Privacy Preserving System for Real-Time Enriched-Integrated Service with Feedback to Providers

Kaisei Kajita, Kazuto Ogawa(✉), and Go Ohtake(✉)

Japan Broadcasting Corporation, Tokyo, Japan
{ogawa.k-cm,ohtake.g-fw}@nhk.or.jp

Abstract. We have developed a secure data-providing system for an enriched-integrated service with feedback to providers featuring a verifiable attribute-based keyword search (VABKS). One potential application of the system is the Integrated Broadcast-Broadband (IBB) service, which acquires information related to broadcast programs via broadband networks. One of the services IBB provides is a recommendation service that delivers recommended information matching user preferences (such as TV programs) based on user viewing history. Another application is in mobile environments featuring smart-phone usage, where services based on user location can be suggested. In this study, we propose a secure system that adds the functions of privacy preservation and feedback to providers. Thereby the functions provide increased business benefit to users of the IBB service for mobile usage, and feedback property provides another benefit to the providers of IBB services.

Keywords: Privacy preserving · Secure systems · Attribute-based keyword search · Attribute-based encryption · 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 now come into everyday use. Providing users (viewers) with valuable experiences through communication devices is an important issue concerning service providers. Many systems for integrated services, such as Hulu, Netflix, and Amazon, have been developed. For example, by referencing a user's *purchasing, sightseeing, and viewing history*, a service provider can provide personally recommended goods, places, and programs according to that user's preference.

Access to traditional broadcasting was mainly limited to inside buildings. It was not possible to provide personalized services because there was no Internet environment. In light of the current circumstances, the broadcast industry has

A. Moallem (Ed.): HCII 2019, LNCS 11594, pp. 385–403, 2019.
https://doi.org/10.1007/978-3-030-22351-9_26

developed the *Integrated Broadcast-Broadband (IBB)* system, and the International Telecommunication Union (ITU) has approved several recommendations related to IBB [1]. Many IBB systems have been developed on the basis of these recommendations, such as Hybridcast in Japan, Freeview Play in the UK, Hybrid broadcast broadband (Hbb) TV in Europe, and Ginga in Brazil, which enable information related to a broadcast program to be transmitted via a broadband network. It is expected that new user experiences will be provided in IBB services.

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. The companion screen device (e.g., a smart phone or a tablet) is connected to the TV set via a home LAN network and can be taken out of the building. As a result, IBB services are able to provide recommended broadcast programs that match the user's preference according to the viewing history of the broadcast program he or she has viewed in the past.

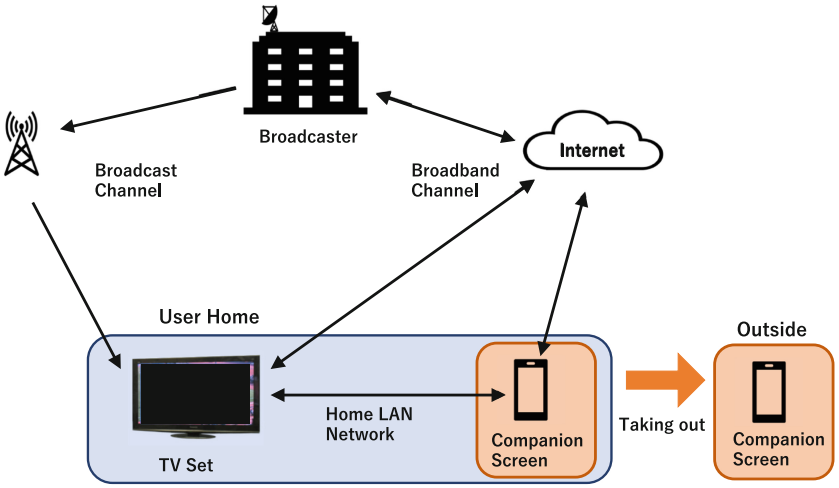

Fig. 1. Basic architecture of a mobile IBB system

Most IBB systems for mobile will use a cloud server, primarily because the space for hardware can be reduced, and all parts of the data can be managed centrally at the cloud server. When using the cloud server, one of the most important issues is how to protect personal information. A cloud 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 [12] shows that the security problems are the threats in the cloud; these threats along with solutions to them have been summarized by Tang et al. [17]. 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 an IBB system with a general encryption scheme, in which an encryption key and a decryption key are one-to-one, is not efficient. 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. As a result, when the number of users becomes large, this becomes inefficient.

As a solution to the above, we focus on a scheme called *verifiable attribute-based keywords search (VABKS)* [16], 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 keywords while encrypting them and to verify the search results, it can fulfil the requirements for our proposed IBB service model (described in Sect. 3). With the above characteristics, we can satisfy the requirements of our service model by using VABKS. One disadvantage is that, although VABKS is highly functional, the search time is lengthy and it is not possible to construct a system with a real-time property. Due to the properties of VABKS, privacy preservation cannot be completely ensured at this point because only limited entities can generate ciphertexts without ingenuity. In this paper, we propose an efficient and secure system that outsources a large amount of data to a cloud server and preserves multiple kinds of personal information by using VABKS.

1.2 Related Works

We focus on VABKS because of its high affinity with IBB services. VABKS can trace its roots to *attribute-based encryption (ABE)* [7], which does not need to generate a distinct encryption key for each entity, and a *searchable encryption* [11], whose ciphertexts can be searched while encrypted.

