Privacy-Preserving Targeted Advertising Scheme for IPTV Using the Cloud

by

LEYLI JAVID KHAYATI

Submitted to the Graduate School of Sabancı University
in partial fulfillment of the requirements for the degree of
Master of Science

Sabancı University

Fall, 2012

Privacy-Preserving Targeted Advertising Scheme for IPTV Using the Cloud

Approved by:

Assoc. Prof. Dr. Erkay Savaş ....
(Thesis Supervisor)

Assoc. Prof. Dr. Albert Levi ....

Assoc. Prof. Dr.Berrin Yanıkoğlu ....

Assist. Prof. Dr. Cemal Yılmaz ....

Assist. Prof. Dr. Bahattin Koç ....

Date of Approval: 02.02.2012

PRIVACY PRESERVING TARGETED ADVERTISING SCHEME FOR IPTV USING THE CLOUD

LEYLI JAVID KHAYATI

Computer Science and Engineering, Master's Thesis, 2012

Thesis Supervisors: Assoc. Prof. Dr. Erkay Savaş

Keywords: IPTV, Targeted advertising, Privacy, Cryptography, Cloud server

Abstract

Targeted advertising is an emerging business area that provides effective services for advertisers and end users. Advertising agencies need information about users to send them targeted advertisements. However, uncontrolled access to users' sensitive information for advertising purposes violates the privacy of individuals. Therefore, efficient techniques must be used to not only preserve users privacy but also enable advertisers to reach right users.

In this thesis, we present a privacy-preserving scheme for targeted advertising via the Internet Protocol TV (IPTV). The scheme uses a communication model involving a collection of viewers/subscribers, a content provider (IPTV), an advertiser, and a cloud server. To provide high quality directed advertising service, the advertiser can utilize not only demographic information of subscribers, but also their watching habits. The latter includes watching history, preferences for IPTV content and watching time, which are published on the cloud server periodically along with anonymized demographics (e.g. weekly). Since the published data may leak sensitive information about subscribers, it is safeguarded using cryptographic techniques in addition to the anonymization of demographics. The techniques used by the advertiser, which can be manifested in its queries to the cloud, are considered (trade) secrets and therefore are protected as well. The cloud is oblivious to the published data, the queries of the advertiser as well as its own responses to these queries. Only a legitimate advertiser, endorsed with a so-called *trapdoor* by the IPTV, can query the cloud and utilize the query results. The performance of the proposed scheme is evaluated with experiments, which show that the scheme is suitable for practical usage.

IPTV için Bulut Üzerinde Mahremiyeti Koruyan Hedeflemeli Reklamcılık Metodu

LEYLI JAVID KHAYATI

Bilgisayar Bilimi ve Mühendisliği, Yüksek Lisans Tezi, 2012

Tez Danışmanları: Doç. Dr. Erkay Savaş

Anahtar Kelimeler: IPTV, Hedeflemeli reklamcılık, Mahremiyet, Şifreleme, Bulut sunucusu

Özet

Hedeflemeli reklamcılık, reklamcı ve son kulanıcılar için etkin hizmetler sunan gelişmekte olan bir iş alanıdır. Reklam ajanslarının hedeflemeli reklam gönderebilmek için kullanıcılar ile ilgili yeterli bilgiye sahip olmaları gerekir. Ancak kullanıcıların hassas bilgilerine ulaşmak onların mahremiyetlerini tehlikeye atabilir. Bunun için hem kullanıcıların mahremiyetini koruyan hem de reklam ajansının doğru hedef kullanıcılara ulaşmasını sağlayan bir yöntem kullanılmalıdır.

Bu tezde Internet Protokolu üzerinden televizyon servisi (IPTV) için mahremiyeti koruyan bir hedeflemeli reklamcılık yöntemi önerilmiştir. Bu yöntemdeki iletişim modeli belirli sayıda izleyici/abone, bir içerik sağlayıcı (IPTV), bir reklam ajansı ve bir bulut sunucusundan oluşur. Kullanıcılara yüksek kaliteli hedeflemeli reklamcılık servisi sunabilmek için, reklam ajansı hem kullanıcıların demografik bilgilerinden hem de izleme alışkanlıklarından yararlanabilir. İzleme alışkanlığı, kullanıcının tercih ettiği programları ve seyretme zamanlarını içerir ve bu bilgi bulut sunucu üzerinden belli periyotlarla, demografik verinin anonimliği korunarak yayınlanır. Yayınlanan veri, abonelerin hasas bilgilerini açık edebileceği için verinin anonimleştirilmesinin yanı sıra şifreleme teknikleri de kullanılarak verinin güvenliği sağlanır. Reklam ajansının sorguları onun ticari sırları ile ilgili bilgi içerebileceği için korunması gerekir. Bulut sunucusu, yayınlanmış veriler, reklam ajansının sorguları ve bu sorgulara verdiği yanıtlar hakkında bilgi sahibi değildir. Sadece meşru bir reklam ajansı, IPTV'nin verdiği gizli kapıyı (trapdoor) kullanarak bulut üzerinde sorgu yapabilir ve yanıtlarından yararlanabilir. Yapılan deneylerin sonuçları önerilen metodun performansının pratik kullanım için uygun olduğunu göstermektedir.

to my beloved family

Acknowledgments

I wish to express my sincere gratitude to Assoc. Prof. Erkay Savaş and Assist. Prof. Dr. Berkant Ustaoğlu, for their continuous support and worthwhile guidance throughout my masters studies. Assoc. Prof. Erkay Savaş's support starting from my application process till today, was always a great motivation during my studies. Also I am thankful to Assist. Prof. Dr. Berkant Ustaoğlu for believing in the topic I was interested in. In addition, I am thankful to my thesis defense committee members: Assoc. Prof. Albert Levi, Assoc. Prof. Berrin Yanıkoğlu, Assist. Prof. Cemal Yılmaz and Assist. Prof. Bahattin Koç for their support and presence.

I am grateful to TÜRKTELEKOM, for the M.Sc fellowship supports.

I am also immensely thankful to my husband Aidin Gharehchaie for being there when I needed him to be, for believing in me and supporting me throughout all my decisions.

I appreciate Yarkin Doröz, Cengiz Örencik and İsmail Fatih Yıldırım for their help during the implementation process. I would also like to thank to my friends Burcu Özçelik, Duygu Karaoğlan, Bariş Altop, Emre Kaplan, for their help.

Last, but not the least, I am immensely thankful to my family for supporting me throughout all my decisions.

Contents

List of Figures

List of Tables

1 Introduction

Literature suggests [15] that content targeting (e.g. advertisement to customers) is potentially a huge and lucrative business. Traditional media such as TV, radio, or newspaper can do only a little to customize advertisements of products or services for their customers. It is suggested [18] that sponsors are more interested in sending advertisements of their products to prospective customers with high accuracy. Online media offers better opportunities for targeting prospective customers by utilizing customers online history, observed behavior, and demographics. Therefore targeted advertising, where advertisements are selected on the basis of online traits and demographics of individuals, is preferred by many advertising agencies to increase the total benefit gained from advertisement. However, a major issue is the potential violation of the privacy of individuals.

It appears as if IPTV (Internet Protocol TV) is becoming a preferred online media with potentially millions of subscribers. IPTV service providers (henceforth only IPTV) are technically able to store watching history of subscribers and their preferences for IPTV content; they can also obtain personal information by handing out questionnaires when signing a contract with subscribers [12]. Thus IPTV is a hot spot for advertising agencies that have the incentives to utilize the information that IPTV collects about its subscribers. The collected information generates subscriber profiles which are undoubtedly sensitive. Although IPTV is willing to protect subscriber profiles from unauthorized access, he has an economic interest to sell the data to third parties.

As in many other areas the data collected by IPTV is accumulating over time and therefore there is a motivation to outsource data warehousing to a cloud service. Such outsourcing can decrease cost by mitigating the burdens of storage, service management and expenditure on hardware or software [7]. The media attention given to cloud computing suggests that it is gaining a considerable attention in business environments allowing their customers to store and access data remotely. Sensitive data such as personal health records, private videos and photos, email etc. can also be stored on cloud servers. It is suggested to encrypt data before outsourcing to protect the privacy of data and prevent unauthorized access to data in the cloud [13]. However, by encrypting data, one of the key functionality of database systems, i.e. keyword-based search operation, becomes a challenging issue. Many searchable encryption schemes such as [9], [22] lack query flexibility and/or use complicated cryptographic algorithms which require high computational power. The high computational cost may render outsourcing the data collected by IPTV as ineffective solution.

In our setting the advertiser needs to perform multi-predicate queries on a remote database (e.g. implemented in a cloud server) to find records matching with the predicates in his query. Hiding keywords in a query by the advertiser is essential to prevent the disclosure of its trade secrets to the cloud server, to other advertisers and perhaps to IPTV itself. Thus the advertiser needs to query an encrypted database using encrypted keywords. The core challenge we address in this thesis is to facilitate this efficiently.

Our aim is to devise a practical privacy-preserving scheme, whereby each authorized advertiser can perform multi-predicate queries on a remote database (e.g. stored in the cloud server), and sends personalized advertisements to subscribers matching with the predicates in the query. The proposed scheme is different from the previous works on the same subject, in which IPTV operator (i.e. the owner of the database) is also in charge of processing the data and selecting the best advertisements for the subscribers. Naturally, in that setting the privacy of the subscribers is not a major concern. However, in our case the data is outsourced to the cloud which is honest but curious (i.e. the server does not modify the message but is curious to analyze the message content to infer additional information); consequently the privacy of users must be protected. The additional benefits of the proposed scheme are: i) partially relieve IPTV of management cost and processing of data, ii) data mining required for targeted advertising can be performed by advertising agencies, iii) IPTV can generate revenue from subscribers data by providing sufficient data-protection safeguards in comply with relevant legislations [1], and iv) advertiser's mining techniques are not exposed to the cloud server and IPTV. The scheme is also useful when the advertiser is a division within IPTV. In this case, IPTV has robust data protection practices perhaps in accordance with the relevant legislations. That is IPTV keeps sensitive data in encrypted form; data used for targeted advertisement includes only necessary, anonymized information about the subscribers; and access control to data can be exercised in a fine-grained manner.

