PRIVACY-PRESERVING TARGETED ADVERTISING SCHEME FOR IPTV USING THE CLOUD

Leyli Javid Khayati[1], Erkay Savaş[1], Berkant Ustaoğlu[2] and Cengiz Örencik[1]
[1] *Faculty of Engineering & Natural Sciences, Sabancı University, Istanbul, 34956 Turkey*
[2] *Faculty of Science, Izmir Institute of Technology, Izmir, 35430 Turkey*
{leyli, erkays, cengizo}@sabanciuniv.edu, bustaoglu@uwaterloo.ca

Keywords: IPTV: Targeted Advertising: Privacy: Cryptography: Cloud Server.

Abstract: In this paper, 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 rate, which are published on the cloud server periodically (e.g. weekly) along with anonymized demographics. 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.

1 INTRODUCTION

Literature suggests (Kodialam et al., 2010) 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 claimed (Min and Cheong, 2009) 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 advertisemnets selcted on the basis of online traits and demographics of individuals are preferred by advertising agencies aiming to increase the benefit from advertisement. However, a major issue is the potential violation of the privacy of individuals.

It appears as if IPTV is becoming a preferred online media with potentially millions of subscribers. Thus, IPTV is a hot spot for advertising agencies that have the incentives to utilize the data that IPTV collects about its subscribers. 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 (Armbrust et al., 2009).

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 (Kamara and Lauter, 2010). However, by encrypting data, one of the key functionalities of database systems, i.e. keyword-based search operation, becomes a challenging issue. Remote querying of encrypted databases on external servers is proposed by (Ceselli et al., 2005). Their approach is based on the use of indexing information which is attached to the encrypted database. Many searchable encryption schemes such as (Boneh et al., 2004), (Wang et al., 2010) 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 an ineffective solution.

In our setting the advertiser performs queries on a remote database of subscribers data (outsourced to cloud server by IPTV) to find records matching with the keywords in its queries. Using the responses to its queries as input to its private strategy, the advertiser's goal to match viewers with relevant advertisements. Keywords (conjunctive and otherwise) in queries are needed to be encrypted to prevent the disclosure of private strategy of advertiser 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 paper is to facilitate this efficiently in the context of a practical privacy-preserving scheme, whereby authorized advertisers can send personalized advertisements to subscribers.

The proposed scheme is different from the previous works on the same subject, in which the 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 content and flow, but may analyze them to infer additional information); consequently the privacy of users must be protected. The benefits of the proposed scheme are multifold: i) the IPTV is partially relieved of management cost and processing of data, ii) advertiser's mining techniques are not exposed to the cloud server, iii) data mining required for targeted advertising can be performed by advertising agencies, which have the relevant expertise and tools, and iv) the IPTV can generate additional revenue from subscribers' data by ensuring sufficient data-protection safeguards in comply with relevant legislations[1]. The scheme is also useful when the advertiser is a division within the IPTV (e.g., private cloud). In this case, the IPTV has robust data protection practices in accordance with the relevant legislations. That is, the 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.

2 RELATED WORK

As mentioned in the previous section, 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 fast and efficient multi-keyword queries[2]. Many searchable encryption schemes focus on a single keyword search. Wang et al. (Wang et al., 2010) 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. (Boneh et al., 2004) present the first searchable encryption construction using public key cryptosystem to perform search on encrypted data. Several works on multi-keyword search were proposed (Cao et al., 2011), (Boneh and Waters, 2007), (Ballard et al., 2005), (Shen et al., 2009) 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. (Wang et al., 2009). A searchable index is generated for each document, which contains all the keywords in the document. They use cryptographic hash function to obtain indexes and trapdoors that allow secure searching for keywords in indexes.