In the case of an ABE scheme, only entities having an attribute that satisfies an access-control policy can decrypt a ciphertext, where decryption keys or ciphertexts are generated based on attributes (residential place, sex, age, membership type, industry type, reputation, etc.). Therefore, one of the solutions for combatting illegal access is to use an ABE. The first ABE scheme was proposed as an extension of identity-based encryption (IBE) by Sahai et al. [7]. By using VABKS, data such as personal information is encrypted only once. That is, because only the attributes of the service provider that can decrypt at the time of encryption are specified, it is possible to efficiently distribute personal information to multiple service providers. ABE schemes are classified as either key-policy ABE (KP-ABE) [8] or ciphertext-policy ABE (CP-ABE) [9]. 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 [19]. One of the solutions for security threats in a cloud is to use a searchable encryption. As for the notion of searchable encryption, in a method proposed by Song,

Wagner, and Perring [11], 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) [11] or public-key encryption with keyword search (PEKS) [6]. PEKS was integrated by Wang et al. [13] 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 [10]. However, it is inefficient because it requires composite-order bilinear groups. *Attribute-based keyword search (ABKS)*, proposed in [13–15], satisfies the properties of both ABE and searchable encryption. An ABKS scheme with efficient user revocation was proposed by Sun et al. [15]. A novel cryptographic primitive *verifiable* ABKS called VABKS was proposed by Zheng et al. [16], who proposed a generic construction of VABKS that enables anyone to verify the result of a keyword search.

1.3 Contributions

In this work, we propose a data-providing system with feedback to providers while preserving personal information by applying VABKS. There are three contributions: a secure IBB system, an efficient keyword search system, and an interactive system with feedback to providers.

Privacy-Preserving IBB System. A secure IBB system with cryptographic techniques has not yet reached the point of practical application. We construct a secure IBB system by applying VABKS, which has many functions. The first is to control user access to a cloud. The users have attributes and they can access the database only if their attributes satisfy an access-control policy, since access is controlled by the attributes in our system. Our system also has functions to search keywords, while keeping the viewing history and location data encrypted, and to verify the result of the keyword search.

The proposed system preserves privacy by encrypting the user's personal information as search tokens. When a data owner outsources his or her data to a cloud server, our system protects the data by means of the symmetric encryption algorithm included in the VABKS scheme. We can then achieve a secure system that exploits personal information while personal information is preserved.

Two-Stage Keyword-Search System. Although VABKS is highly functional, its heavy computing and searching load means we cannot obtain a real-time property just by applying VABKS to a system without any ingenuity. We therefore construct an efficient *two-stage* system that incorporates the VABKS scheme. It is assumed that a viewer has two kinds of personal information: viewing history and location data. Viewing history is used for narrowing down the keywords and data files in the database to make the user's unique database, and location data is used for queries to the user's unique database. We implement a two-stage

system and a one-stage (normal) system with VABKS. In the one-stage system, the time required for the search increases linearly. This can be a fatal defect for IBB services since virtually all of them require quick responses in real-time. In our two-stage keyword-search system, we narrow down data relevant to the user in advance. As a result, if the number of keywords is about fifty, it takes about 1 s to search for keywords. Therefore, the proposed system is more efficient than a system that only repeats keyword searches at one stage.

Feedback to Provider. In our proposed service model, we consider not only the business operator that provides the information to the user but also the feedback that is given to the business provider. Concretely, in the service model without feedback, a business operator only provides data and can not know who uses the data or how many users are accessing it. If the business operator knows that, it is possible to provide a personalized service or a new service utilizing statistical analysis. This would also enable interactive communication between users and service providers.

2 Envisioned IBB Service

We assume a case where an IBB service is enjoyed in a mobile environment and a viewing history is stored in a mobile communication device that can be moved around. It is possible to use not only information related to the viewing history of the broadcast program but also the location information of the user by using other functions (e.g., the GPS installed in the mobile device). As the information that can be provided increases, better services will be possible. For example, when a user is physically located in a place related to a broadcast program that was previously viewed, services will soon be able to provide the user with information about this place, namely, to provide on-the-spot information etc.

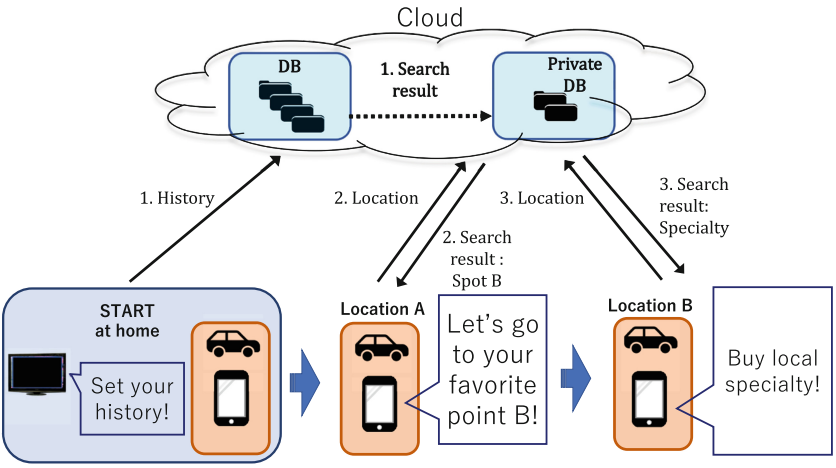

Fig. 2. Service model

2.1 Service Model

Users obtain the service in the mobile environment and we extend existing services to integrated services with mobile (see Fig. 2). For example, when a user drives somewhere related to the viewed broadcast program, he or she is provided with on-the-spot information about that place.

There are four entities involved in this model: providers, users, cloud, and manager. Each provider provides data to the cloud and each user uses the data. It is natural to use the cloud to reduce costs, but there are risks of information leakage.

Users send their histories and their location data to the cloud and the cloud chooses some data to recommend from among all the data provided by providers in accordance with the user's history and location. The user's location data is obtained by using the GPS function of the mobile device. As the provided data increases, better services will be possible, but the time it takes to choose what data to recommend will also increase. In addition, there still remains the crucial issue of preserving personal information (such as user histories).

We assume that all subscribers (users) can enjoy an IBB service by using a mobile terminal and that each user's viewing history is stored in his or her own mobile device, as shown in Fig. 2. It is possible to use not only information related to the viewing history of the broadcast program but also the location data of the user by using other functions such as GPS. For example, when the user is physically located in a place related to the viewed broadcast program, the services will inform the user of nearby places related to the program.

In addition, the service provider provides information on recommended programs and the like according to the user's preference from the viewing history. When a user drives a car, the provider provides certain information based on both the user's viewing history and the current location of the car. Private navigation-guide services like that provide on-the-spot information. There are other cases, where the service provider recommends TV programs related to the driving route. In such a case, 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 private navigation guide service as a form of IBB service.