The rest of the thesis is organized as follows: Section 2 contains background information on structure of IPTV, anonymization and clustering methods, and cryptographic algorithms followed by the related work. In Section 3, motivation and contributions of the thesis are presented. Then we

[1] Current legislations usually stipulate users consent and proper data protection techniques such as encryption and anonymization before disclosure and/or processing.

describe our proposed solutions in detail in Section 4. In Section 5, we argue the extend the privacy requirements are met and in Section 6 the evaluation of our proposed solution is presented. Finally, we conclude the thesis and provide a roadmap for future work in Section 7.

2 Background Information and Related Work

In this section, we explain IPTV system. Then, we give an introductory information about privacy-preserving data publishing techniques. Following that, we explain the clustering techniques and the cryptographic algorithms we utilize. Finally, we conclude the section with the related works in the field of targeted advertising and searchable encryption on the cloud.

2.1 Internet Protocol Television (IPTV)

IPTV is a system, through which a television services are delivered using the Internet Protocol (IP) over broadband instead of through satellite signal or cable television formats [4]. The IPTV service is offered by most telecom operators.

IPTV operator usually provides the following services:

- live television - TV contents simultaneously with the broadcast TV;
- time-shifted television - replays of TV shows that were broadcast earlier;
- video on demand (VOD) - selection from a list of videos not related to TV content.

In traditional television, all programs are broadcast at the same time and the viewer selects the program by changing the channels. In contrast, with IPTV all the streams remain on the service provider's network and only one program is sent at a time due to the bandwidth limitation. Thus transmission of the stream depends on the viewers' program selections. Whenever a channel is changed, a new stream is transmitted to the viewer. Consequently the service provider knows exactly the viewing habits of each viewer, i.e. every program watched, duration of watching, etc. Such information can be used for better provisioning and targeted advertising.

Since IPTV operators are technically able to record users' viewing habits, they can sell that information to third parties such as advertisers [4]. Under normal circumstances an IPTV service provider is assumed to be an honest party and does not divulge any information about an individual user to a third party. However, to generate revenues from the collected data, IPTV can sign an agreement with advertising agencies to send their targeted advertisements to viewers. To diminish the cost and burden of data management and processing, IPTV can outsource viewers' data in encrypted format to a cloud server and enable advertising agencies to query, search, and mine the viewers' data.

2.1.1 Architecture of IPTV

The architecture of IPTV is shown in Figure 1. This figure can be divided into following components [1]:

- Dish, which is placed at the IPTV service provider's head-end and receives the video contents from the broadcasters.

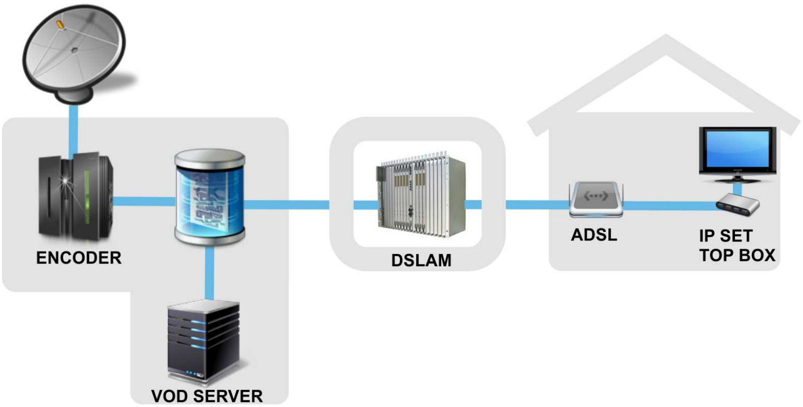

Figure 1: Architecture of IPTV

- Video encoder, encodes the video contents received by dish into IP stream.
- DSLAM (Digital Subscriber Line Access Multiplexer), distributes IPTV streams to customer premises.
- ADSL modem, Set-top box and TV are located at the customer premises. IPTV streams are received by modem and then sent to Set-top box. IP streams are reassembled into a coherent video stream by Set-top box and then are transformed to the format that can be displayed by TV.

2.2 Targeted Advertising

Targeted advertising is a mechanism through which advertisements are sent to the right consumers based on some filtering methods utilizing consumer information such as demographics, viewed items history, geographical location, user behavior, etc. [16]. This mechanism is beneficial for advertising agencies since it maximizes the probability of an advertisement being taken into consideration or clicked by consumers. Techniques for targeted advertising can be divided into two categories:

1. Behavioral targeting: behavioral targeting considers users profile by analyzing watched or bought items in the past and finds other products with similar characteristic. Several online stores such as Amazon [17] use this technique to recommend their products.

2. Contextual targeting: contextual targeting uses the content that is currently watched by a user and suggests similar products. Google's Adsense [2] service uses this technique by exploring the content of web page and estimates the advertisements which are more relevant to the content. For example, if a website visited by a user is related to pop music, it can suggest the advertisements for pop concerts close to the user's location.

In this thesis we provide an efficient scheme for behavioral targeting.

2.3 Privacy-preserving Data Publishing Techniques

In our proposed scheme, subscriber profiles which are considered sensitive information are published on a cloud server. The IPTV has to employ privacy-preserving techniques to anonymize data before outsourcing it to the cloud server. In the following section we explain some of these techniques.

2.3.1 Anonymization

Privacy concerns arise wherever the collected information can identify a person. Information such as healthcare records, financial transactions, home address and geographical records can be used to identify an individual, which, in turn, compromise the privacy of people. The challenge in data privacy is to share or publish data while protecting identifiable information. Shared data is beneficial for different groups of people such as researchers or advertising agencies. There are different techniques such as randomization, generalization, k-anonymity and l-diversity for anonymizing sensitive data.

2.3.2 k-anonymity

k-anonymity guarantees that every record in a dataset is indistinguishable from at least $k-1$ other records with respect to certain identifying attributes. If a record satisfies this condition, it can be published on public server [21]. Data holders mostly remove users' identifier such as social security number when releasing data on public servers to hide users' identity. However, in most cases removing the identity (an other identifier information) will not guarantee the anonymity of users. Released information might still include quasi-identifiers data such as ZIP code, medical records, birth date, etc. which can be linked with other public data to acquire sensitive information about users.

2.3.3 l-diversity

There are two kinds of attributes in a dataset: quasi identifiers, and sensitive attributes. The former one is the set of attributes which can identify a unique user when they are linked with external data. In k-anonymity records are anonymized with respect to their quasi identifiers, however, the sensitive data may remain on the server. An attacker who has some background knowledge about a user can easily estimate the value of a sensitive attribute for the corresponding user from the published data. Hence, it will be more difficult for the attacker to estimate the sensitive attributes if there are l sensitive values for k-blocks of data [14].

2.3.4 Generalization

In this technique general values are used instead of the exact values of an attribute to protect data. In our dataset we take advantage of this method to generalize some attributes. For example the age of subscribers are represented by categorical values such as: 15-20, 20-30, 30-40, etc. In addition, instead of exact location of each subscriber in the dataset, we divide locations into three different categories: low-income areas, middle-income areas, and high-income areas. In our dataset we remove the social identifier (social security number, citizenship number etc.) and name of subscribers when publishing data.

To protect the privacy of subscribers we assume that IPTV applies anonymization techniques before outsourcing data to a cloud server. Clustering can be an alternative to prevent from publishing all subscribers' data on the cloud. However, there are some weaknesses with this method as we explain in Section 3.1. Next we give detailed information about clustering.

2.4 Clustering

Clustering is a process of partitioning the data (or objects) into a set of classes called cluster so that objects in a cluster are similar to one another but are dissimilar to objects in other clusters [5, Chapter 7]. A cluster of data objects can be treated collectively as a single object and therefore clustering is considered as a form of data compression. Thus, clustering techniques can be used for data anonymization purposes. In Section 3.1 we employ a clustering method over data and publish the representative point of each cluster on a cloud server. In that section, we will explain in more details whether this method is beneficial to provide anonymization or not.

2.4.1 Data Types

There are different types of data represented in cluster analysis, and each is preprocessed differently. Suppose that our dataset (for IPTV) to be clustered contains n subscribers, each represented by p variables such as age, marital status, education, occupation, and so on. Therefore the dataset can be represented by a matrix where each row contains data of a subscriber and each column is an attribute.

$$\begin{bmatrix} x_{11} & ... & x_{1f} & ... & x_{1p} \\ ... & ... & ... & ... & ... \\ x_{i1} & ... & x_{if} & ... & x_{ip} \\ ... & ... & ... & ... & ... \\ x_{n1} & ... & x_{nf} & ... & x_{np} \end{bmatrix}$$

Dissimilarities are calculated based on the attribute values describing each object. Having n objects, dissimilarity can be represented by following n-by-n matrix where $d(i, j)$ is the dissimilarity between objects i and j.

$$\begin{bmatrix} 0 & d(1,2) & d(1,3) & ... & d(1,n) \\ d(2,1) & 0 & d(2,3) & ... & ... \\ d(3,1) & d(3,2) & 0 & ... & ... \\ ... & ... & ... & 0 & ... \\ d(n,1) & ... & ... & ... & 0 \end{bmatrix}$$

The main diagonal entries are zero $d(i,i) = 0$, and entries outside the diagonal are symmetric with respect to main diagonal that is $d(i,j) = d(j,i)$. Typically clustering algorithms operate on dissimilarity matrix. Measures of dissimilarity are computed differently for objects represented by different variable types.

In many real databases, objects are represented by different type of variables. We explain three types of variables in our dataset.

"**Binary**" variables: having only two states 0 or 1. For instance gender is a binary variable where 1 may indicate that subscriber is female, while 0 indicates otherwise.

"**Categorical**" variables: are generalization of the binary variables with more than two states. States can be represented by letters or numbers.

"**Ordinal**" variables: resemble a categorical variable, except that the states of the ordinal value are ordered in a meaningful sequence.

To compute the dissimilarity between objects of mixed variable types, it is better to process all variable types together, performing a single cluster analysis. One such technique combines the different variables into a single dissimilarity matrix, bringing all of the meaningful variables onto a common scale of interval [0.0,1.0].

Suppose that the database contains p variables of mixed types. The dissimilarity between objects i and j is calculated as:

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}$$

where $\delta_{ij}^{(f)}$=0 if there is no measurement of variable f for object i (x_{if}) or object j (x_{jf}); otherwise $\delta_{ij}^{(f)}$=1.
$d_{ij}^{(f)}$is computed depending on its type:

- If f is binary or categorical: $d_{ij}^{(f)}$=0 if x_{if}=x_{jf}; otherwise $d_{ij}^{(f)}$=1.

