))$.
- If Verify outputs 0, DU terminates the decryption phase; otherwise, DU decrypts $sk_{\text{SE}} \leftarrow \text{Dec}_{\text{ABE}}(sk_{\text{ABE}}, c_{sk})$ only when DU's attribute α satisfies DP's policy $\mathcal{P}$. DU then obtains $d_i \leftarrow \text{Dec}_{\text{SE}}(sk_{\text{SE}}, c_{d_i})$ according to $\mathbf{v}$ and g .

viii. Feedback phase: DP obtains DU's ID ID_u as follows.

- If DU uses d_i , DU encrypts ID_u , $c_{\text{ID}} \leftarrow \text{Enc}_{\text{PKE}}(\text{pk}_{\text{PKE}}, \text{ID}_u)$ and sends it to DP with the index i .
- DP obtains c_{ID} and knows who uses its data by executing $\text{ID}_u = \text{Dec}_{\text{PKE}}(\text{sk}_{\text{PKW}}, c_{\text{ID}})$.

As long as DU keeps accessing the service, it repeats the procedure from the query phase to decryption phase.

5. Evaluation

We evaluate the system in accordance with the requirements in Sect. 2.2. We prove the system security in Sect. 5.2.1 including privacy-preservation, verifiability, and unforgeability. We then compare the system with a trivial system using a conventional encryption scheme in terms of CPU load and security in Sect. 5.3. In addition, we evaluate the system regarding to practicality from the viewpoint of CPU cost in Sect. 5.4.

5.1 System Evaluation

Our proposed system SYSV has four features; light load, efficient key management, feedback to providers, and IP anonymity. Light load is achieved with two-stage keyword-search, the efficient key management is achieved with the use of VABKS, and the feedback to DP is achieved by making a user ID's return pass. Moreover, we stress novelty that provider's anonymity is achieved with IS from our preliminary system.

Light load: Our two-stage keyword-search mechanism can dramatically reduce the computational load. The algorithmic loads on VABKS are heavy, so if it is used straight-forwardly in the system, the IBB service will need a lot of time to search for recommendations, which is generally not acceptable to users. We therefore have proposed a two-stage keyword search system to provide a fast real-time service. The processes with heavy loads are run in the preliminary stage, and the processes with light loads are run in the second real-time stage. In the preliminary phase, the viewing history is used for reducing the size of the DB in CL and making a small, unique DB for DU. This is what enables real-time services to be provided. This two-stage search is not limited to the above application; if the search process is divided into two processes, the same sort of scheme can be used to secure other online applications. Moreover, it is possible to construct a multiple-stage search.

Key management: Different secret keys for each user's attributes are generated in the proposed system, and only one public key is used for encryption. For this reason, it is necessary to encrypt the data of a broadcast program only once using the public key, regardless of the number of users. In this way, the burden of key-storage management and key distribution can be reduced compared with that of a system using a one-to-one cryptographic technique such as TLS communication. The number of keys each entity must have is only one, a public key of VABKS, and each entity does not need to manage any other key.

Feedback to the provider: DP's public key is included in each data stored in CL. Each DU who used the data can inform its ID to the DP. Only the DP who handles data used by a variety of DUs can know exactly who the DUs are. This function is indispensable when DP charges DUs.

IP anonymity: When DP outsource to the CL via IS, IS bridge the data between CL and DP. Then IS hides the IP address of DP. Therefore, no information can be leaked about who owns the data.

5.2 Security Evaluation

The system makes it possible to search keywords, while keeping the viewing history and location data encrypted, and to securely provide more interesting and suitable information for DUs by preventing leakage of personal information despite using a cloud.

5.2.1 Security Proof

Here, we show that SYSV has the properties of data secrecy, unlinkability between the search result and DP, unlinkability of search tokens, data unforgeability, verifiability of search results, and collusion resistance. Consequently, the proposed system for IBB service has high security.

Theorem 1: *If VABKS is selectively secure against a chosen-keyword attack (CKA) in the generic bilinear group model and ABE and SE are secure against a chosen-plaintext attack (CPA), SYSV has data secrecy and unlinkability of search tokens in the known ciphertext model.*

CKA is a model of an attack against VABKS [24]. Security against CKA means that, without being given any matching information between the search tokens and the keywords, an adversary $\mathcal{A}_{VABKS}$ cannot infer any information about the plaintext keyword of a keyword ciphertext. Selective security means that $\mathcal{A}_{VABKS}$ must determine the plaintext input of VABKS it intends to attack before the system is bootstrapped.