2.2 Requirements

Since the proposed IBB service involves driving a car, high latency will significantly affect the system that provides "on-the-spot" information. This means the proposed system needs to satisfy a low-latency property: real-time property, as we say. In addition, the service utilizes the user's viewing history and location data. Such data comprises critical personal data and naturally must be protected. Data held by the service providers should not be leaked to unauthorized entities and the recommended information to users should be sent only to the target user. As for the recommended information, it should not be modified. We thus define the requirements of this service as follows.

- **System requirements**
  - Light load: The user can access the service in real-time without high latency.
  - Efficient key management: Key management costs of each entity should be light.
  - Secret searchability: The cloud server can search without decryption.
  - Feedback possibility: The service provider can know which users access the data.
- **Security requirements**
  - Privacy preservation: There are three considerations here. One, the user's personal data should not be leaked to any entities. Two, the source data held by service providers should not be leaked to any unauthorized entities. Three, the personalized data sent to a certain user should not be leaked to any entity other than that user.
  - Verifiability: The personalized 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

3.1 Cryptographic Tools

We describe the cryptographic schemes and algorithms used in our proposed system. Let $a \leftarrow S$ denote selecting an element a from a set S uniformly at random.

Symmetric Encryption. Let SE be a symmetric encryption, such that $\mathsf{SE} = (\mathsf{KeyGen}_{\mathsf{SE}}, \mathsf{Enc}_{\mathsf{SE}}, \mathsf{Dec}_{\mathsf{SE}})$, where $\mathsf{KeyGen}_{\mathsf{SE}}$ is used for generating a symmetric key, $\mathsf{Enc}_{\mathsf{SE}}$ is used for encrypting a message, and $\mathsf{Dec}_{\mathsf{SE}}$ is used for decrypting the ciphertext.

Digital Signature. Let Sig be a digital signature, such that $\mathsf{Sig} = (\mathsf{KeyGen}_{\mathsf{Sig}}, \mathsf{Sign}_{\mathsf{Sig}}, \mathsf{Verify}_{\mathsf{Sig}})$, where $\mathsf{KeyGen}_{\mathsf{Sig}}$ is used for generating a pair of public and private keys, $\mathsf{Sign}_{\mathsf{Sig}}$ is used for generating a signature for a message, and $\mathsf{Verify}_{\mathsf{Sig}}$ is used for verifying whether the message matches the signature.

Attribute-Based Encryption. CP-ABE is defined as follows.

Let $\mathsf{Setup}_{\mathsf{ABE}}$ be an algorithm that generates public parameter $\mathsf{pm}_{\mathsf{ABE}}$ and master key $\mathsf{mk}_{\mathsf{ABE}}$. Let $\mathsf{KGen}_{\mathsf{ABE}}$ be an algorithm that takes $\mathsf{pm}_{\mathsf{ABE}}, \mathsf{mk}_{\mathsf{ABE}}$, and attribute $\mathcal{A}$ as inputs and outputs secret key $\mathsf{sk}_{\mathsf{ABE}}$. Let $\mathsf{Enc}_{\mathsf{ABE}}$ be an algorithm that takes access-control policy $\mathcal{P}$ and data d as inputs and outputs ciphertext c. Let $\mathsf{Dec}_{\mathsf{ABE}}$ be an algorithm that decrypts d from c by using $\mathsf{sk}_{\mathsf{ABE}}$. As a result, $\mathsf{ABE} = (\mathsf{Setup}_{\mathsf{ABE}}, \mathsf{KGen}_{\mathsf{ABE}}, \mathsf{Enc}_{\mathsf{ABE}}, \mathsf{Dec}_{\mathsf{ABE}})$ is given as the attribute-based encryption scheme.

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

Verifiable Attribute-Based Keyword Search. A verifiable attribute-based keyword search scheme (VABKS) can verify the correctness of search results. Let data collections (also referred to as "data sets") $\mathsf{D} = (\mathsf{KS}, \mathsf{MP}, \mathsf{FS})$ denote a set of data files, keyword sets, and their indexes. $\mathsf{KS} = \{\mathsf{KS}_1, \ldots, \mathsf{KS}_n\}$ is a set of n keyword sets, in which elements are encrypted with the same access-control policy. $\mathsf{MP} = \{\mathsf{MP}(w) | w \in \bigcup_{i=1}^{n} \mathsf{KS}_i\}$ is the set of $\mathsf{MP}(w)$ that consists of a set of identifiers for identifying data files associated with keyword $w \in \bigcup_{i=1}^{n} \mathsf{KS}_i$. $\mathsf{FS} = \{\mathsf{F}_1, \ldots, \mathsf{F}_N\}$ is a set of N data files. VABKS consists of six algorithms ($\mathsf{Setup}_{\mathsf{VABKS}}$, $\mathsf{KeyGen}_{\mathsf{VABKS}}$, $\mathsf{BuildIndex}_{\mathsf{VABKS}}$, $\mathsf{TokenGen}_{\mathsf{VABKS}}$, $\mathsf{SearchIndex}_{\mathsf{VABKS}}$, $\mathsf{Vrfy}_{\mathsf{VABKS}}$) defined as