In our solution we adopt the scheme by Wang et al. (Wang et al., 2009) utilizing a keyed hash function (HMAC) to map keywords in a subscriber's data to a sequence of r-bit index using a secret key known only to the data owner. A similar approach is used in (Örencik and Savaş, 2012) for multi-keyword search on encrypted data. Advertisers must, in advance, obtain so-called *secure trapdoors* from the data owner which enable to search for corresponding keywords in subscribers data. Since those trapdoors are generated using the data owner's secret key, the server is not able to learn any information about the keywords 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. The proposed solution addresses the relevant security

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

[2] Multi-keyword (or multi-predicate) queries are essential to find database entries that are relevant to all keywords in the query.

and privacy requirements of interactive TV (el Diehn et al., 2011) which are also applicable in our scenario. In the following section we provide the motivation for the adopted model.

3 MOTIVATION

Since we aim to place subscribers' data on a cloud we need to anonymize it. One particular anonymization technique would cluster the subscribers having sufficiently close demographics and watching traits. Then cluster representatives would be placed in the cloud, whereby the advertisers utilized this summary data to match the relevant subscribers with their advertisement portfolio. This technique is similar to the *k*-anonymity (Sweeney, 2002) algorithm, but not the best in our application scenario for the following reasons: i) clustering leaks the information of some subscribers, ii) the advertisers have to use static clusters formed by the IPTV, iii) IPTV cannot control who access/utilize the data, and iv) loss of accuracy. Since leaking subscribers information is the most damaging problem from privacy perspective, more robust solutions are needed. We propose one such solution in the remainder of this work.

4 THE PROBLEM STATEMENT

The goal is to send targeted advertisements to viewers. We begin by describing the entities in our system and their goal and knowledge.

4.1 Entities

There are three entities in our system:
Data owner.[3] 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.
Advertiser (ADV). Advertisers are entities that generate targeted advertisements based on viewer demographics and watching habits. Advertisements are generated based on a private advertiser strategy (e.g., data mining rules/techniques) 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 the IPTV. Since mining rules can reveal information about advertiser strategy, the advertiser wants to keep those rules secret.
Server. 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 Framework of Interaction

The general framework is illustrated in Figure 1. The 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 external entity, 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 useful mining queries without IPTV assistance and permission. Any prior IPTV assistance should be useful only to the designated entities. Furthermore, designated entities (advertisers) should not be able to combine responses to queries to infer any information that identifies individuals or link them to another

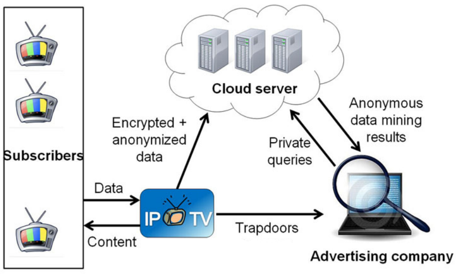

Figure 1: General framework of directed advertising service.

[3] In fact, the regulations state that the owner of an entry is the person whom the entry is about. We use the term data owner for IPTV in the sense it is the owner of the database.

database. This necessitates applying anonymization and generalization to the database before encryption.

The advertiser purchases access to the 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; the trapdoor information should not be 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 the 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.

5 CONSTRUCTION

In this section, we provide details of our solution.

5.1 Data Model

We consider a scenario where the profile of each subscriber consists of seven demographic features and eleven watching preferences without loss of generality. The demographic features are age, gender, marital status, education, occupation, city, and location, whose values are the keywords in our scheme. Morning watching preferences are marriage, news, and health programs. Similarly afternoon watching preferences are marriage, TV series, and sport programs, while prime-time watching preferences are news, competition, series, talk show, and sport programs[4]. Each watching preference is also taken as a keyword in the proposed solution.

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 real number in $[0,1]$ 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, i.e., how much the retrieved profiles are relevant to the query, we use the relation described in Table 1. For example the value 0.5 that appears in a viewer profile under prime time news implies that the viewer under consideration is frequently watching such program.

Table 1: Rate table for watching habit features.

Rate of watch	**Rank**	**Level**
0	not watched	0
> 0	Seldom	1
$>= 0.15$	Average	2
$>= 0.30$	Frequent	3

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. 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.