- If f is ordinal: f has M_f ordered states, representing the ranking 1,..., M_f. Each x_{if} is replaced by its corresponding rank r_{if} and $z_{if} = \frac{r_{if}-1}{M-1_f}$ is computed. Then, the distance between i and j is calculated as:

  $$d_{ij}^{(f)} = \frac{|z_{if}-z_{jf}|}{max_h x_{hf} - min_h x_{hf}}$$

  where h runs over all non-missing objects for variable f.

2.4.2 Clustering Methods

There are several clustering algorithms in the literature. A suitable one for our dataset is *partitioning* method. Given a dataset of n objects and k as the number of partitions, a *partitioning* method distributes n objects into k clusters such that the objects within a cluster are similar, while the objects of different clusters are dissimilar in terms of the dataset attributes. There are well-known methods in partitioning such as k-means, k-medoids, and their variations [5, Chapter 7]. We focus on k-medoids method since k-means method is not suitable for categorical attributes in a dataset.

In contrast to k-means method, where the mean value of the objects in a cluster is taken as a reference point, actual objects represent clusters in k-medoids. Thus, one representative object is chosen arbitrarily per cluster. Each remaining object is assigned to the cluster of the closest representative object. The iterative process of replacing representative objects by non-representative objects of the cluster continues as long as the quality of the clusters is improved, that is the sum of the dissimilarities between each object in a cluster and cluster representative point is minimized. The sum of dissimilarities which is called absolute-error is represented by E and calculated as

$$E = \textstyle\sum_{j=1}^{k} \sum_{p \epsilon C_j} \mid p - o_j \mid$$

where p is an object in cluster C_j and o_j is the representative object of C_j.

In Algorithm 1 different steps of generating k clusters using k-medoids partitioning method are shown.

Algorithm 1 K-medoids partitioning algorithm

```
Input:
  • $k$: the number of clusters,
  • $D$: a dataset containing n objects.
Output: $k$ clusters.
  1. choose $k$ objects arbitrarily from dataset $D$ as the initial representative
     objects (a.k.a. medoids, $o_1, \ldots, o_k$);
  2. for each remaining object compute a distance to all representative objects
     and assign the object to the cluster with the nearest representative object;
  3. for each representative object $o_j$, i.e. for $i$ from 1 to $k$,
     (a) For each non-medoid data point $o_r$;
         i. compute the absolute-error, $E$, for the configuration where
            the object $o_j$ is swapped with $o_r$
  4. Select the configuration with the lowest cost;
  5. repeat the steps 2 and 5 until there is no change in the medoids;
```

The k-medoids method is more robust than k-means in the presence of noise and outliers since a medoid is less influenced by outliers or other extreme values than a mean. However, its processing is more costly than the k-means method. Both methods require the user to specify k, the number of clusters.

2.5 HMAC (Hash-based Message Authentication Code)

HMAC is a mechanism to verify the message integrity and authenticity using cryptographic hash function in combination with secret keys [3]. Any cryptographic hash function, such as MD5, SHA-1, SHA-2 can be used in the calculation of HMAC. The cryptographic strength of the HMAC depends on the characteristic of the underlying hash function, the length of a hash value, and the size and quality of secret key.

2.5.1 HMAC definition

The $HMAC$ is defined as :

$HMAC(k, m) = H((k \oplus opad) \| H((k \oplus ipad) \| m))$,

where:

- m is a message.
- k is a secret key.
- H is a cryptographic hash function.
- $\oplus$ denotes XOR (exclusive or).
- $\|$ denotes concatenation.
- *opad* is the outer padding (0x5c5c5c. . . 5c5c, one-block-long hexadecimal constant).
- *ipad* is the inner padding (0x363636. . . 3636, one-block-long hexadecimal constant).

We use HMAC in our proposed method.

2.6 Recommender systems

Recommender systems can assist customers to find products that match with their needs and preferences [6]. They can recommend any content (movie, book, TV program, video on demand, music, advertisement, etc.) or services that are likely to be of interest to the customers. A recommender system estimates a user rating or preference degree for contents not being purchased or used by the user in the past. Estimation is done by comparing a user profile with some characteristics depending on the filtering method used by the recommender. A variety of techniques have been proposed for performing recommendation, including content-based, collaborative, knowledge-based, etc. Sometimes these methods are combined in hybrid recommenders to improve performance.

The best known techniques for recommendation are as follows:

- **Collaborative Filtering:** In this method, the profile of users and their rates to different products are considered to generate different groups of users. Recommendation is offered based on inner-user comparison. That is, the system recommends a product to a user depending on the behavior of other group members.

- **Content-based Filtering:** In this method, products are compared based on their content. Each product content is defined by some features. Recommender system finds user interest by observing the features of products the user has rated before. It recommends products with features that are very similar to the features of the products rated by the user.

- **Hybrid Filtering:** In hybrid recommender systems, two or more techniques are combined to attain better performance. Collaborative and content-based characteristics may be unified in one single model (e.g., simultaneously using user and item attributes to compute ratings).

2.7 Public Key Cryptography

Public key cryptography is a cryptographic system where the keys used to encrypt and decrypt a message are different. Each user has a pair of two keys referred as public key and private key, respectively. The public key is publicly accessible to other entities, while the private key is known only to his owner. Although these keys are mathematically related, it is computationally infeasible to deduce the value of a private key from the corresponding public key. The public key cryptography provides two important cryptographic primitives, namely public key encryption (PKE) and digital signature algortihm (DSA). There are different techniques in public key cryptography such as Diffie–Hellman key exchange, RSA, ElGamal, etc. We use RSA since it is the most popular public key cryptography system.

2.7.1 RSA Public Key Cryptography

RSA algorithm is named after its inventors, Ron Rivest, Adi Shamir, and Len Adleman in 1978. It is an algorithm that is based on the difficulty of factoring large integers.

User of a RSA system performs the following stages to generate his public and private key.

1. Chooses two large primes p, q.

2. Computes $n = pq$.

3. Computes $\phi(n) = (p-1)(q-1)$.

4. Chooses a random integer, $1 < e < \phi(n)$ such that $gcd(e, \phi(n)) = 1$.

5. Computes $d = e^{-1} \pmod{} \phi(n)$ that is $ed = 1 \pmod{} \phi(n)$.

6. Public key: (e, n) and private key: (d, p, q).

In public key encryption, a message is encrypted with the public key of a recipient, and no one except the recipient can decrypt the message. Public key encryption is used for confidentiality to prevent the disclosure of information to unauthorized parties. The encryption and decryption operations are illustrated below:

RSA encryption: $y = m^e \pmod{} n$ where $m < n$

RSA decryption: $m = y^d \pmod{} n$

In digital signatures, a message signed with the private key of a sender, and anyone, who has access to the public key of the sender, can verify a signature as follows:

RSA signature: $s = m^d \pmod{} n$

RSA verification: $m = s^e \pmod{} n$

2.8 Related Work

Most of the previous works which provide targeted content to subscribers of IPTV rely on recommender system techniques. In those works, the most appropriate content is selected for a certain customer using so-called filtering techniques. In [19], the proposed system uses subscribers watching history and program preferences to provide a high-quality program recommendation. The privacy of subscribers is not a major concern in this context since IPTV that already has subscribers' data is in charge of recommending content to users. Kodialam et al. [15] propose a method for on-line scheduling of targeted advertisement for IPTV which chooses a set of advertisements in each time slot and assigns users to one of these selected advertisements. Advertiser's bid for TV advertisement is kept at IPTV side. Assigning advertisement to subscribers is done by considering those bids and users demographic or behavioral information. Konow et al. [16] proposed a recommender system using collaborative filtering for selecting the most appropriate advertisement for a certain customer based on the success that the advertisement has had in the past among other customers having similar preferences. Advertiser provides meta-data about the profile of people the advertisement is aimed at. In this system IPTV processes customers profile data and selects the best advertisement matches with their profiles.

In our scheme, we provide the opportunity for advertising agencies to mine over encrypted data (outsourced by the data owner to a cloud server) and select target viewers for their advertisements without sending meta-data to the IPTV and cloud server (i.e. advertisers mining techniques are not manifested to the cloud server and possibly IPTV itself). In contrast to previous works on similar schemes, our proposal relieve the data owner of the cost associated with managing and processing of the data for advertisement purposes.

As mentioned in the Section 1, sensitive data has to be encrypted before outsourcing it to protect data privacy. However, data encryption hinders traditional data utilization techniques based on plaintext keyword search. Considering the large number of users and documents in the cloud server, it is crucial for the search service to facilitate multi-predicate queries. Many searchable encryption schemes focus on a single keyword search. Wang et al. [22] provide ranked keyword search over encrypted cloud data. In their method the server knows the relevance order of documents containing specific keyword; however, it is limited to single keyword search queries. In the public key setting, Boneh et al. [9] present the first searchable encryption construction using public key cryptosystem to perform search on encrypted data. Several works on multi-keyword search were proposed [11, 10, 8, 20] that enable conjunctive and disjunctive search options, but these schemes incur large overhead in computation and/or communication costs.

An efficient scheme for conjunctive keyword-based search is proposed by Wang et al. [23]. A searchable index is generated for each document which contains all the keywords in the document. They use a cryptographic hash function to map every keyword to a sequence of l bits, where l is the length of hash digest and can be represented by r digit number in base 2^d. However, if the keyword set is small the solution is not secure. In the case of IPTV, the number of keywords in subscribers data is small, so it is easy for the server to guess the keywords in a query by a brute force attack.

In our solution, we adopt the scheme by Wang et al. [23] using keyed hash function (HMAC) to map keywords to a sequence of l bits using a key known only to the data owner. To search the encrypted data, advertisers must in advance obtain so-called secure trapdoors from the data owner. Since those trapdoors are generated using the data owner's secret key, the server is not able to learn any information about the search terms in the advertiser query

In our scenario, the cloud server is in charge of processing queries and sending results back to the advertiser. The proposed solution is unique in the sense that it protects the privacy of both subscribers and advertisers as well as the IPTV business interests. In the following section we provide a motivation for the chosen model.