- $(\mathsf{mk}_{\mathsf{VABKS}}, \mathsf{pm}_{\mathsf{VABKS}}) \leftarrow \mathsf{Setup}_{\mathsf{VABKS}}(1^l)$ takes system parameter l as input, and generates public parameter $\mathsf{pm}_{\mathsf{VABKS}}$ and a master key $\mathsf{mk}_{\mathsf{VABKS}}$.
- $\mathsf{sk}_{\mathsf{VABKS}} \leftarrow \mathsf{KGen}_{\mathsf{VABKS}}(\mathsf{mk}_{\mathsf{VABKS}}, \mathcal{A})$ takes $\mathsf{mk}_{\mathsf{VABKS}}$ and $\mathcal{A}$ as inputs and generates a secret key $\mathsf{sk}_{\mathsf{VABKS}}$.
- $(\mathsf{Au}, \mathsf{Index}, \mathsf{D}_{\mathsf{cph}}) \leftarrow \mathsf{BuildIndex}_{\mathsf{VABKS}}(\mathsf{D}, \mathcal{P}, \mathcal{P}')$: Auxiliary information Au, Index and data ciphertext $\mathsf{D}_{\mathsf{cph}}$ are obtained by running this algorithm by encryption of $\mathsf{D} = (\mathsf{KS}, \mathsf{MP}, \mathsf{FS})$, where $\mathcal{P}$ is a set of access-control policies to encrypt n keyword groups in KS and $\mathcal{P}'$ is a set of access-control policies for encrypting N data files in FS.
- $\mathsf{tk}_{\mathsf{VABKS}} \leftarrow \mathsf{TokenGen}_{\mathsf{VABKS}}(\mathsf{sk}_{\mathsf{VABKS}}, w)$ issues a search token $\mathsf{tk}_{\mathsf{VABKS}}$ with credential $\mathsf{sk}_{\mathsf{VABKS}}$ and a keyword w.
- $(\mathsf{proof}, \mathsf{rslt}) \leftarrow \mathsf{SearchIndex}_{\mathsf{VABKS}}(\mathsf{Au}, \mathsf{Index}, \mathsf{D}_{\mathsf{cph}}, \mathsf{tk}_{\mathsf{VABKS}})$ searches Index and outputs search result rslt and a proof proof.
- $\{0, 1\} \leftarrow \mathsf{Verify}_{\mathsf{VABKS}}(\mathsf{sk}_{\mathsf{VABKS}}, w, \mathsf{tk}_{\mathsf{VABKS}}, \mathsf{rslt}, \mathsf{proof})$ verifies $(\mathsf{rslt}, \mathsf{proof})$ with respect to search token $\mathsf{tk}_{\mathsf{VABKS}}$.

Correctness of VABKS requires that, given $(\mathsf{mk}_{\mathsf{VABKS}}, \mathsf{pm}_{\mathsf{VABKS}}) \leftarrow \mathsf{Setup}(1^l)$, $\mathsf{sk}_{\mathsf{VABKS}} \leftarrow \mathsf{KGen}_{\mathsf{VABKS}}(\mathsf{mk}_{\mathsf{VABKS}}, \mathcal{A})$, for any keyword-based data collections D and a keyword w, $(\mathsf{Au}, \mathsf{Index}, \mathsf{D}_{\mathsf{cph}}) \leftarrow \mathsf{BuildIndex}_{\mathsf{VABKS}}(\mathsf{D}, \mathcal{P}, \mathcal{P}')$, $\mathsf{tk}_{\mathsf{VABKS}} \leftarrow \mathsf{TokenGen}_{\mathsf{VABKS}}(\mathsf{sk}_{\mathsf{VABKS}}, w)$, and $(\mathsf{proof}, \mathsf{rslt}) \leftarrow \mathsf{SearchIndex}_{\mathsf{VABKS}}(\mathsf{Au}, \mathsf{Index}, \mathsf{D}_{\mathsf{cph}}, \mathsf{tk}_{\mathsf{VABKS}})$, $(\mathsf{rslt}, \mathsf{proof})$ always holds $1 \leftarrow \mathsf{Verify}_{\mathsf{VABKS}}(\mathsf{sk}_{\mathsf{VABKS}}, w, \mathsf{tk}_{\mathsf{VABKS}}, \mathsf{rslt}_{\mathsf{VABKS}}, \mathsf{proof})$.

3.2 Preparations for Design

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

Entities. In this service, there are four entities: a trusted authority (TA) that is a trusted third party such as a system administrator, a data owner (DO) such as a broadcasting station, a data user (DU) who is both a subscriber and a driver, and a cloud server (CL). We define the roles of these four entities as follows.

DU: DU obtains personalized data from DO. This data is based on DU's personal information. DU does not want to leak his personal information to anyone else. DU has his own attribute. If the attribute satisfies DO's access-control policy, DU can access the data provided from DO.

DO: DO has a large amount of data. A data collection consists of data files, keyword sets, and their indexes. DO has an access-control policy and controls which users can access the data it provides. DO does not provide DU with the data directly; instead, it passes the data to DU via CL. DO does not give the data to anyone except for the authorized DUs.

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

TA: TA authorizes all DUs and DOs.

3.3 Security Model

Here, we show the security model for the proposed system. We define the security requirements and attack models 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 assuming the worst case. A malicious model is thus adopted for CL. It assumes that CL might not follow the protocol and might illegally modify the data. Moreover, it might analyze the data stored on the cloud server in order to learn personal information about DU or obtain data files belonging to DO.

It is assumed that DUs are honest-but-curious entities, which means they follow the protocols but try to access the data in CL. If DU do not follow the protocols, they cannot enjoy the service, but they might illegally obtain another user's data in CL. From the viewpoint of privacy preservation, DU's personal information should not be leaked anywhere. DUs have several types of personal information that is used as search keywords, and they want to obtain data files possessed by DO according to DU's personal information.

In the proposed system, DO plays the role of service provider. If DO is not an honest entity, the system does not stand. If DO provides incorrect data files, incorrect information will be sent to DU and the service will not work. Thus, an honest model is adopted for DO.

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

1. **Data secrecy:** The cloud server should not deduce any keyword from the encrypted data files and search tokens. Data secrecy requires data sets whose data files and keyword sets should be unrecoverable from encrypted data.
2. **Unlinkability of search tokens:** CL cannot link one search token to another even if they are for an identical keyword. Unlinkability of search tokens requires a non-deterministic search-token-generation function, and queries must be properly represented and securely encrypted.
3. **Data unforgeability:** The malicious CL cannot forge any correctly encrypted data files.
4. **Verifiability of search results:** If the malicious CL returns an incorrect search result, DU can detect the cheating behavior.
5. **Collusion resistance:** The honest-but-curious DU can collude with other DUs in order to obtain another DU's secret key or access a personal database

in CL. Collusion resistance requires that no DU be able to decrypt another DU's ciphertexts in an encrypted personal database even if all DUs except for the authorized DUs collude.