5.2 Index Generation

The actual solution is based on the construction proposed by Wang et al. (Wang et al., 2009). The main idea is to represent each database record as a binary string referred as *index*. To accommodate n-ranked search each record is expanded to n indexes. Without loss of generality following Table 1 there are three ranks "seldom", "average", and "frequent" according to the rate given to each program per week.

Subsequently index for each database record is cloned three times, once for each level. Each 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 1. We will describe how to generate the indexes corresponding to rank one, others are generated in an analogous way. Firstly, the IPTV selects an 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 $\{w_1, \ldots, w_n\}$, where $n \leq m$. To generate the corresponding rank binary string (index) the IPTV computes the l-bit HMAC h_i of each keyword w_i

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

(HMAC: $\{0,1\}^* \to \{0,1\}^l$). Let

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

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

$$I_i = (I_i^{r-1}, \ldots, I_i^j, \ldots, I_i^1, I_i^0), \tag{2}$$

where

$$I_i^j = \begin{cases} 0 & \text{if } h_i^j = 0 \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

the index for the given profile at the given rank is

$$I = \odot_{i=1}^n I_i, \tag{4}$$

where $\odot$ denotes bitwise product.

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.

5.3 Query Index Generation and Oblivious Search on the Database

An advertiser can purchase from the IPTV any subset of keywords associated with database entries and corresponding trapdoors. In the selected index generation scheme, multi-keyword (a.k.a. conjunctive keywords) queries can be efficiently constructed, resulting in an r-bit binary sequence independent of the number of keywords in the query. A multi-keyword query index of keywords, $w_1, \ldots, w_\delta$, is the bitwise product of the corresponding trapdoors

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

A query from the advertiser to the cloud server is such r-bit binary sequence known as query index and search is done only by r-bit comparisons. The server response is a list of p_{id} such that each p_{id}'s field in the response I matches the query index I^q. We say that $I = j_{r-1} \ldots j_0$ matches $I^q = q_{r-1} \ldots q_0$ if

$$\forall i \in [0, \ldots, r-1], q_i = 0 \Rightarrow j_i = 0. \tag{5}$$

In case of disjunctive query separate response is generated for each keyword in the query, which needs to be combined with others by appropriate operations such as union, subtraction, etc.

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

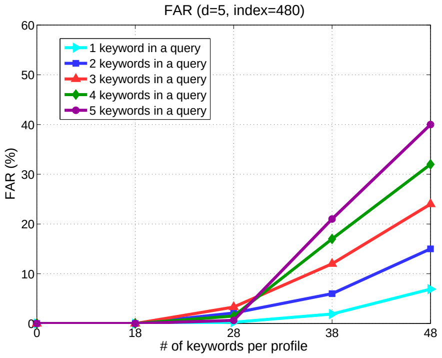

Figure 2: False Accept Rates(d=5, index=448 bits).

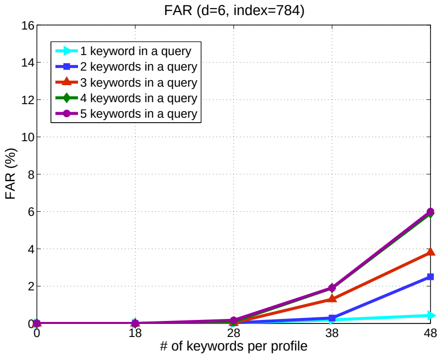

Figure 3: False Accept Rates(d=6, index=784 bits).

To prevent an advertiser from selling its trapdoors to other parties (or using other parties trapdoors in its queries), trapdoor non-transferability protocol, given in Appendix is designed. This protocol stipulates that the advertiser submit a proof of ownership for the trapdoors in its queries. Although relatively expensive, the signature verification part of this protocol, which dominates its timing, is executed once after the purchase of trapdoors.