CPA is a model of an attack against many different encryption schemes: e.g. ABE and SE. Security against CPA means that, without being given any matched ciphertext, an adversary $\mathcal{A}_{\text{encryption}}$ cannot infer any information about the plaintext of a ciphertext [14].

Proof. We show that if there exists a polynomial-time algorithm $\mathcal{A}$ that breaks SYSV's data secrecy and search token unlinkability with advantage ε , we can construct a polynomial-time algorithm $\mathcal{B}$ that breaks CPA security for either ABE or SE with advantage $\frac{\varepsilon}{N^2}$, or selective CKA security for VABKS with advantage $\frac{\varepsilon}{(NM)^2}$, where N is the number of data files to be encrypted and M is the maximum number of keywords in one data file; i.e., the number of keywords to be searched is bounded by NM .

We consider two cases: (i) the challenger proceeds with a conventional CPA security game [14] with $\mathcal{A}$, or (ii) it proceeds with a selective security against CKA game [24] with $\mathcal{A}$. In the challenge phase, suppose $\mathcal{A}$ present two data collections $D_0 = (KS_0, MP, FS_0)$ and $D_1 = (KS_1, MP, FS_1)$, where $KS_0 = \{KS_{(0,1)}, \dots, KS_{(0,M)}\}$, $FS_0 = \{FS_{(0,1)}, \dots, FS_{(0,N)}\}$, $KS_1 = \{KS_{(1,1)}, \dots, KS_{(1,M)}\}$, and $FS_1 = \{FS_{(1,1)}, \dots, FS_{(1,N)}\}$ and policy $\mathcal{P}$.

- (i) The challenger selects $\lambda \leftarrow \text{bits}$ and encrypts FS_λ with ABE and $\mathcal{P}$. Now let us consider the advantage of $\mathcal{A}$ correctly guessing λ . The advantage of distinguishing which message was encrypted by the hybrid encryption of ABE and SE is equal. Therefore, given two sets of data files, FS_0 and FS_1 , if the advantage of distinguishing which data collection was encrypted is ε , then the advantage of distinguishing which data file was encrypted is $\frac{\varepsilon}{N^2}$ when selecting one data file from FS_0 and one from FS_1 .
- (ii) The challenger selects $\lambda \leftarrow \text{bits}$ and encrypts KS with VABKS. Since ABE is CPA-secure, the probability of

$\mathcal{A}$ inferring λ is negligible. Then, let us consider the advantage of $\mathcal{A}$ correctly guessing λ from keyword ciphertexts. The advantage of distinguishing two keywords encrypted by VABKS is equal. Therefore, given two keyword sets KS_0 and KS_1 , if the advantage of distinguishing which keyword set was encrypted is ε , then the advantage of distinguishing which keyword was encrypted is bounded by $\frac{\varepsilon}{(NM)^2}$ when selecting one keyword from KS_0 and one from KS_1 .

Therefore, we can construct $\mathcal{B}$ whose advantage is $\frac{M^2+1}{(NM)^2} \varepsilon = \frac{\varepsilon}{N^2} + \frac{\varepsilon}{(NM)^2}$ in the known ciphertext model if there exists a polynomial-time algorithm $\mathcal{A}$ that breaks SYSV's data secrecy and search token unlinkability with advantage ε . $\square$

Theorem 2: SYSV has unlinkability between the search results and DP in the known IP-address attack model if Theorem 1 holds.

Security against a known IP-address attack means that, without being given any matching information between the actual IP address of DP and the data received from IS, CL cannot get any information related to DP whose data is used by DU.

Proof. When Thm. 1 holds, only IS knows which DP is the provider of each data stored in CL (except for the provider). It is assumed that IS is honest, so IS does not leak any information. IS does not collude with other entities, so the other entities do not know each data's provider. In addition, from Thm 1, the SYSV has data secrecy and CL cannot get any information from the data stored in CL. This means that CL cannot learn any information from DU's queries or DP. $\square$

Theorem 3: SYSV has data unforgeability and verifiability of search results if Sig is non-adaptively unforgeable against a known ciphertext attack.

This security means that SYSV is secure against an attack from CL aiming to intentionally or unintentionally modify stored data.