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

1. **Known ciphertext model.** The cloud server can only 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 a cloud server attacks with intent to forge encrypted data, or to obtain information concerning DU's keywords or DO's data files.
2. **Collusion attack model.** In addition to the known ciphertext model, DU colludes with another DU. They collude with the aim of obtaining another user's data possessed by the DO or another user's secret key.

In the following section, we show the concrete construction of our proposed system. It uses a VABKS scheme and 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 (see Fig. 3).

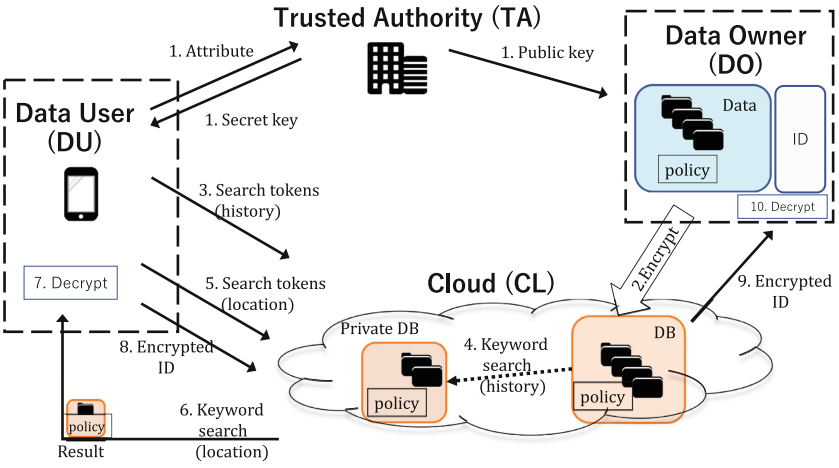

Fig. 3. Proposed system

4.1 System Overview

The system utilizes the VABKS scheme. The algorithms of key generation, encryption, decryption, and keyword search outlined below are those of VABKS.

TA generates public information including an encryption key, generates decryption keys using DU's attributes, and distributes each decryption key to a distinct DU. The data is encrypted by DO, who also determines its access policy. The encrypted data and DO's public key, which is its original one and not identical with that of VABKS, are outsourced to CL. When DU accesses the data, he generates search tokens from his history and sends them to CL. CL searches for encrypted data that matches the tokens and uses it to construct DU's private database. 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 that the size of the private database 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 private database and returns it to DU. DU decrypts the data and uses it. He or she also encrypts his or her ID and the data's ID by using the public key attached to the data and then sends it to CL. CL adds the encrypted ID to the data that DU used. DO can know which DU that accessed the data.

4.2 Construction

Here, we show the concrete construction of our proposed system with VABKS. The basic idea underlying this construction is two-stage keyword searching. The first is a preliminary stage to narrow down the large size of the original data collections into a small size more suitable for DU's personal database. The second is a real-time stage to search personalized data, hence enabling high-speed retrieval from the smaller database on the basis of DU's keywords (e.g., personal information). DO's access-control policy $\mathcal{P}$ for the data and keyword search can be specified by DO when it encrypts data in CL. DU possessing attribute $\mathcal{A}$ satisfying $\mathcal{P}$ can search encrypted data.

The proposed system consists of Sig, SE, ABE, and VABKS. This system, called "SYSV", is shown below. Let $\{l_1, l_2, l_3\}$ be the security parameters. In this system, we assume $\mathcal{P} = \mathcal{P}'$.

Set-up phase: Given security parameters l_1, l_2, the TA executes $(\mathsf{mk}, \mathsf{pm}) \leftarrow \mathsf{Setup}_{\mathsf{VABKS}}(1^{l_1})$ and $(\mathsf{mk}_{\mathsf{ABE}}, \mathsf{pm}_{\mathsf{ABE}}) \leftarrow \mathsf{Setup}_{\mathsf{ABE}}(1^{l_2})$. It then sets $\mathsf{mk} = (\mathsf{mk}, \mathsf{mk}_{\mathsf{ABE}})$ which the TA keeps secret, and $\mathsf{pm} = (\mathsf{pm}, \mathsf{pm}_{\mathsf{ABE}})$, which is publicly known.

Key-generation phase: TA obtains $\mathcal{A}$ from DU, then generates $\mathsf{sk} = (\mathsf{sk}, \mathsf{sk}_{\mathsf{ABE}})$ such that $\mathsf{sk} = \mathsf{KeyGen}_{\mathsf{VABKS}}(\mathsf{mk}, \mathcal{A})$ and $\mathsf{sk}_{\mathsf{ABE}} = \mathsf{KeyGen}_{\mathsf{ABE}}(\mathsf{mk}_{\mathsf{ABE}}, \mathcal{A})$. TA returns sk to DU.

Data-encryption phase: DO executes the following algorithm.

1. Generate a symmetric encryption key $\mathsf{sk}_{\mathsf{SE}} \leftarrow \mathsf{KeyGen}_{\mathsf{SE}}(1^{l_3})$.

2. Encrypt data $(d_1, \ldots, d_N)$ by $c_{d_i} \leftarrow \mathsf{Enc}_{\mathsf{SE}}(\mathsf{sk}_{\mathsf{SE}}, d_i)$ and generates encrypted data set $\mathsf{CD} = \{c_{d_1}, \ldots, c_{d_N}\}$.
3. Encrypt symmetric key $\mathsf{sk}_{\mathsf{SE}}$ by $c_{\mathsf{sk}} \leftarrow \mathsf{Enc}_{\mathsf{ABE}}(\mathsf{sk}_{\mathsf{SE}}, \mathcal{P})$ using DO's policy $\mathcal{P}$.
4. Execute $(\mathsf{Au}, \mathsf{Index}, \mathsf{CPH}) \leftarrow \mathsf{BuildIndex}_{\mathsf{VABKS}}(\mathsf{D}, \mathcal{P}, \mathcal{P}')$ and encrypt each keyword $(w_{i,1}, \ldots, w_{i,n})$ of the data d_i, which is in keyword set KS, and then generate encrypted keywords $\mathsf{CPH} = \{\mathsf{cph}_{i,j}\}_{i,j}$ and auxiliary information $\mathsf{Au} = (\sigma, \mathsf{BF}, \mathsf{vk}_{\mathsf{Sig}})$, where σ is a digital signature for each keyword set, BF is a bloom filter and $\mathsf{vk}_{\mathsf{Sig}}$ is a verification key for SIG.
5. Set index set MP, which represents the relation between CD and CPH, and send $c = (\mathsf{CD}, c_{sk}, \mathsf{CPH}), \mathsf{MP}, \mathsf{Au}$ to CL.