5.4 False Accept Rates

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 keywords in the query, that profile will definitely be a match to the query. The FAR is calculated as:

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

In Figure 2 we compare the false accept rates of queries containing two, three, four, and five keywords

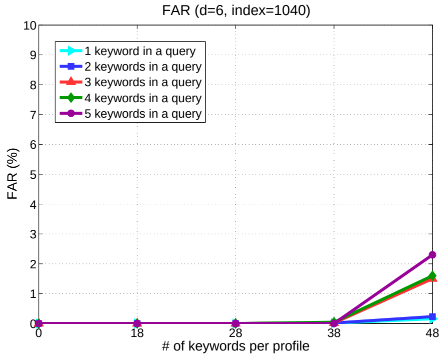

Figure 4: False Accept Rates(d=6, index=1040 bits).

for the profiles having 18, 28,. . . , 48 keywords where index size (r) is 480 bits. The false accept rates increase rapidly after 38 keywords per profile since the number of zeros in the index file increase.

To obtain a better intuition we extend 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 3 and 4 are calculated for 784-bit and 1040-bit index lengths when $d = 6$. We observe that the false accept rates decrease significantly in comparison to the rates in Figure 2.

Having 1040-bit index and $d = 6$ leads to the lowest false accept rate as illustrated in Figure 4. More keywords in profiles leads to more zeros in the index for profiles, therefore a higher chance of matching these profiles incorrectly. We observe a similar trend for the multi-keyword queries, where more trapdoors in a query tend to increase the false accept rates. In any case, acceptably low false accept rates can always be achieved by increasing the index length and using larger base 2^d.

In our system the possibility of false accept and false reject rates are zero due to the number of keywords per profile is low.

6 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 know the mapping between the keywords and trapdoors. We further assume that the statistical model of the database is initially known only by the data owner.

6.1 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. (Wang et al., 2009) with a keyed HMAC function. Following (Wang et al., 2009, 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 indexes. With the current technology for 128-bit random key no adversary can reconstruct the user profile.

6.2 Unlinkability

A privacy-preserving targeted advertising scheme is said to provide unlinkability for queries if it prevents an adversary from linking two queries consisting of identical set of keywords.

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) that 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 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 outputs for the same messages under different keys are related, queries to different databases cannot be linked. Similarly the values contained in different databases cannot be associated with each others.

6.3 Preventing Unauthorized Access

The trapdoor non-transferability protocol in Appendix prevents advertisers 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 s/he did not purchase. Furthermore, the advertiser id in the IPTV signature allows the CS to identify which advertiser purchased the access to the database. The encryption the CS uses

Table 2: Notations.

N	number of profiles
n	number of keywords in each profile for the given rank
t	number of purchased keywords by an advertiser
δ	number of keywords in a (multi-keyword) query
β	number of profiles matched with the query
α	number of nominated ids for a specific advertisement
r	size of an index
h	hash digest length
η	number of rank levels
l	hash digest length

prevents an advertiser A from selling keywords to another advertiser B. Indeed if advertiser B purchases keywords from the advertiser A rather than the 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 the IPTV.

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

7 COMPLEXITY

In this section, we evaluate the complexity of the proposed technique. The communication and computational costs are analyzed separately. We give the adopted notation in the Table 2. The parameter values should be selected based on real data, which at the present is not available to us. The performance evaluation presented here only serves to provide intuition for the possible incurred cost.

7.1 Communication Costs

Communication costs are given on basis of interactions between the participants in Table 3. Values inside the brackets should be taken into account when the trapdoor non-transferability protocol in Appendix is used where the RSA modulus is 1024 bits.

- **Data Owner-Server communication.** The data owner sends profile indexes to the server periodically (e.g. weekly), which is Nr bits or $Nr\eta$ bits in case η levels are used for ranking.
- **Data Owner-Advertiser communication.**

  The data owner sends the trapdoors for the purchased keywords to the advertiser. This communication is performed again if the advertiser purchases a new keyword or if the secret key of the data owner is changed.

  The advertiser sends nominated ids and advertisement to the data owner. Therefore, the advertiser sends 32α bits to the data owner assuming that each id is a 32-bit integer, plus the content of the advertisement.
- **Advertiser-Server communication.** The advertiser sends an r-bit query index to the server. The server sends only an anonymized id list as a response to the advertiser. When ranking is used η lists are sent.

Table 3: Communication costs incurred by each party in the proposed system.