Proof. This theorem can be proved from the security definition of Sig directly. Given correct $(\text{rslt}, \text{rslt}', \text{proof}, \text{proof}')$ and $(\mathbf{v}, g, \text{tk}_v, \text{tk}_g)$, DU executes the verification algorithm and outputs 1 with overwhelming probability from the verifiability of secure Sig. Moreover, we assume CL attacks in the known ciphertext model. If Sig is non-adaptively unforgeable, CL cannot forge new encrypted data files. $\square$

Theorem 4: SYSV achieves collusion resistance in the collusion attack model if Theorem 1 holds.

This security means that, even if multiple DUs collude to get the other DU's personal information, they cannot get any information.

Proof. This theorem can be directly proved from the security definitions of ABE and VABKS. Attackers can recover data only if they have enough attributes to satisfy the policy the data sender defined. That is, at least one user among

the colluders has to have enough attributes. Even if multiple users collude, there is negligible probability of them being able to generate another user's secret key sk or obtain another user's personal DB c_{DU} since each secret key sk is randomized by the secure key generation algorithms $\text{KeyGen}_{\text{ABE}}$ and $\text{KeyGen}_{\text{VABKS}}$. $\square$

5.2.2 System Security

Regarding to system security, privacy preservation is achieved with the use of VABKS and IS, verifiability is achieved with the verifiability of Verify of VABKS, and unforgeability is achieved with unforgeability of VABKS.

- **Privacy-preservation:** The search tokens from DUs are encrypted and the data provided to them is also encrypted. In addition, when CL obtains data from DPs, the IP address is changed. This IP address change effectively anonymizes the address. It means that CL cannot know DP of the used data. This means that CL cannot learn any information related to DUs and DP.
- **Verifiability:** VABKS has an algorithm of $\text{Verify}_{\text{VABKS}}$ that can verify the correctness of search results. Each DU uses the algorithm straight-forwardly for the verification of search results from CL.
- **Unforgeability:** The system uses a EUF-CKA secure VABKS. Therefore, the system prevents CL's forgery of the stored data.

5.3 Comparison

We compared our system, SYSV, with our previous version KOO19 [13] and a trivial system using only a symmetric encryption scheme (as shown in Table 1). We found that the encryption and decryption load of the trivial system were very light and satisfies real-time property due to its use of the symmetric encryption scheme such as AES. Regarding the CPU load, the trivial system was superior to ours for the same reason. In the trivial system, DP has to manage all DUs' encryption keys and its cost increases in proportion to the number of DUs; in contrast, SYSV can efficiently manage keys by using its unique attributes and policies. SYSV can search for the keywords and data files without decryption. In the trivial system, CL cannot obtain any plaintexts (such as search keywords and data files) from the encrypted data. However, if all entities share the same secret keys, all data may be leaked, since CL is a malicious

Table 1 Comparison of systems.

	Trivial	KOO19 [13]	Proposed
Privacy preservation	✓	✓	✓
Light load	✓	✓	✓
Efficient key management	—	✓	✓
Secret searchability	—	✓	✓
Verifiability	—	✓	✓
Unforgeability	—	✓	✓
Feedback possibility	—	✓	✓
IP anonymity	—	—	✓

Table 2 Experimental results (sec). #Keywords denote the number of keywords in the DB of DP. Encryption time denotes the process time for the encryption algorithm, and the number of effective digits is three. TokenGen, SearchIndex, Verify, and Decrypt denote the TokenGen, SearchIndex, Verify, and Dec algorithms, respectively. Search denotes the sum of TokenGen, SearchIndex, Verify, and Decrypt.

#Keywords	Encryption time	Search time (total)			
		TokenGen	SearchIndex	Verify	Decrypt
1	0.657	0.778			
		0.0267	0.730	0.0116	0.0100
12	2.95	0.842			
		0.0271	0.784	0.0111	0.00978
54	11.7	1.18			
		0.0306	1.02	0.0456	0.0377
134	29.3	1.72			
		0.0283	1.46	0.100	0.0815
918	196	7.06			
		0.0282	5.84	0.663	0.532
1836	402	13.3			
		0.0294	10.9	1.32	1.06
2476	525	17.7			
		0.0278	14.5	1.75	1.40
3672	804	27.7			
		0.0291	21.2	3.61	2.87

Fig. 4 Relationship between the number of keywords and search time.

entity. In this sense, the trivial system is secure only when CL is not malicious. That is, it is secure even if CL is a malicious entity. As we mentioned earlier, SYSV and KOO19 satisfy the requirements of verifiability, unforgeability because both systems employ VABKS scheme. Moreover, SYSV and KOO19 have feedback possibility. The advantage of SYSV against other systems is especially that SYSV places IP anonymous server between CL and DP. SYSV can prevent CL from attacking system by using IP address of DP. This improves system security and satisfies IP anonymity.