3 Motivation and Contribution of the Thesis

This section provides information on why we select this subject for research and our contributions.

3.1 Motivation

Collected information about subscribers such as demographic information, watching history and preferences for IPTV contents is of particular interest for the advertising business. On the one hand, the IPTV is in charge of protecting data from unauthorized access, but on the other hand he is willing to sell subscriber profiles to third parties (e.g. advertising agencies) to generate revenues. In this work we aim to propose an efficient targeted advertising system, which does not require the involvement of IPTV in processing of data for advertising purpose, while preserving the privacy of subscribers and advertisers. We aim to place subscribers data on a cloud server and enable advertisers to search over encrypted data. Since subscriber profiles are accessed by the advertisers, we need to anonymize their demographic profiles to prevent any privacy violation of subscribers by advertising agencies. In addition, to enable IPTV to control who access data on the cloud, we need to use a technique that allows only advertisers who purchase access to data to query the database. One approach for anonymization would be to cluster the subscribers having sufficiently similar demographics and watching traits. Then cluster representatives would be placed in the cloud, whereby the advertisers utilize this summary data to match the relevant subscribers with their advertisement portfolio. This technique is similar to the k-anonymity algorithm, but is not the best in our setting due to the following reasons: i) clustering may leak information about some subscribers, ii) the advertisers have to use static clusters formed by the IPTV (loss of precision) and iii) IPTV cannot control who access the data (loss of revenues). Since leaking subscribers information is the most damaging problem from privacy perspective, we elaborate on how much information a malicious entity can gain.

We employ clustering technique on a dataset of 100 subscribers to find whether subscribers data can be compromised or not. For the clustering, we use k-medoids partitioning method [5], which groups n objects into k clusters given a dataset of n objects. The k-medoids is a suitable alternative for our dataset in which data objects have attributes of mixed type (e.g. Binary: gender, Categorical: marital status, Ordinal: education and rate of watching, etc.). Using this method, instead of publishing all subscribers to the cloud server, only cluster representatives are presented.

In the following we show two strategies to gain private information (i.e. extract some features) from clustered data.

- ***Fake users attack:*** Assume that the attacker knows some features from the profile of a target subscriber existing in IPTV dataset. The attacker can generate a couple of fake subscribers whose features are very similar to the target subscriber and register them with the IPTV. Since the IPTV uses k-medoids method, it is probable that the fake subscribers along with the target subscriber are assigned to the same cluster. Furthermore, the target subscriber may become the medoid of the cluster if the features of the fake subscribers are suitably generated by the attacker. If that scenario holds, the attack is successful since the profile of one of the medoids published in the cloud server is the same as the target subscriber. We performed an experiment to test the scenario and the results are given in Table 1. We assumed that the attacker knows all the features of the target subscriber profiles except those listed in the *Unknown features* column in the Table. The number of subscribers is 100, and we apply each attack on 30 different target subscribers. The *Success rate* in the table shows the percentage of attacks that are successful. An attack is considered successful when the target subscriber is assigned to the same cluster with the fake subscribers and as a result the target user becomes the medoid of that cluster. By being the medoid, the target profile is released on the cloud server and the attacker is able to find the unknown features of the target subscriber. For example, the first row in Table 1 refers to an attacker who knows all the target features except the education level. The attacker registers six fake subscribers whose features are exactly the same as the target subscriber, but the education levels have different values. This attack is successful 96.66% of the time when applied to 30 different targets.

dataset size	***# of attacks***	***Avg # of fake subscribers***	***Unknown features***	***Success rate***
100	*30*	*6*	*education*	*96.66*
100	*30*	*7*	*age*	*93.33*
100	30	*14*	*education, age*	*90*
100	*30*	*12*	*education, marital st*	*96.66*
100	*30*	*20*	*education, marital st, gender*	*86.66*
100	*30*	*118*	*education, marital st, gender, age*	*66.66*
100	*30*	-	*education, marital st, gender, age, occupation*	*Not successful*
100	*30*	*10*	*education, watching habits*	*80*
100	*30*	*10*	*education, marital st, watching habits*	*50*

Table 1: Success rate of generating fake subscribers attacks

- ***The periodic updating of watching habits***: Since clusters are generated based on the features of subscribers, updating their watching habits (as they change by time) can change the members of each cluster causing alteration of clusters medoids. In our experiment on dataset of 100 subscribers we choose $k = 10$ as number of clusters. We observe that as the watching habits of the subscribers are updated a significant number of cluster medoids changes. When we run the clustering algorithm for initial dataset, the cluster medoids which are subscribers with following IDs (1,26, 3, 16, 81, 10, 89, 28, 98, 43) are published in the cloud server. By updating the watching habits of 30 subscribers the medoid of the clusters are changed to subscribers with IDs (67, 26, 3, 28, 5, 95, 71, 28, 15, 18). In addition, taking the average of watching habits in two consecutive weeks alter the medoids to subscribers (1, 26, 3, 16, 81, 95, 15, 28, 98, 41). Thus as time progresses, more information about subscribers is available in the cloud.

These observations suggest the need for a more robust solutions. Instead of clustering, privacy-preserving techniques in data publishing such as generalization, k-anonymity, and l-diversity are preferred to be used by IPTV to anonymize demographic profile of subscribers. We propose a solution in the remainder of this work to increase the privacy of subscribers and enable IPTV to control access to data on the cloud server,

3.2 Contribution of the Thesis

In this thesis we focus on several issues involving privacy-preserving targeted advertising scheme for IPTV. The first issue is to provide an efficient technique to encrypt the profile of IPTV subscribers and make it possible for authorized advertising agencies to query over encrypted data. The second issue we deal with is to measure the efficiency of the provided method i.e. how much the query results are related to the search terms and how long does the server response take.

We provide a technique for ranked search, i.e. ranking the retrieved profiles in the order of their relevancies to corresponding query. By this method the advertiser is able to learn more relevant profiles to its queries with the additional benefit of decreasing the communication overhead. The advertisers who are endorsed with trapdoors by the IPTV can query the cloud and utilize the results; since these trapdoors are same for all advertisers, we also design a protocol to prevent advertisers to sell their trapdoors to each other and utilize unauthorized trapdoors.

4 Proposed Privacy-preserving Targeted Advertising Scheme

This section provides detailed explanation for the proposed scheme. We begin by describing the entities in our system and their goal and knowledge. Then we will show the data model and general framework of the system. Finally, we explain the methodology for proposed scheme.

4.1 Entities

There are three entities in our scheme.

- **Data owner** also called IPTV, is an entity that provides content to viewers. It collects information about viewers demographics and weekly watching habits, which is stored in a database. Each individualized entry, called viewer profile, is considered private information. However, statistical aggregation of viewer profiles can be released as long as individual entries cannot be recovered from the released information. To increase profit the data owner is willing to sell any type of information which does not violate the privacy of individual viewers. Furthermore, the data owner aims to reduce management costs by outsourcing to third parties database storage, backup and maintenance. Outsourcing is considered information release and therefore should conform to privacy restriction.

- **Advertisers** are entities that can generate targeted advertisements based on viewer demographics and watching habits. Advertisements are generated based on a secret advertiser strategy that requires as input information about target viewers. Therefore the advertiser is willing to purchase access to any database containing viewer profiles such as a database generated by IPTV. However, since mining rules can reveal information about advertiser strategy, the advertiser wants to keep those rules secret.

- A **server** is a professional entity (e.g. cloud server, CS for short) that offers computing and storage services to any party according to specifications provided by said party. The CS does not deviate from the provided specification but curious to infer any information from the use of its services. We assume that it is against the business interest of the CS to collude with any entity against other entities. As such the

CS is what is known as “honest but curious" entity.

4.2 General framework

The general framework of the proposed system is illustrated in Figure 2.

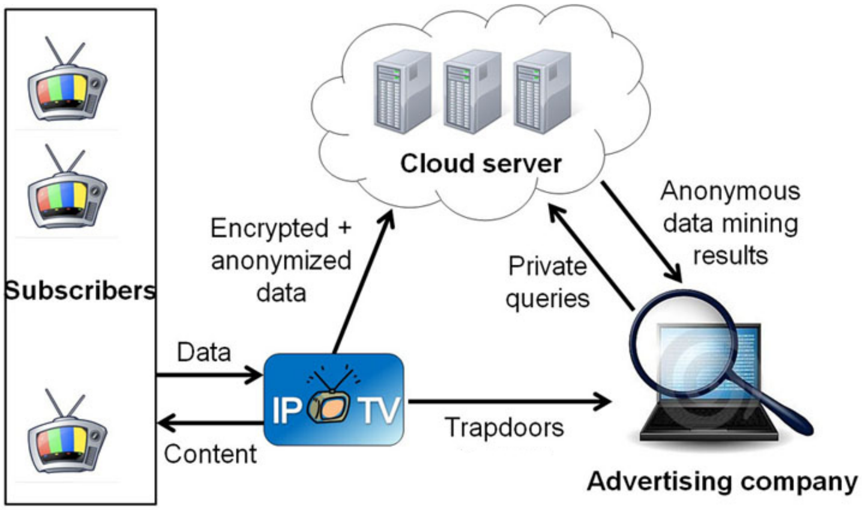

Figure 2: General framework of directed advertising service

IPTV provides personalized program contents to viewers and collects their information. To reduce costs the IPTV is outsourcing its database management to a cloud server. Since the cloud server is an entity external to IPTV, the database can only be outsourced in a form which does not violate the privacy of individual viewers, for example after encryption. Furthermore, the cloud server should not be able to perform mining queries without IPTV assistance and permission. Any prior IPTV assistance should be useful only to the designated entities.

Advertiser purchases access to database stored on the cloud server. To perform mining the advertiser may require assistance from the IPTV. In our solution the database is stored encrypted on the cloud server and therefore the advertiser needs in advance certain trapdoor information generated by the IPTV. It is also important that the trapdoor information is not transferable. In addition, since the advertiser considers its mining rules trades secret, the cloud server should not be able to infer any information about advertiser's queries.

The cloud server in advance obtains protocol specification from IPTV and according to them services requests from authorized entities. Throughout these interactions the CS does not collude with any entity to violate another entity's secret or private information. Furthermore, it does not deviate from the provided specifications.