Narrowing-down phase (preliminary stage): CL narrows down the encrypted database as follows.

1. After CL receives c, MP, and Au from DO, they are stored in a database on CL. Note that all data in the database are encrypted, except MP and Au.
2. DU takes keywords $\mathbf{v} = \{v_1, \ldots, v_q\}$ and sk as inputs, generates $\mathsf{tk}_v \leftarrow \mathsf{TokenGen}_{\mathsf{VABKS}}(\mathsf{sk}, v)$, and sends $\mathsf{tk}_{\mathbf{v}} = \{\mathsf{tk}_{v_1} \ldots, \mathsf{tk}_{v_q}\}$ to CL.
3. CL executes the search algorithm of VABKS for $\mathsf{cph}_{i,j}$ by using a search token $\mathsf{tk}_{\mathbf{v}}$, and it outputs $(\mathsf{proof}', \mathsf{rslt}') \leftarrow \mathsf{SearchIndex}_{\mathsf{VABKS}}(\mathsf{Au}, \mathsf{cph}_{i,j}, \mathsf{tk}_{\mathbf{v}})$, where proof' is a certification that includes σ, and rslt' denotes the result of the search.
4. Let encrypted data set $\mathsf{CD}_{DU} = \{c_{d_1}, c_{d_2} \cdots\}$ be in the user's personal database which corresponds to $\mathsf{cph}_{i,j}$ by using the index set MP. Let $\mathsf{CPH}_{\mathrm{DU}} = \{\mathsf{cph}_{\mathrm{DU},1}, \mathsf{cph}_{\mathrm{DU},2}, \cdots\} \subseteq \mathsf{CPH}$. Then, CL stores $c_{DU} = (\mathsf{CD}_{\mathrm{DU}}, \mathsf{CPH}_{\mathrm{DU}})$ in DU's personal database.

Query phase (real-time stage): DU queries CL with DU's personal database, which is smaller than the original encrypted database, as follows.

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

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

1. DU verifies whether the results of searching by CL are forged; namely, $\{0,1\} \leftarrow \mathsf{Verify}_{\mathsf{VABKS}}(\mathsf{sk}, (\mathbf{v}, g), (\mathsf{tk}_{\mathsf{v}}, \mathsf{tk}_g), (\mathsf{proof}, \mathsf{proof}'))$.
2. If Verify outputs 0, DU terminates decryption phase; otherwise, DU decrypts $\mathsf{sk}_{\mathsf{SE}} \leftarrow \mathsf{Dec}_{\mathsf{ABE}}(\mathsf{sk}_{\mathsf{ABE}}, c_{\mathsf{sk}})$ only when DU's attribute $\mathcal{A}$ satisfies DO's policy $\mathcal{P}$. DU then obtains $d_i \leftarrow \mathsf{Dec}_{\mathsf{SE}}(\mathsf{sk}_{\mathsf{SE}}, c_{d_i})$ according to $\mathbf{v}$ and g.

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

If DU uses d_i, DU encrypts ID_u, $c_{\mathsf{ID}} \leftarrow \mathsf{Enc}_{\mathsf{PKE}}(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{ID}_u)$ and send it to CL with the index i.

CL add the c_{ID} to the line of c_{d_i}.

DO accesses CL and obtains c_{ID} added to the data CD. DO knows who uses DO's data by executing $\mathsf{ID} = \mathsf{Dec}_{\mathsf{PKE}}(\mathsf{sk}_{\mathsf{PKW}}, c_{\mathsf{ID}})$.

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

4.3 System Security

We show that SYSV has the properties of data secrecy, unlinkability of search token, 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 chosen-keyword attack (CKA) in the generic bilinear group model and* ABE *and* SE *are secure against chosen-plaintext attack (CPA), SYSV achieves data secrecy and unlinkability of search tokens in the known ciphertext model.*

Proof. We show that if there exists a polynomial-time algorithm A that breaks SYSV's data secrecy and search token unlinkability with the advantage ε, we can construct a polynomial-time algorithm B that breaks CPA security for either ABE or SE with the advantage of $\frac{\varepsilon}{N^2}$, or selective security against the CKA game of VABKS with the advantage of $\frac{\varepsilon}{(NM)^2}$, 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 bound to NM.

We consider two cases: (i) the challenger proceeds with a conventional CPA security game with A, or (ii) it proceeds with a selective security against CKA game with A. In the challenge phase, suppose A presents two data collections $\mathrm{D}_0 = (\mathrm{KS}_0 = \{\mathrm{KS}_{(0,1)}, \ldots, \mathrm{KS}_{(0,\mathrm{M})}\},\ \mathrm{MP},\ \mathrm{FS}_0 = \{\mathrm{FS}_{(0,1)}, \ldots, \mathrm{FS}_{(0,\mathrm{N})}\})$, $\mathrm{D}_1 = (\mathrm{KS}_1 = \{\mathrm{KS}_{(1,1)}, \ldots, \mathrm{KS}_{(1,\mathrm{M})}\}, \mathrm{MP}, \mathrm{FS}_1 = \{\mathrm{FS}_{(1,1)}, \ldots, \mathrm{FS}_{(1,\mathrm{N})}\})$ and policy $\mathcal{P}$.