5.4 Implementation

Next, we evaluated the proposed system from the viewpoint of performance. Concretely, we implemented SYSV and made sure of the real-time property. Specifically, we implemented a two-stage search to achieve the real-time property and evaluated its effects. Figure 4 shows the relationship between the number of keywords in the DP's DB and the search time; the details are shown in Table 2. All algorithms were implemented on a PC whose specifications are as follows: CPU: Intel Core i7-4790 (3.60GHz), memory: 8GB,

OS: Cent OS 7.2, and browser: Firefox 38.3.0. Almost all of the encryption algorithms were written in JavaScript (some of them were written in C/C++ due to the limitations of the crypto library). As shown in Fig. 4 and Table 2, the search time was proportional to the number of keywords. The maximum number of data files used in the experiments was 884, and the number of keywords was 3672. The results show that it took 27.7 seconds to search data files matched to one token. Actually, the number of data files provided by DPs is very large in the IBB system, and the number of tokens may be more than one. We assumed that the data files of the DPs consisted of program title data and time along with related keywords (title, genre, cast members, etc.). Therefore, in the case that DP is a broadcaster, the number of data files should increase in accordance with the number of broadcast programs. For example, Channel 1 in Japan has about 300 programs per week, and each program has its own keywords. If there are four keywords per program, The total number of keywords per year is about $62,400(\text{keywords/year}) = 300(\text{programs/week}) \times 52(\text{weeks/year}) \times 4(\text{keywords/program})$ for the Channel 1. Supposing the data files of n channels are collected onto CL, the number of keywords would be its n -fold ($\approx n \times 62,400$). From the experimental results, it takes roughly $n \times (62,400/3672) \times 27.7 \text{ sec} (\approx 8n \text{ min})$ to search data files matched to one token. Supposing that the number of tokens is m , the time increases another m -fold ($8nm \text{ min}$). Although large data files enable better recommendations, it takes too much time to search them. If replies to queries take a long time, the service will not be successful. In the preliminary stage, therefore, the data files are reduced. For instance, if CL can reduce the size to 25 data files including 100 keywords, DU can obtain the data files within about 1.5 seconds after it sends a token to CL. Such a service would be acceptable to users.

6. Conclusion

We proposed a secure data-providing system using VABKS. Our proposed system provides users with information related to TV programs while preserving the privacy of their personal information, such as their viewing history and location data, by encrypting it. In the proposed system, moreover, the IP anonymous server is installed to improve the privacy, which was a weak point in our preliminary system in [13]. An integrated service with access-control, privacy preservation, data secrecy, and feedback to the provider has not yet reached practical application, so this work is the first system. Consequently, we can construct a system that has both high security and efficiency.

Future works: A smart selection of keywords may enable DP to provide IBB service with a higher level of user satisfaction. Another future work is to consider the optimum size for the personal DB that can satisfy the user's preferences. In the security aspects, we have two future works. First, The reduction loss in theorem 1 is loose, since the size of N and M are possibly very large in the proposed system, so yet another future work is to minimize the security reduction loss, and thereby make a more secure system. Second, the known IP packet model in Sect. 3.2.2 (2)-2 does not cover the case that CL can trace DU's IP address. We consider that CL can attack only by DU's search results and DP's IP address, not by DU's IP address in known IP packet model. Since DU may use a telecommunications carrier in general, we assume that the cloud server and DU use different providers and IP address of DU is not passed to CL in our system. It is natural at this point. However, in the case the system uses IPv6, IP address of DU does not change even if DU and CL use different providers. Thus, an additional trusted proxy server is introduced and is placed between CL and DU. This is a trivial solution and we study more efficient ways for IPv6 system in future.

Acknowledgments

We thank the Security Fundamentals Laboratory of National Institute of Information and Communications Technology for understanding and supporting our research.