4.3 Data Model

We consider a scenario where the profile of each subscriber consists of seven demographic features and eleven watching preferences. The demographic features are age, gender, marital status, education, occupation, city, and location; morning watching preferences are marriage, news, or health programs, afternoon – marriage, series, or sport programs, prime time – news, competition, series, talk show, or sport program [2] which are called watching habits. Our synthetic demographic profiles are assigned randomly within the predefined categories (e.g. married, single, widow, or divorced for "marital status") and watching habits are calculated according to these demographic features. The value for a watched program is a number between zero and

[2] Afternoon sport and prime time sport programs are independent fields.

one indicating the rate of time the subscriber spends on watching the program during a week, the sum of these numbers for each week and for each viewer must not exceed one. To provide ranked search to decide how much the retrieved profiles are relevant to the query, we use the relation described in Table 5. For example, the value 0.5 that appears in a viewer profile under prime time news program implies that the viewer under consideration is frequently watching such program.

We also speculate that watching habits of individuals are correlated to their demographic features. Therefore, any synthetic data should take into account such dependencies. We use our custom made data generator which uses probabilistic selection process. The process conforms to some simple expectations about the types of correlations that exist between demographic features and watching preferences as illustrated in Table 2. In Table 2, the rules used for each feature and preference are given. For instance, *rnd* stands for uniformly random selection for all possible values of the corresponding feature while *0.2-0.3* represents uniformly random selection in the interval $[0.2, 0.3]$. These rules are not rigid and do not impose any restriction on our proposed solution. Furthermore, the demographic features and watching preferences can be modified and extended to real world scenarios. Demographic features and watching habits of one instance record (i.e. profile) of the viewers database illustrated in Table 3 and 4 respectively.

	Marriage	***Health***	***News***	***Competition***	***Series***	***Talk show***	***Sport***
15-20 M	*rnd*	*rnd*	*rnd*	*rnd*	*0.2-0.3*	*rnd*	*0.35-0.45*
20-30, 30-40 M illiterate	*rnd*	*rnd*	*rnd*	*rnd*	*0.2-0.3*	*0*	*0.2-0.3*
20-30, 30-40 M educated	*0*	*rnd*	*0.2-0.3*	*rnd*	*rnd*	*0.1-0.2*	*0.2-0.3*
20-30, 30-40 M illiterate	*0.2-0.3*	*rnd*	*rnd*	*0*	*rnd*	*0*	*0.15-0.25*
20-30, 30-40 M educated	*rnd*	*rnd*	*0.2-0.3*	*rnd*	*rnd*	*rnd*	*0.15-0.25*
60+ M illiterate	*0.20-0.25*	*0.1-0.25*	*0.05-0.15*	*0*	*rnd*	*0*	*rnd*
60+ M educated	*0-0.1*	*0.1-0.25*	*0.2-0.35*	*0*	*rnd*	*0*	*rnd*
15-20 F	*rnd*	*rnd*	*rnd*	*rnd*	*0.3-0.4*	*rnd*	*0.15-0.25*
20-30, 30-40 F illiterate	*0.15-0.25*	*0.1-0.15*	*rnd*	*rnd*	*0.2-0.3*	*rnd*	*rnd*
20-30, 30-40 F educated	*rnd*	*rnd*	*0.15-0.25*	*rnd*	*0.2-0.3*	*rnd*	*rnd*
20-30, 30-40 F illiterate	*0.2-0.35*	*0.2-0.25*	*rnd*	*rnd*	*0.3-0.4*	*0*	*rnd*
20-30, 30-40 F educated	*rnd*	*0.2-0.25*	*0.2-0.3*	*rnd*	*0.3-0.4*	*rnd*	*rnd*
60+ F	*0.2-0.35*	*0.1-0.25*	*rnd*	*0*	*0.2-0.35*	*0*	*rnd*

Table 2: Rules for generating synthetic data and correlations of watching preferences and demographics in synthetic database

ID	***Age***	***Gender***	***Marital Status***	***Education***	***Occupation***	***City***	***Location***
1	*20-30*	*f*	*married*	*phd*	*dentist*	*Izmir*	*high-income*

Table 3: Demographic features

Program	***Rate of watch in a week***
marriage(M[a])	*0*
marriage(A[b])	*0*
health(M)	*0.15*
news(M)	*0.1*
news(P[c])	*0.35*
competition(P)	*0*
series(A)	*0*
series(P)	*0.2*
talkshow(P)	*0.05*
sport(A)	*0.05*
sport(P)	*0.1*

[a] *Morning*

[b] *Afternoon*

[c] *PrimeTime*

Table 4: Watching habits

Rate of watch	***Rank***	***Level***
0	*not watched*	*0*
> 0	*seldom*	*1*
>= 0.15	*average*	*2*
>= 0.30	*frequent*	*3*

Table 5: Rate table

4.4 Methodology

Our proposed approach is composed of three phases: Index generation, query generation and oblivious search. In the following sections, we give detailed information on how these phases are performed.

4.4.1 Index Generation

The actual solution is based on the construction proposed by Wang et al. [23]. The main idea is to represent each database record as a binary string. To accommodate n-ranked search each record is expended to n binary strings. Without loss of generality following Table 5 there are three ranks "seldom", "average", and "frequent" according to the rate given to each program per week. Subsequently each database record is cloned three times, once for each level. Each level clone contains the same demographic information but includes a given program field only if the corresponding numerical value in the original record is non zero and exceed the rate of watch lower bound as in Table 5.

Table 6 is an example showing three rank levels for the sample viewer profile in Tables 3, 4. All demographic features appear in three levels. According to Table 5 the level one clone also contains programs having rate greater than 0. Consequently the level two clone contains all the programs having rate equal or greater than 0.15. Finally the third level contains programs with rate equal or greater that 0.3 which in this case is only $news(P)$ program.

Level	***rank***	***keywords***
1	*seldom*	*20-30, f, married, phd, dentist, Izmir, high-income, health(M), health, news(M), news(P), series(P), talkshow(P), sport(A), sport(P)*
2	*average*	*20-30, f, married, phd, dentist, Izmir, high-income, health(M), news(P), series(P)*
3	*frequently*	*20-30, f, married, phd, dentist, Izmir, high-income, news(P)*

Table 6: Rank Levels for a sample profile

We will describe how to generate the binary string records corresponding to rank one, other ranks are generated in an analogous way. Firstly, the IPTV selects a HMAC (Hash-based Message Authentication Code) key. This key is updated for each novel data sent to the cloud server. With m we bound the maximum number of fields in each profile; from now on we call these fields keywords and each database record a profile. A profile may have less that m keywords.

Suppose a given profile at a given rank contains keywords $\{w_1, \ldots, w_m\}$. To generate the corresponding rank binary string (index) IPTV computes the l-bit HMAC h_i of each keyword w_i (HMAC: $\{0,1\}^* \rightarrow \{0,1\}^l$). Let

$$h_i = h_i^{r-1}, \ldots, h_i^1, h_i^0 \tag{4.1}$$

be the base 2^d representation of h_i. From h_i for each keyword w_i the IPTV computes a binary string (keyword trapdoor) I_i

$$I_i = (I_i^{r-1}, \ldots, I_i^j, \ldots, I_i^1, I_i^0), \tag{4.2}$$

where

$$I_i^j = \begin{cases} 0 & \text{if } h_i^j = 0 \\ 1 & \text{otherwise} \end{cases} \tag{4.3}$$

The index for the given profile at the given rank is computed as

$$I = \odot_{i=1}^{m} I_i. \tag{4.4}$$

where $\odot$ denotes bitwise production.

The IPTV sends to the cloud server records which have the form

$$(p_{id}, I^{seldom}, I^{average}, I^{frequent}),$$

where I^{rank} is the index of the viewer profile at rank $rank$ and p_{id} is an anonymized pseudonym for a viewer.

As an example, if the l-bit output of HMAC h_i is 101100001101001101110000; it can be represented by 6-digit number in base 2^4 that is h_i=11,0,13,3,7,0. According to (4.3), h_i is reduced to 6-bit binary string where each bit is assigned zero if $h_i^j = 0$ and one otherwise. The binary string of the given HMAC output will be 101110.

4.4.2 Query Generation and Oblivious Search on the Database

Data mining can be performed on the information available on the cloud server if an entity knows keyword trapdoors. Only the IPTV can compute those trapdoors as that computation is equivalent to producing HMAC whose key is available only to the IPTV. An advertiser can purchase from the IPTV any subset of keywords associated with database entries. A conjunctive query of keywords (i.e. $w_1, \ldots, w_n$) is the bitwise product of the corresponding trapdoors

$$I^q = \odot_{i=1}^{n} I_i = q_{r-1} \ldots q_0.$$

A conjunctive query from the advertiser to the cloud server is such r-bit binary sequence (no matter how many search terms exist in the query) and search is done only by r-bit comparisons. The index of the query is compared with the first level index of each profile. The indices of two other levels only be compared if the query matches with the first level index. The server response is a list of p_{id} such that each p_{id}'s index, $I_{p_{id}}$, matches the query I^q. We say that $I_{p_{id}} = j_{r-1} \ldots j_0$ matches $I^q = q_{r-1} \ldots q_0$ if

$$\forall i \in [0, \ldots, r-1], q_i = 0 \Rightarrow j_i = 0. \tag{4.5}$$

Responses are returned rank wise for the records, that is for seldom, average, and frequent ranks the cloud server returns separate anonymized id list.

Our proposed method supports conjunctive search formulas meaning that all the predicates in the query (predicate1 AND predicate2 AND ....) must exist in the profile to become an answer to that query. However, the advertiser is able to perform disjunctive queries by merging the responses of several conjunctive or single-word queries.

4.4.3 False Accept Rate

The indexing method that we employ covers all the information on keywords in a single r-bit index file. Independent from the hash function, after reduction and bitwise product operation there is a possibility that index of a query may wrongly match with an irrelevant profile which is called False Accept Rate (FAR for short). The system is free from false rejects, meaning if a profile contains all of the predicates in the query, that profile will definitely be a match to the query. The FAR is calculated as:

$$\frac{number\ \ of\ \ incorrect\ \ matches}{number\ \ of\ \ all\ \ matches}$$

We illustrate FAR with an example. Suppose that the zero bits in the index of the query Q_1, correspond to zero bits in the index of Q_2.
Q_1(30-40, F, Istanbul, News-primetime)
Q_2(50-60, M, Istanbul, Talkshow-primetime, Competition-primetime).
Therefore subscribers who are in the category of the Q_2 are wrongly included in the response of the Q_1. Those wrongly included subscribers are incorrect matches.