	Communication cost (bit)
Data owner-server	Nr
	$Nr\eta$ (η ranking)
Data owner-advertiser	$tr+$[1024+r]
Advertiser-data owner	$32\alpha + \mathit{advertisement}$
Advertiser-server	$r + [1024 + r]$
Server-advertiser	response+[1024]

7.2 Computational costs

Following are the computational costs for each party in the system.

- **Data Owner.** Creates profile indexes periodically.
- **Advertiser.** Only prepares an index for a query which involves bitwise product of trapdoors corresponding to keywords in a conjunctive query.
- **Server.** Performs search operation, which is basically r-bit binary comparisons of query indexes with the database entries. If ranking is used, the server performs at most η-1 additional binary comparisons for each matching profile.

Computational costs are summarized in Table 4. Values inside the brackets should be taken into account when the trapdoor non-transferability protocol in Appendix is used where the RSA modulus is 1024 bits..

7.3 Timing Results for Simple Multi-keyword Queries

In this section, we perform experiments for the scenario where the advertiser sends multi-keyword queries to find subscribers of matching profiles for an advertisement.

The proposed system is implemented and tested on a Windows 7 machine with Intel Xeon 6 Core running at 3.2 GHz. Our experiments are carried on

Table 4: Computational costs incurred by each party in the proposed system.

	Computational costs (bit)
Data Owner	$\sum_{j=1}^{j=N}\sum_{i=1}^{i=\eta}(n \times h$-bit hash$+n$ bitwise product of r-bit string). [for each advertiser: $t\times(h$-bit hash)+t bitwise products of r-bit trapdoors. 1024-bit RSA signature.]
Advertiser	r bitwise products of δ trapdoors (conjunctive query) [1024-bit RSA signature verification. To get response: RSA decryption of the symmetric key and symmetric decryption of the response.]
Server	[1024-bit RSA signature verification. r binary comparisons.] (without ranking) $N \times r$ binary comparisons. (with ranking) $N \times r + (\eta - 1) \times \beta \times r$ binary comparisons. One RSA Encryption of responses.

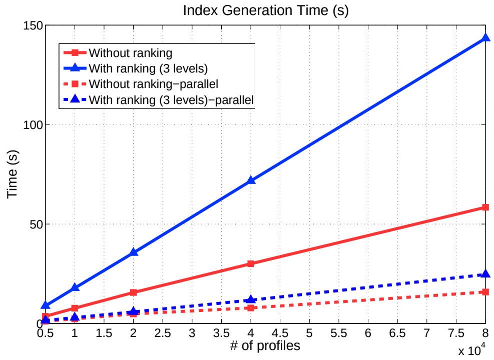

Figure 5: Timings for index construction with 18 keywords per profile (on data owner side).

custom-made synthetic database. In the experiments, the HMAC function produces 300 bytes output, which is generated by concatenating different length SHA2-based HMAC functions. We choose $d = 5$, so after reduction the index size (r) is 60 bytes (480 bits). The database has different number of subscribers varying from $5,000$ to $80,000$ with each profile having at most 18 keywords.

Timing results for operations on data owner side are presented in Figure 5. Since these operations are performed periodically (e.g. weekly) and index calculation can be parallelized, the presented technique is practical and efficient from the data owner perspective. Timing results for index generation with parallelization utilizing all six-cores in the platform are illustrated by dashed lines in Figure 5.

Timing results on the server side for the search operation with and without ranking indicate relatively low latency as shown in Figure 6. Timings for query search when using the trapdoor non-transferability protocol are also illustrated by dashed lines in Figure 6, which is again relatively low. Note that the signature verification in the protocol, which dominates the timing, does not have to be performed every time. Therefore, the scheme is efficient from cloud server perspective. Query construct by the advertiser takes negligible amount of time and therefore omitted here.