References

- [1] Amazon, <https://www.amazon.com/Prime-Video/b?ie=UTF8&node=2676882011>
- [2] Cloud Security Alliance, "The Notorious Nine: Cloud computing top threats in 2013," https://downloads.cloudsecurityalliance.org/initiatives/top_threats/The_Notorious_Nine_Cloud_Computing_Top_Threats_in_2013.pdf, 2013.
- [3] DigitalUK, "Freeview play - technical specification," 2016.
- [4] ETSI, "Technical specification 102 796 v1.4.1. European Telecommunications Standards Institute," 2016.
- [5] Ginga, "ABNT NBR 15606 series," <http://www.abntcolegiao.com.br/coltv.aspx?Q=CING45WV08&ID=361862>, 2016.
- [6] Hulu, <http://www.hulu.com/>
- [7] IPTVForum, <https://www.iptvforum.jp/en/hybridcast/>
- [8] ITU, "Recommendation ITU-R BT.2075-1," https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.2075-1-201701-S!!PDF-E.pdf, 2017.
- [9] Netflix, <http://www.netflix.com/>
- [10] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," *Proc. IEEE S&P '07*, IEEE, pp.321–334, 2007.
- [11] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," *Proc. International Conference on the Theory and Applications of Cryptographic Techniques*, pp.506–522, Springer-Verlag, 2004.
- [12] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," *Proc. ACM-CCS '06*, pp.89–98, ACM, 2006.
- [13] K. Kajita, K. Ogawa, and G. Ohtake, "Privacy preserving system for real-time enriched-integrated service with feedback to providers," *Proc. HCI-CPT '19*, pp.385–403, 2019.
- [14] J. Katz and Y. Lindell, *Introduction to Modern Cryptography: Principles and Protocols*, Chapman & Hall/CRC, 2007.
- [15] G. Ohtake, K. Ogawa, and R. Safavi-Naini, "Privacy preserving system for integrated broadcast-broadband services using attribute-based encryption," *IEEE Trans. Consum. Electron.*, vol.61, no.3, pp.328–335, IEEE, 2015.
- [16] G. Ohtake, R. Safavi-Naini, and L.F. Zhang, "Outsourcing of verifiable attribute-based keyword search," *Proc. Nordic Conference on Secure IT Systems '17*, pp.18–35, Springer-Verlag, 2017.
- [17] A. Sahai and B. Waters, "Fuzzy identity-based encryption," *Proc. Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp.457–473, Springer-Verlag, 2005.
- [18] J. Shi, J. Lai, Y. Li, R.H. Deng, and J. Weng, "Authorized keyword search on encrypted data," *Proc. European Symposium on Research in Computer Security '14*, pp.419–435, Springer-Verlag, 2014.
- [19] D.X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," *Proc. IEEE S&P '00*, pp.44–55, IEEE, 2000.
- [20] 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, IEEE, 2016.
- [21] J. Tang, Y. Cui, Q. Li, K. Ren, J. Liu, and R. Buyya, "Ensuring security and privacy preservation for cloud data services," *ACM Computing Surveys (CSUR)*, vol.49, no.1, pp.13:1–13:39, ACM, 2016.
- [22] C. Wang, W. Li, Y. Li, and X. Xu, "A ciphertext-policy attribute-based encryption scheme supporting keyword search function," *Proc. Cyberspace Safety and Security '13*, pp.377–386, Springer-Verlag, 2013.
- [23] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," *Proc. Public Key Cryptography '14*, pp.53–70, Springer-Verlag, 2011.
- [24] Q. Zheng, S. Xu, and G. Ateniese, "VABKS: Verifiable attribute-based keyword search over outsourced encrypted data," *Proc. IEEE Infocom '14*, pp.522–530, IEEE, 2014.

Kaisei Kajita received the B.S. degree from University of Electro Communication in 2015 and M.S. degree from Tokyo Institute of Technology in 2017. He joined NHK (Japan Broadcasting Corporation) in 2017. He is currently a research engineer of NHK Science & Technology Research Laboratories. His research interests include cryptography, information security, Integrated Broadcast-Broadband (IBB) system.

Go Ohtake received the B.E. and M.E. degrees from Tokyo Institute of Technology, Tokyo, Japan in 1999 and 2001, respectively. He joined NHK (Japan Broadcasting Corporation) in 2001 and received Ph.D. degree from Institute of Information Security in 2009. He was a visiting researcher at University of Calgary from December 2014 to November 2015. He is currently a senior manager of NHK Science and Technology Research Laboratories. His research interests include public key cryptography and its ap-

plication for copyright protection and privacy preserving.

Kazuto Ogawa received the B.E. and Ph.D. degrees from the University of Tokyo in 1987 and 2008, respectively. He joined NHK (Japan Broadcasting Corporation) in 1987. Since 2020, he has been a senior researcher of National Institute of Information and Communications Technology. He has mainly engaged in research and development on video image processing systems and digital content rights management systems.