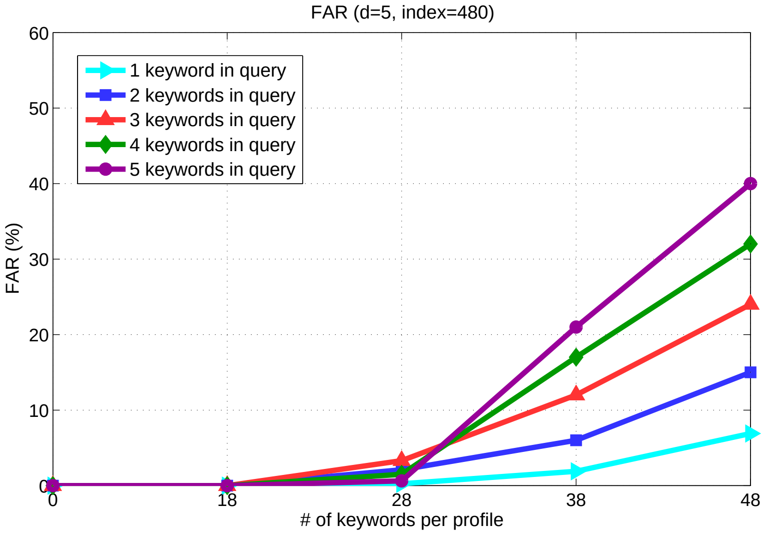

Figure 3: False Accept Rate($d = 5$, $index = 480$ bits)

Figure 3 compares false accept rate of queries containing two, three, four, and five predicates between the profiles containing $0, \ldots, 48$ keywords where index size (r) is 480 bits and d=5. The FAR results shown in the figures are computed by taking average of 100 random queries. The false accept rate increases rapidly after 38 keywords per profile due to the increase in the number of zeros in the index file. In our experiments the maximum number of keywords per profile is 18. To get a better intuition we extended our experiment to test the relationship between the parameters. If more keywords are required per profile, false accept rates can be reduced by increasing the index size (i.e. choosing a longer HMAC function) and choosing larger base 2^d. The false accept rates given in Figure 4 and 5 are calculated for 784-bit and 1040-bit index and d=6. We observe that the false accept rates decrease significantly in comparison to the rates in Figure 3. Having 1040-bit index and d=6 leads to the lowest false accept rate as illustrated in Figure 5.

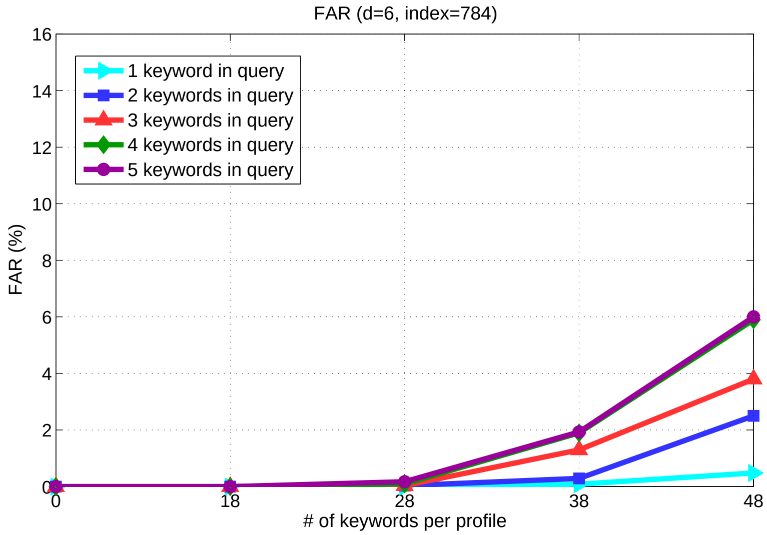

Figure 4: False Accept Rate($d = 6$, $index = 784$ bits)

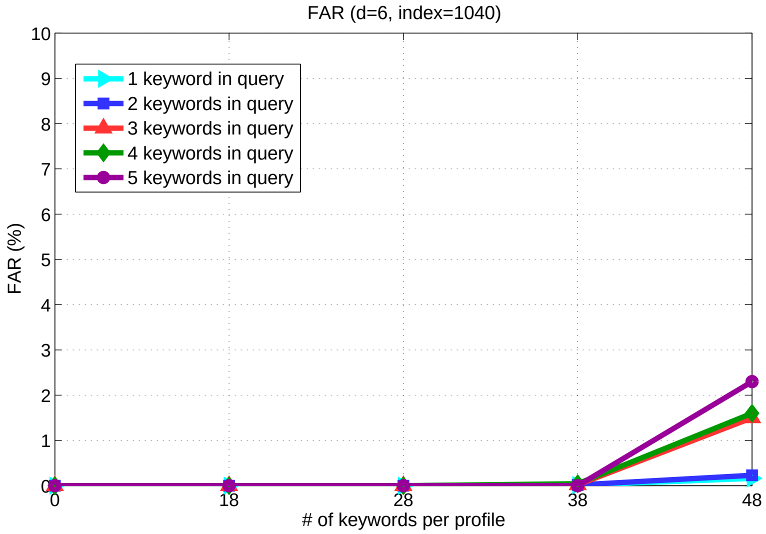

Figure 5: False Accept Rate (d=6, index=1040 bits)

The more keywords in a profile the more zeros in the index of the profile, therefore a higher chance of matching these profiles incorrectly. However, for the number of keywords in a query no conclusive results are obtained about the relationship between the number of keywords in the query and the false accept rate.

For example, if $Q_1 \subset Q_2$, then

$$result(Q_2) \subset result(Q_1) \Rightarrow FA(Q_2) \subset result(Q_1)$$

the falsely accepted (FA) profiles for Q_2 are the subset of Q_1 results. The FA profiles of Q_1 and Q_2 are not necessarily identical. There may be several profiles which are matched incorrectly with Q_1 index, while the zero bits in the indexes of these profiles do not match with the zero bits of Q_2 index. In addition, the FA profiles for Q_2 may be in the list of correct matches for Q_1. For example, suppose Q_2=(m, 20-30, married) and Q_1=(m, 20-30). Among the FA profiles for the Q_2 can be a profile with (m, 20-30, widow, Istanbul, sport-afternoon), which is a correct match for the Q_1. On the other hand, increasing the keywords in a query is likely to decrease the number of matches returned as query results. Consequently, having fewer number of incorrect matches may create greater FAR. In our system the possibility of false accept rate or false reject rate is zero due to relatively fewer number of keywords per profile.

Increasing the index size, which is useful to decreases the false accept rate, increase the computation cost and storage requirements. Table 11 in Section 6.3 shows the times it takes to generate indexes of the profiles for different values for r and d. There is a trade-off between computation cost and false accept rate. Optimized value for index size should be chosen considering the requirements of the applications where the proposed method is used.

4.4.4 Trapdoor Non-transferability Protocol

The trapdoors which are legally purchased by the advertisers are called authorized trapdoors. Since the trapdoors for the same purchased keywords are the same, it is possible for advertising agencies to sell their authorized trapdoors to each others and utilize unauthorized trapdoors in their queries. In this section we provide a protocol to prevent advertising agencies from utilizing unauthorized trapdoors in their queries.

Public keys of the participants: Assume that PU_X and PR_X are public key and the private key of the party X, respectively. Further assume that an advertiser (ADV) purchases the following trapdoors $(I_1, I_2, \ldots, I_n)$ corresponding to the keywords $(w_1, w_2, \ldots, w_n)$. The **IPTV** performs the following steps:

1. Computes $I = \odot_{i=1}^{n} I_i$.

2. Generates the signature for the trapdoors sold to ADV

   $S = SIGN_{PR_{IPTV}}(I, PU_{ADV})$.

3. Sends (S, I) to ADV.

The advertiser validates the signature S. For a query that involves the keywords $\{w_1, w_2, \ldots, w_t\}$, the advertiser and the cloud server execute the following protocol steps:

1. ADV computes $I^q = \odot_{i=1}^{t} I_i$.

2. ADV sends (I, S), I^q to CS.

3. CS continues if the signature is verified, otherwise it aborts the protocol.

4. For $I = j_{r-1} \dots j_0$ and $I^q = q_{r-1} \dots q_0$ CS checks

   $\forall i \in [0, \dots, r-1], q_i = 0 \Rightarrow j_i = 0$

   if I^q matches with I, meaning that ADV is authorized to ask such query, and CS continues, otherwise it aborts the protocol.

5. CS performs the query and generates the response R, which includes the list of ids matching the predicate in the query.

6. CS selects a symmetric key k and performs the following encryptions $ENC_{PU_{ADV}}(k)$ and $ENC_k(R)$. Next he sends the resulting ciphertexts to ADV.

7. ADV decrypts $ENC_{PU_{ADV}}(k)$ with his private key and obtains k. Knowing the symmetric key, the advertiser can obtain the response.

5 Privacy Arguments

In this section we show how the proposed scheme addresses the privacy requirements in our setting. We assume that the database schema (i.e. keywords and their values) is known information and that the adversary does not have statistical information about the database and does not know the mapping between the keywords and trapdoors.

5.1 Data and Query Privacy

We argue that the encrypted index does not leak useful information about subscribers profile. Our scheme substitute the hash function of Wang et al. [23] with a keyed HMAC function. Following [23, Theorem 1] a polynomially bounded adversary can only learn trivial information about the data entries from the index published on the cloud server.

The underlying HMAC function protects the users' profiles against unauthorized access. To recover the complete information given only an index value a polynomial time adversary $\mathcal{A}$ needs to perform an exhaustive search on all possible HMAC keys. This is true both for database and query indices. With the current technology for 128-bit random key no adversary can reconstruct the user profile.

5.2 Unlinkability

We argue that an advertiser has hardly any reason to send the same query twice to the same database. Given HMAC tags for two different messages (keywords) are not related an adversary can only learn that a set of keywords in a given query are the subset of the keywords in another query by comparing the position of zeros similar to Equation 4.5.