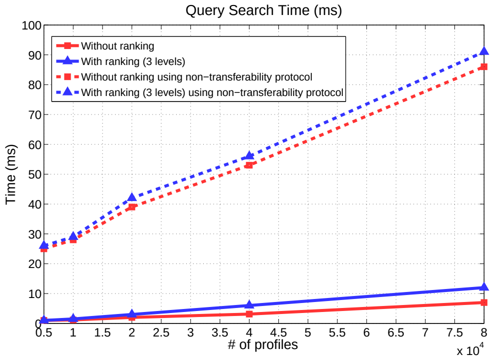

Figure 6: Timings for query search with and without ranking (on the cloud server side).

7.4 Timing Results for Data Mining Algorithms

We performed two experiments to demonstrate the efficiency of the two most common data mining algorithms in our setting, namely clustering and content-based filtering. The same implementation on the same platform as in Section 7.3 are used in these experiments.

7.4.1 Experiment 1: Clustering

Advertising agencies can use demographic filtering (Aïmeur et al., 2008) to categorize subscribers based on their demographics and send advertisements accordingly. Since there are seven demographic features in our data model and total number of keywords pertaining to the demographic features is 42 the advertising agency sends a total of 42 different (single-keyword) queries to obtain information about viewers' demographics. Then the agency can utilize the results of queries to generate a relevant viewers' segmentation for his advertising purpose. This experiment is implemented using reliable socket communication (i.e. TCP) in the same computer.

The timing results are enumerated in Table 5 for different number of subscribers, where the second column lists the total elapsed time for the advertiser while the third column indicates the time cloud server spends during the entire process. The last column shows the time spent on the communication. Note

Table 5: Timing results for obtaining the entire demographic data used in clustering based on demographics.

Number of profiles	Advertiser total time (ms)	Cloud Server - Query search time (ms)	Communication time (ms)
5,000	164	58	106
10,000	422	112	310
20,000	656	219	437
40,000	1,345	451	894
80,000	2.639	1,185	1,454

that the time spent for actual clustering algorithm is not reported here.

7.4.2 Experiment 2: Content-Based Filtering

The advertising agency can use content-based filtering (Aïmeur et al., 2008) to send its advertisements to related subscribers. Each advertisement can be described by some attributes or characteristics. Advertiser uses a similarity metric between the advertisement attributes and viewers demographics and watching habits to predict viewers' potential interests in a specific advertisement. The similarity $\mathcal{S}$ can be defined as a number in the interval $[0,1]$. If the advertisement c has non-zero similarity to viewers demographics and watching habits $(a_1, a_2, \ldots, a_\sigma)$, i.e. $\mathcal{S}(a_i, c) \neq 0$, then the prediction for the interest of a subscriber s on advertisement c is computed as

$$P_{c,s} = \frac{\sum_{i=1}^{\sigma} v_{s,a_i} \cdot \mathcal{S}(a_i, c)}{\sum_{i=1}^{\sigma} \mathcal{S}(a_i, c)}, \tag{6}$$

where v_{s,a_i} is the relevancy score of the subscriber s to the query containing attribute a_i. If the subscriber is not a match for a query, then his score will be zero.

As an example, suppose that an advertisement for a shaving blade brand will be sent to viewers who are male, leading active life styles and interested in sports. We can define non-zero similarities of the advertisement to the following demographics and watching habits: (*male, news-morning, news-primetime, sport-afternoon, sport-primetime*). The advertiser can send five different queries to the cloud server and calculate a prediction for matching subscribers using the rates showing the relevancy scores of the subscribers (i.e. v_{s,a_i}) to the query. The advertiser can specify a threshold τ for the advertisement, whereby a subscriber will become a nominate (target) for the advertisement if the prediction for the subscriber is greater than τ. Timings for the advertiser to compute predictions for the advertisement described by five attributes are 15 ms, 40 ms, 207 ms, 599 ms, 2,249 ms for number of profiles 5000, 10000, 20000, 40000, 80000, respectively.

8 CONCLUSION AND FUTURE WORK

We propose a practical solution for secure and privacy-preserving targeted advertising service for IPTV subscribers utilizing cloud computing. While the privacy of subscribers is protected, the advertiser agencies can target their advertisements to prospective customers with high precision since the false accept and reject rates can be made practically zero. Also by utilizing the rating mechanism adopted in our solution, the advertiser can select only viewers who demonstrate desired relevance to its queries. Furthermore, advertising agencies have another incentive to use the proposed system due to the fact that their private strategies in reaching potential customers are not exposed to cloud server and other advertisers.