(i) The challenger selects $\lambda \leftarrow \{0,1\}$ and encrypts FS_λ with ABE and $\mathcal{P}$. Now let us consider the advantage of 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 $\frac{\varepsilon}{N^2}$ by selecting one data file from FS_0 and one from FS_1.

(ii) The challenger selects $\lambda \leftarrow \{0,1\}$ and encrypts KS with VABKS. Since ABE is CPA-secure, the probability of A inferring λ is negligible. Then, let us consider the advantage of 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 $\frac{\varepsilon}{(NM)^2}$ by selecting one keyword from KS_0 and one from KS_1.

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

Theorem 2. *SYSV achieves data unforgeability and verifiability of search results if* Sig *is non-adaptively unforgeable against a known ciphertext attack.*

Proof. This theorem can be proved from the security definition of Sig directly. Given correct (rslt, rslt', proof, proof') and $(\mathbf{v}, g, \mathsf{tk}_{\mathbf{v}}, \mathsf{tk}_g)$, 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 date files. □

Theorem 3. *SYSV achieves collusion resistance in the collusion attack model if Theorem 1 holds.*

Proof. This theorem can be proved from the security definition of ABE and VABKS directly. Attackers can recover data only if they have enough attributes to satisfy the tree T_0, so at least one user should be valid to satisfy the privilege tree. Even if multiple users collude in the collusion attack model, they are not able to recover the other user's secret key sk or to obtain the other user's personal database c_{DU} since each secret key sk is randomized by secure key generation algorithms $\mathsf{KeyGen}_{\mathsf{ABE}}$ and $\mathsf{KeyGen}_{\mathsf{ABKS}}$. □

5 Properties

The system makes it possible to search keywords, while keeping the history and location data encrypted, and to securely provide more interesting and suitable information for users by preventing leakage of personal information despite using a cloud. In addition, the provider can know which user is accessing its data. Concretely, it realizes the following properties:

Efficient Two-Stage Keyword-Search: We construct a *two-stage* system. In the preliminary phase, the history is used for narrowing down the database in the cloud and making a small, unique database for the user. This small size is what makes the real-time service possible. This two-stage search is not limited to the above application, and if the search process is divided into two processes, the same construction is possible. Moreover, it is possible to construct a multiple-stage search.

Privacy-Preservation: The search tokens from users are encrypted and the data provided to the users are also encrypted. Hence, even the cloud cannot know the information related to the users.

Data Secrecy: Users can access the database only if their attributes satisfy an access-control policy.

Feedback to Provider: Only the provider who handles data used by a variety of users can know exactly who the users are. When the provider charges the user, this function is indispensable.

An integrated service with access control, privacy preservation, data secrecy, and feedback to the provider has not yet reached practical application, so the

Table 1. Property evaluation: comparison of proposed system with conventional system

	Conventional	Proposed
Encryption and decryption load	light	heavy
Key management cost	heavy	light
Secret searchability	—	√
Privacy preservation	√	√
Verifiability	—	√
Unforgeability	—	√

work we report is the first such system. In addition, we implemented the proposed system and made sure of the real-time property.

We evaluate the proposed system from the viewpoints of both properties and performance.

5.1 Property Evaluation

First, we evaluate the system from the viewpoint of properties. We compare our proposed system, SYSV, with the most 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 was very light and its processes very fast due to its use of the symmetric encryption scheme. Regarding CPU load, the trivial system is superior to ours because it only uses the symmetric key encryption scheme. DO has to manage all DUs' encryption keys and its cost increases in proportion to the number of DUs in the trivial system; however, the proposed system, SYSV, can efficiently manage keys by using its unique attributes and policies. Moreover, SYSV can search for the keywords and data files without decryption. In both systems, CL cannot obtain any plaintexts (such as search keywords and data files) from encrypted data since both systems protect data by encryption. However, if all entities share the same secret keys, all data may be leaked, since CL is a malicious entity. In this sense, the trivial system is a secure system only when CL is not malicious. As we mentioned earlier, the proposed system satisfies both verifiability and unforgeability.

5.2 Implementation

We next evaluate the proposed system from the viewpoint of performance. Specifically, we implemented a two-stage searching system to achieve a real-time property and evaluate its effects. Figure 4 shows the relationship between the number of keywords in DO's database and the search time, and the details are shown in Table 2. All algorithms are implemented on a PC whose specifications are as follows: CPU: Intel Core i7-4790(3.60 GHz), memory: 8 GB, OS:

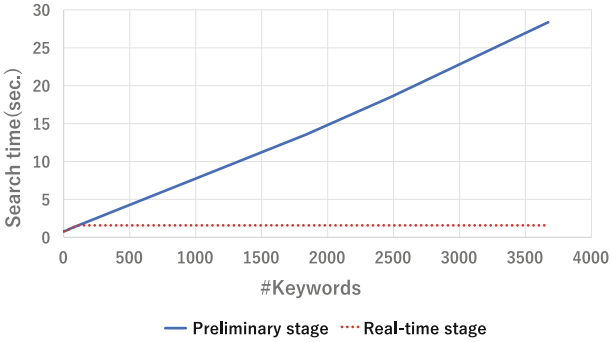

Fig. 4. Relationship between the number of keywords and search time

Cent OS 7.2, and browser: Firefox 38.3.0. Almost all of the encryption algorithms are written in JavaScript (some of them are written in C/C++ due to the limitation of the crypto library). As shown in Figs. 4 and 2, the search time is 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 s to search data files matched to one token. Actually, the number of data files provided by DOs is very large in the IBB system, and the number of tokens may be more than one. We assume that data files of DOs consist of program title data and time along with related keywords (title, genre, cast members, etc.). Therefore, in the case where DO is a broadcaster, the number of data files will increase in accordance with the number of broadcast programs. For example, Ch.1 in Japan has about 300 programs per week and each program has its own keywords. The number of keywords depends on the program and there are at least four keywords per program. The number of all keywords per year is about $62400(keywords/year) = 300(programs/week) \times 52(weeks/year) \times 4(keywords/program)$ for the Ch.1. When the data files of n channels are collected on CL, the number of keywords would become its n-fold ($\approx n \times 62400$). From the experimental results, it takes roughly $n \times (62400/3672) \times 27.7$ sec ($\approx 8n$ min) to search data files matched to one token. When the number of tokens is m, the time rises another m-fold ($8nm$ min). Although the large size of data files enables a better recommendation, it takes too much time to search the recommended data files. If the reply to a query takes a long time, the service will not stand. In the preliminary stage, the number of data files are narrowed down. If CL can narrow down the size to 25 data files including 100 keywords, DU can obtain the data files within about 1.5 s after DU sends a token to CL. Such services would be acceptable to DU.