Updating the HMAC key for each novel data sent to the CS causes trapdoors and all the records in the cloud to be changed. Unless there is a weakness in HMAC key whereby, given two HMAC tags for the same messages under different keys are related, queries to different databases cannot be linked. Similarly the values contains in different databases cannot be associated with each others.

5.3 Privacy of the Non-transferability Protocol

The protocol in Section 4.4.4 prevents advertisers from utilizing unauthorized trapdoors in their queries. The index in IPTV signature allows the CS to identify if the keywords in a given query were indeed purchased by the advertiser. This prevents an advertiser from querying the database with the keyword the advertiser did not purchase. Furthermore the advertiser id under the IPTV signature allows the CS to identify which advertiser purchased the access to the database. The encryption CS uses prevents an advertiser A from selling keywords to advertiser B. Indeed if advertiser B purchases keywords from advertiser A rather than IPTV then either advertiser B has to reveal its query to advertiser A or advertiser A has to give its private key to advertiser B. In either case one of the advertisers may be leaking some non-trivial information about their advertising strategies via their queries to a competitor. Thus advertisers have no incentives to collude against IPTV.

Since the CS response is encrypted using the advertiser public key only the designated advertiser learns the information transferred. Therefore the strategies used by various advertisers are hidden from each other.

6 Performance Evaluation

In the following sections, we present complexity of the system, system configuration and discuss the timing results.

6.1 Complexity

In this section, we evaluate the complexity of the proposed technique. The communication and computational costs are analyzed separately. We used the following notation:

N - number of profiles.

m - the number of keywords in each profile for the given rank.

t - number of purchased keywords by an advertiser

k - number of keywords to search

β - number of profiles matched with the query

α - number of nominated ids for targeted advertisement

r - size of index

h - hash digest size

η - number of rank levels

6.1.1 Communication overhead

Data owner-Server communication: Data owner sends profile indices to server weekly, which is Nr bits or $Nr\eta$ bits in case η ranking is used.

Data owner-Advertiser communication:

- *Stage one*: Data owner sends the trapdoors for the purchased keywords to an advertiser. This communication is performed again if the advertiser purchases a new keyword or if the secret key of the data owner is changed.

- *Stage two*: Advertiser sends nominated ids and advertisement to the data owner. Therefore advertiser sends 32α bits to the data owner assuming each id is a 32-bit integer, plus the content of the advertisement.

Advertiser-Server communication: Advertiser sends a single r-bit query index to the server if the query is conjunctive. The server sends only anonymized ids, therefore transmitting 32β bits to the advertiser.

The communication costs are summarized in Table 7.

	Communication cost (bit)
Data owner-Server	Nr $Nr\eta$ if ranking is used
Data owner-Advertiser ***Advertiser-Data owner***	tr $32\alpha + advertisement$
Advertiser-Server ***Server-Advertiser***	r 32β

Table 7: Communication cost in the proposed system

When the trapdoor non-transferability protocol in Section 4.4.4 is used, the communication cost incurred by each party is explained below and illustrated in Table 8, the RSA modulus n is 1024 bits.

Data owner-Server communication. This part of the communication is not affected by the non-transferability protocol.

Data owner-Advertiser communication.

- *Stage one*: Data owner sends trapdoors of t purchased keywords (i.e. tr bits), an r-bit index computed by bitwise product of$t$ trapdoors, and a 1024-bit signature.
- *Stage two*: Advertiser sends nominated ids and advertisement to the data owner. Therefore advertiser sends 32α bits to the data owner

assuming each id is a 32-bit integer, plus the content of the advertisement.

Advertiser-Server communication. The advertiser sends the signature, r-bit trapdoor (computed by bitwise production of t trapdoors), and r-bit query index if the query is conjunctive. The server sends an RSA encryption of the symmetric key and symmetric encryption of the query's response to the advertiser.

	Communication cost (bit)
Data owner-Server	Nr $Nr\eta$ if ranking is used
Data owner-Advertiser	$tr + 1024 + r$
Advertiser-Data owner	$32\alpha + advertisement$
Advertiser-Server	$1024 + 2r$
Server-Advertiser	$1024 +$ response

Table 8: Communication costs in the proposed system using non-trasferability protocol

6.1.2 Computational Overhead

The following is the computational cost for each party in the system.

Data owner. Creates weekly profile indices.

Advertiser. Only prepare an index for a query which involves bitwise products of binary strings.

Server. The server performs search operation, which is basically r-bit binary comparisons with the database entries. If ranking is used, the server performs at most $\eta - 1$ additional binary comparison for each matching profile.

The computational costs are summarized in Table 9.

	Computational cost (bit)
Data owner	$\sum_{j=1}^{j=N}\sum_{i=1}^{i=\eta}(m \times h$-bit hash $+m$ bitwise product of r-bit binary string)
Advertiser	bitwise products of k indices $(k \times r)$
Server	***(Without ranking)*** $N \times r$ binary comparison ***(With ranking)*** $N \times r + (\eta - 1) \times \beta \times r$ binary comparisons

Table 9: Computational costs in the proposed system

For the case where the non-trasferability protocol is used, the computational cost is summarized in Table 10.

Data owner creates profile indices periodically (e.g. weekly). For each advertiser, the data owner computes r-bit trapdoors for the t purchased keywords and bitwise product of t trapdoors to obtain an r-bit index I. Data owner signs I and the public key of the advertiser together and sends the resulting signature to the advertiser.

Advertiser. Verifies the signature. To perform a conjunctive query of k keywords, the advertiser calculates bitwise product of k trapdoors. To get the response, the advertiser performs an RSA decryption to obtain a symmetric key followed by a symmetric decryption.

Server. The server verifies the signature and checks if the advertiser is authorized to ask the query by comparing the index of the query and I. To generate the response, CS performs r-bit binary comparisons with the database entries. If ranking is used, the server performs at most $\eta - 1$ additional binary comparisons for each matching profile. Finally CS selects a symmetric key which is encrypted with the public key of the advertiser and encrypts the response with the symmetric key.

	Computational cost (bit)
Data owner	$\sum_{j=1}^{j=N} \sum_{i=1}^{i=\eta}(m \times h$-bit hash $+m$ bitwise product of r-bit binary string) $t \times h$-bit hashes $+t$ bitwise products of r-bit trapdoors 1024-bit RSA signature
Advertiser	1024-bit RSA signature verification. $(k \times r)$ bitwise products of k trapdoors To get response: RSA decryption of the symmetric key and symmetric decryption of the response
Server	1024-bit RSA signature verification. r-bit binary comparisons ***(Without ranking)*** $N \times r$ binary comparisons ***(With ranking)*** $N \times r + (\eta - 1) \times \beta \times r$ binary comparisons RSA encryption of the symmetric key and symmetric encryption of the response

Table 10: Computational costs in the proposed system using non-trasferability protocol

6.2 System configuration

The proposed framework is developed in C++. The whole system is tested on a workstation with the following configuration:

- Widows 7 Professional (64-bit)
- Intel Xeon Core 6 Processor at 3.20 GHz
- 8 GB RAM.

Database specifications:

- custom made synthetic database
- 7 demographic features

- 11 watching habits
- number of profiles varies from 5,000 to 80,000.
- Index size is 480 bits.

Used algorithm specifications:

- HMAC using SHA-2 hash function. The output size of the hash function can be 224, 256, 384, or 512 bits. The block size for 224 and 256 bits output size is 64 bytes and 128 bytes for the others. The size of secret key is 128 bits. (for HMAC the codes in http://code.google.com/p/rainmeter are used)
- RSA based public key encryption and digital signature with 1024 bits modulus. (Miracl library is used for RSA operations)

6.3 Timing Results

In our first experiment the HMAC function produces 300 bytes output, which is generated by concatenating outputs of SHA2-based HMAC functions of different lengths. We choose $d = 5$, thus after reduction the index size (r) is 60 bytes (480 bits). In our experiments, the datasets have different number of subscribers varying from 5,000 to 80,000 with each profile having at most 18 keywords. Data owner timing results for index generation are presented in Figure 6. Since these operations are performed periodically (e.g. weekly) and index calculation can be parallelized, the presented technique is practical and efficient from data owner's perspective. Timing results for index calculation with parallelization are illustrated in 7, which are approximately five times less than the original case.

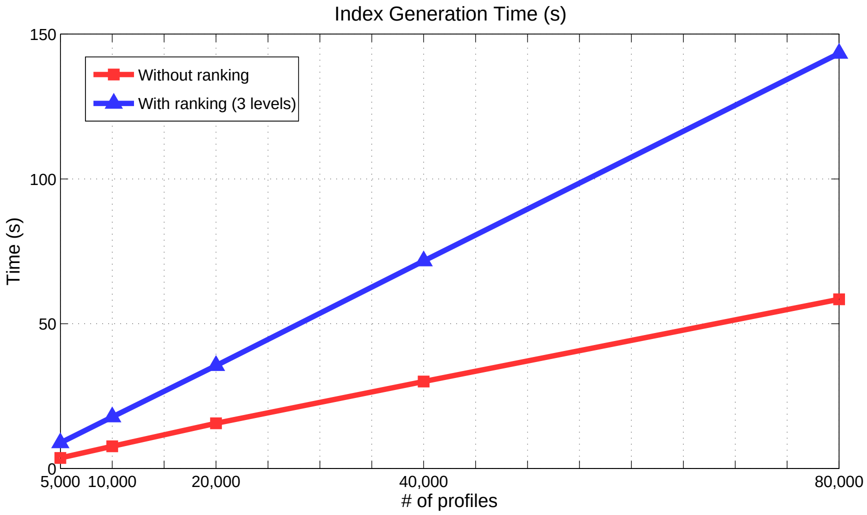

Figure 6: Timing for index generation (d=5, index=480 bits)

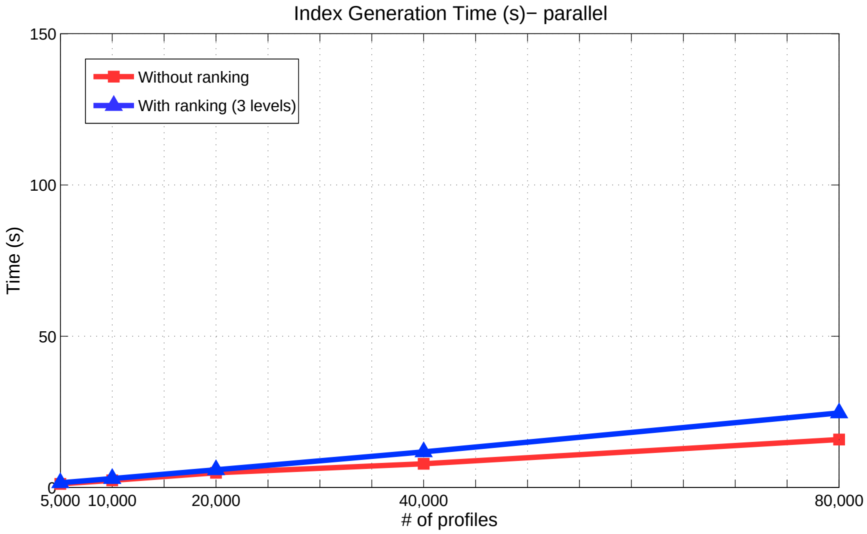

Figure 7: Timing for index generation with parallelization

Timings for search with and without ranking is relatively low as shown in Figure 8. Therefore, the scheme is efficient from cloud server's perspective as well The time to construct a single query by the advertiser is negligible and therefore the overall system is efficient.