From the IPTV's perspective, the proposed system has many advantages. It allows outsourcing all the services pertaining to advertisement without any adverse effect on security and privacy issues concerning its subscribers. Furthermore, advertisers (or any other party who has access to the database on the cloud) can utilize the data only if they are authorized to do so. The IPTV can control the access to the database in fine grained manner in the sense that the advertiser can only access to the part of the database related to keywords, for which it holds trapdoors.

The proposed method supports efficient conjunctive queries and single-keyword queries. As future work, we will develop an efficient system that accelerates complex queries, which combine many keywords in different forms such as (*keyword*$_1$ **OR** *keyword*$_2$ **NOT** *keyword*$_3$). Also, the proposed system considers the entire household as single individual while, in reality it usually consists of individuals with different profiles (e.g. children, young individuals, adults). A future system should represent a household with more than one profile to send relevant advertisements at the right time, which is necessary to find the best targets for an advertisement.

We will strengthen the security and privacy of the proposed system to prevent leakage for the applications where cloud server has some knowledge about the statistical model of the database. This is especially important where the responses to given queries conform to the statistical model.

As a first step before extending this work we plan to introduce formal definitions and arguments that justifies the soundness of our solutions.

ACKNOWLEDGEMENTS

This work was supported by Turk Telekom under Grant Number 3014-02.

REFERENCES

Aïmeur, E., Brassard, G., Fernandez, J., and Mani (2008). Alambic : a privacy-preserving recommender system for electronic commerce. *International Journal of Information Security*, 7(5):307–334.

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

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

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

Boneh, D. and Waters, B. (2007). 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.

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

Ceselli, A., Damiani, E., Vimercati, S. D. C. D., Jajodia, S., Paraboschi, S., and Samarati, P. (2005). Modeling and assessing inference exposure in encrypted databases. *ACM Trans. Inf. Syst. Secur.*, 8:119–152.

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

Kamara, S. and Lauter, K. (2010). Cryptographic cloud storage. In *Proceedings of the 14th international conference on Financial cryptography and data security*, FC'10, pages 136–149. Springer-Verlag.

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

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

Örencik, C. and Savaş, E. (2012). Efficient and secure ranked multi-keyword search on encrypted cloud data. to appear.

Shen, E., Shi, E., and Waters, B. (2009). 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.

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

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

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

APPENDIX

Trapdoor Non-transferability Protocol

This protocol is designed to prevent authorized advertisers selling their trapdoors to each other. Public keys of the participants:

PU_X- public key of the party X

PR_X- private key of the party X

Assume that an advertiser (ADV) purchases the following trapdoors $(I_1, I_2, \ldots, I_t)$ corresponding to the keywords $(w_1, w_2, \ldots, w_t)$

IPTV performs the following steps:

1. Computes $I = \odot_{i=1}^{t} I_i$.
2. Generates the signature for the sold trapdoors $S = SIGN_{PR_{IPTV}}(I, PU_{ADV})$.
3. Sends (S, I) to ADV.

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

1. ADV compute $I^q = \odot_{i=1}^{\delta} I_i$.
2. ADV sends (I, S), and I^q to CS.
3. CS continues if the signature is verified.
4. For $I = j_{r-1} \ldots j_0$ and $I^q = q_{r-1} \ldots q_0$ CS checks $\forall i \in [0, \ldots, r-1], q_i = 0 \Rightarrow j_i = 0$.

   If I^q matches with I, meaning that ADV is authorized to ask such query, otherwise it aborts.
5. CS performs the query and generates a response R, which includes the list of ids matching the predicates 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 ciphertexts to ADV.
7. ADV decrypts $ENC_{PU_{ADV}}(k)$ with its private key and obtains k. Knowing the symmetric key, the advertiser can obtain the response.