Table 2. Experiments results (sec). #Keywords denotes the number of keywords in a database of DO. Encryption denotes the process time for encryption algorithm. TokenGen, SearchIndex, Verify, and Decrypt denote TokenGen, SearchIndex, Verify, and Dec algorithms, respectively. Search denotes the summation of TokenGen, SearchIndex, Verify, and Decrypt.

#Keywords	Encryption time	Search (total)			
		TokenGen	SearchIndex	Verify	Decrypt
1	0.657	0.778			
		0.027	0.730	0.012	0.010
114	24.897	1.534			
		0.030	1.348	0.086	0.070
918	196.078	7.059			
		0.028	5.835	0.663	0.532
1836	401.522	13.345			
		0.029	10.941	1.319	1.057
2476	524.732	17.666			
		0.028	14.482	1.753	1.403
3672	804.076	27.674			
		0.029	21.165	3.608	2.871

6 Conclusion

We proposed an IBB system that provides users with information related to TV programs while preserving their personal information by encryption of personal information such as viewing history and location data. We apply a VABKS scheme to the proposed IBB service. As yet, there is no system that preserves personal information such as viewing history in an IBB service. In our system, it is possible to preserve personal information appropriately by cryptographic technology and to eliminate information leakage on a cloud server. As a result, users can access services with peace of mind. A malicious cloud server cannot steal personal information since all the information sent to the cloud server is encrypted.

Multiple different secret keys for each user's attributes are generated in the proposed system, and only one public key is used for encryption due to an access-control policy. 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. Therefore, the burden of key-storage management and key distribution can be reduced compared to that in a system using a one-to-one cryptographic technique such as TLS communication. Also, since keyword search is possible while data are being encrypted, the cloud server does not need to ask the data owner for permission for every search and can efficiently answer the user's inquiries without leaking information.

In addition, since VABKS can verify the validity of the search result, it is possible to detect if data has been forged. The algorithm 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 recommendation results, which is generally not acceptable to users. We therefore proposed a two-stage keyword search system to achieve faster real-time service. The proposed system with heavy load is run at the preliminary stage, and the process with light load is run at the second real-time stage. This is what enables the provision of real-time services. Subsequently, we can construct a system that satisfies both high security and efficiency.

Future Works. In this paper, we did not address how DU generates keywords from viewing history. A smart selection of keywords may enable DO to provide the IBB service with higher satisfaction. Another future work is to consider the optimum size of a personal database that can satisfy the user's preference. The reduction loss in Theorem 1 is loose, since the size of N and M are possibly very large in the proposed system, so another future work is to minimize the security reduction loss.

References

1. Recommendation ITU-R BT.2075-1 (2017). http://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.2075-1-201701-I!!PDF-E.pdf
2. IPTVFJ STD-0010 STD-0011. http://www.iptvforum.jp/download/
3. DigitalUK. Freeview play - technical specification (2016)
4. Etsi technical specification 102 796 v1.4.1. European Telecommunications Standards Institute (2016)
5. ABNT NBR 15606 series (2016). http://www.abntcolecao.com.br/coltv.aspx?Q=CING45WV08&ID=361862
6. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_30
7. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_27
8. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 89–98. ACM (2006)
9. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy, SP 2007, pp. 321–334. IEEE (2007)
10. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_4
11. Song, D. X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: 2000 IEEE Symposium on Security and Privacy. S&P 2000, Proceedings, pp. 44–55. IEEE (2000)

12. The Notorious Nine: Cloud Computing Top Threats in 2013 (2013). https://downloads.cloudsecurityalliance.org/initiatives/top_threats/The_Notorious_Nine_Cloud_Computing_Top_Threats_in_2013.pdf
13. Wang, C., Li, W., Li, Y., Xu, X.: A ciphertext-policy attribute-based encryption scheme supporting keyword search function. In: Wang, G., Ray, I., Feng, D., Rajarajan, M. (eds.) CSS 2013. LNCS, vol. 8300, pp. 377–386. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-03584-0_28
14. Shi, J., Lai, J., Li, Y., Deng, R.H., Weng, J.: Authorized keyword search on encrypted data. In: Kutyłowski, M., Vaidya, J. (eds.) ESORICS 2014. LNCS, vol. 8712, pp. 419–435. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11203-9_24
15. Sun, W., Yu, S., Lou, W., Hou, Y.T., Li, H.: Protecting your right: verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud. IEEE Trans. Parallel Distrib. Syst. **27**(4), 1187–1198 (2016)
16. Zheng, Q., Xu, S., Ateniese, G.: VABKS: verifiable attribute-based keyword search over out- sourced encrypted data. In: INFOCOM, 2014 Proceedings IEEE, pp. 522–530. IEEE (2014)
17. Tang, J., Cui, Y., Li, Q., Ren, K., Liu, J., Buyya, R.: Ensuring security and privacy preservation for cloud data services. ACM Comput. Surv. (CSUR) **49**(1), 13 (2016)
18. Ohtake, G., Safavi-Naini, R., Zhang, L.F.: Outsourcing of Verifiable Attribute-Based Keyword Search. In: Lipmaa, H., Mitrokotsa, A., Matulevičius, R. (eds.) NordSec 2017. LNCS, vol. 10674, pp. 18–35. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70290-2_2
19. Ohtake, G., Ogawa, K., Safavi-Naini, R.: Privacy preserving system for integrated broadcast-broadband services using attribute-based encryption. IEEE Trans. Consum. Electron. **61**(3), 328–335 (2015)