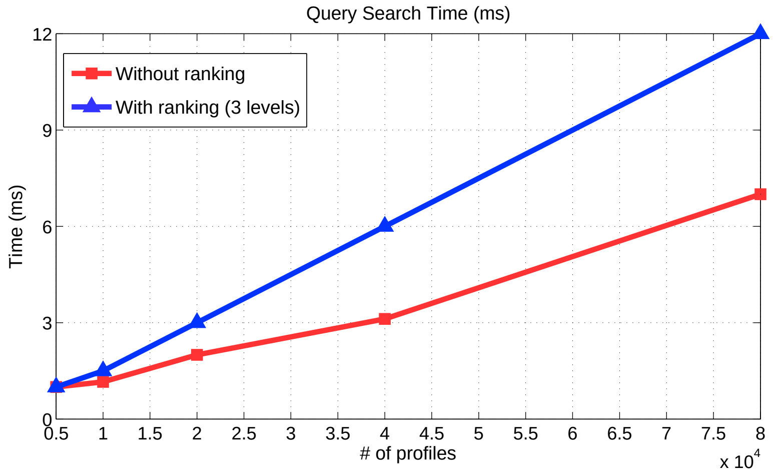

Figure 8: Timing for query search

When the non-transferability protocol presented in Section 4.4.4 is used the response time at cloud server side will increase due to signature verification and RSA encryption as illustrated in Figure 9.

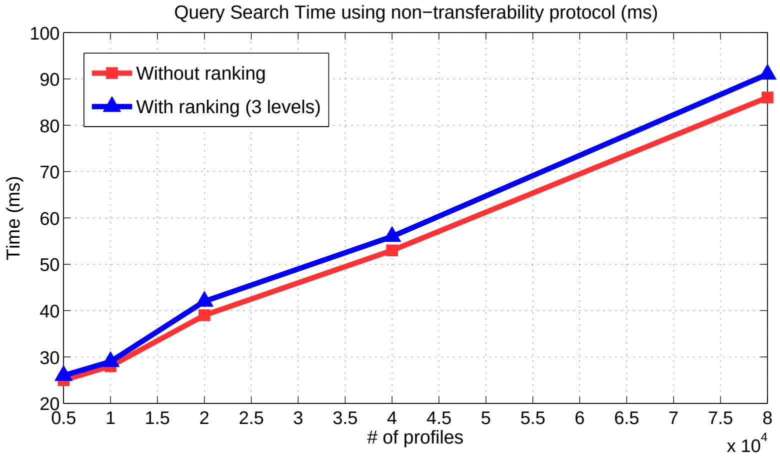

Figure 9: Timing for query search using the non-transferability protocol

Table 11 shows the effect of various d and r (index size) values on index calculation time at data owner side and storage requirement on the cloud server.

	# of profiles	***Index Generation Time (ms) without rank***	***storage (bit)***
d=4, index=344	*5000*	*2225*	5000×344
d=5, index=480	*5000*	*3613*	5000×480
d=6, index=656	*5000*	*6050*	5000×656
d=6, index=784	*5000*	*7122*	5000×784
d=6, index=1040	*5000*	*10468*	5000×1040

Table 11: Index generation time and storage requirement for various r and d values

7 Conclusions and Future Work

In this thesis, we propose a practical scheme for privacy-preserving targeted advertising service using cloud computing. Since the IPTV is not willing to deal with management and processing of subscribers data for advertising purposes, it prefers to outsource data into the cloud. IPTV uses an indexing method to encrypt subscriber profiles and provide trapdoors to advertising agencies to enable them sending queries to the cloud. Our experiments show that the proposed system is efficient from the perspective of each party. Construction of a single query by the advertiser takes negligible amount of time while a query search by the cloud server takes relatively low amount of time, which can be performed even by regular desktop computers. The index generation operations need to be performed periodically that can be parallelized so the technique is practical from data owner's perspective. The false accept rates are quite low showing that a very high percentage of the advertisements will reach to the right subscribers.

The system gives high amount of flexibility to advertisers to perform different queries on the database and use the results in their data mining algorithms to determine a set of subscribers that would be interested in their advertisements. There is no need for the IPTV to design any algorithm or develop software to send advertisements to related subscribers; thus it is relieved partially from management cost and processing of data. Data mining jobs are done by the advertisers who are professional in this subject and the cloud server is responsible to search the database and find the related subscribers. IPTV generates only indices of the profiles, which is performed weekly and sends the targeted advertisements to nominated subscribers.

Following the current research, we propose possible directions for future work. We plan to develop a system that allows to send complex queries. Our proposed method only supports conjunctive (AND) queries as explained in Section 4.4.2. We aim to support queries such as (predicate1 AND predicate2, NOT predicate3).

As a future work, we will enable IPTV to estimate the number of household in a house and their other characteristics such as age range by examining the watching habits during a week or consecutive weeks; this information is beneficial to find the best targets for an advertisement since one house is usually inhabited by several people with different watching habits.

We will test our proposed system on real data to examine its accuracy and efficiency. The privacy arguments will be studied in more details and proofs will be provided.

Also as future work, we will provide a technique to enable advertiser agencies to make sure that their advertisements are delivered to the right subscribers since they pay for this service.

Finally, the technique that encrypts the advertisement content, whereby only the targeted subscribers can see the content, will increase the privacy of both advertisers and subscribers.

APPENDIX - Notation

- P - collection of subscribers profiles, denoted as a set of n profiles $P = \{P_1, P_2, ..., P_n\}$
- Q- represents a query.
- I^q - the index of the query Q.
- R - the response of the query Q.
- h_i - the HMAC output for the keyword w_i.
- l - the output size of HMAC, represented by r digit in base 2^d. ($l = rd$)

References

[1] Architecture of iptv. `www.http://icontrol.in/blogs/2010/03/22/1269239682536.html`.

[2] Google adsense. `www.google.com/adsense`.

[3] Hmac. `www.http://en.wikipedia.org/wiki/HMAC`.

[4] Iptv. `www.http://en.wikipedia.org/wiki/Iptv`.

[5] *Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems).* Morgan Kaufmann, 1st edition, 2000.

[6] E. Aïmeur, G. Brassard, J. Fernandez, and Mani. Alambic : a privacy-preserving recommender system for electronic commerce. *International Journal of Information Security*, 7:307–334, 2008.

[7] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.

[8] L. Ballard, S. Kamara, and F. Monrose. Achieving efficient conjunctive keyword searches over encrypted data. *Springer*, 3873:414–426, 2005.

[9] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword searchs. *In proceedings of Eurocrypt 2004, LNCS 3027*, pages 506–522, 2004.

[10] D. Boneh and B. Waters. Conjunctive, Subset, and Range Queries on Encrypted Data. In *Theory of Cryptography*, volume 4392 of *Lecture Notes in Computer Science*, pages 535–554. Springer Berlin / Heidelberg, 2007.

[11] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou. Privacy-preserving multi-keyword ranked search over encrypted cloud data. In *IEEE INFOCOM*, 2011.

[12] D. el Diehn, I. Abou-Tair, I. Köster, and K. Höfke. Security and privacy requirements in interactive tv. *Multimedia Systems*, 17:393–408, 2011.

[13] S. Kamara and K. Lauter. Cryptographic cloud storage. *Workshop on Real-Life Cryptographic Protocols and Standardization 2010E*, pages 1–9, 2010.

[14] D. Kifer and J. Gehrke. l-Diversity: Privacy Beyond k-Anonymity. In *In ICDE*, 2006.

[15] M. Kodialam, T. Lakshman, S. Mukherjee, and L. Wang. Online scheduling of targeted advertisements for iptv. *INFOCOM, 2010 Proceedings IEE*, pages 1–9, 2010.

[16] R. Konow, W. Tan, L. Loyola, J. Pereira, and N. Baloian. Recommender system for contextual advertising in iptv scenarios. *Computer Supported Cooperative Work in Design (CSCWD), 2010 14th International Conference on*, pages 617–62, 2010.

[17] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE*, 7:76–80, 2003.

[18] W. H. Min and Y. G. Cheong. An interactive-content technique based approach to generating personalized advertisement for privacy protection. In *HCI (9)*, pages 185–191, 2009.

[19] K. Park, J. Choil, and D. Lee. Iptv-vod program recommendation system using single-scaled hybrid filtering. *ISCGAV'10 Proceedings of the 10th WSEAS international conference on Signal processing, computational geometry and artificial vision*, pages 128–133, 2010.

[20] E. Shen, E. Shi, and B. Waters. Predicate Privacy in Encryption Systems. In *TCC '09: Proceedings of the 6th Theory of Cryptography Conference on Theory of Cryptography*, pages 457–473. Springer-Verlag, 2009.

[21] L. Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10:557–570, 2002.

[22] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou. Secure ranked keyword search over encrypted cloud data. In *ICDCS'10*, pages 253–262, 2010.

[23] P. Wang, H. Wang, and J. Pieprzyk. An efficient scheme of common secure indices for conjunctive keyword-based retrieval on encrypted data. In K.-I. Chung, K. Sohn, and M. Yung, editors, *Information Security Applications*, volume 5379 of *Lecture Notes in Computer Science*, pages 145–159. Springer Berlin / Heidelberg, 2009.