Information Sciences 180 (2010) 1060–1072

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# Efficient broadcast encryption with user profiles

Murat Ak *, Kamer Kaya [1], Kaan Onarlıoğlu, Ali Aydın Selçuk

*Department of Computer Engineering, Bilkent University, Ankara, 06800, Turkey*

ARTICLE INFO

*Article history:*
Received 16 October 2008
Received in revised form 11 November 2009
Accepted 16 November 2009

*Keywords:*
Broadcast encryption
CS scheme
SD scheme
User profiles

ABSTRACT

Broadcast encryption (BE) deals with secure transmission of a message to a group of users such that only an authorized subset of users can decrypt the message. Some of the most effective BE schemes in the literature are the tree-based schemes of complete subtree (CS) and subset difference (SD). The key distribution trees in these schemes are traditionally constructed without considering user preferences. In fact these schemes can be made significantly more efficient when user profiles are taken into account. In this paper, we consider this problem and study how to construct the CS and SD trees more efficiently according to user profiles. We first analyze the relationship between the transmission cost and the user profile distribution and prove a number of key results in this aspect. Then we propose several optimization algorithms which can reduce the bandwidth requirement of the CS and SD schemes significantly. This reduction becomes even more significant when a number of free riders can be allowed in the system.

## 1. Introduction

Broadcast encryption (BE) enables secure transmission of data to a large set of users such that only an authorized subset can decrypt it. It has a wide range of applications including pay-TV, content protection, secure audio streaming and Internet multicasting.

The users of a BE system are given a set of pre-installed, long-term keys, typically in a set-top box. These keys are later used to encrypt the broadcast sessions such that only the authorized user set, i.e., the users with the appropriate long-term keys, can decrypt the broadcast. The users who are authorized to receive a particular broadcast are called *privileged* (or *subscriber*) whereas the remaining non-authorized users are called *revoked* (or *non-subscriber*). In certain cases, a number of non-subscribers can be allowed to decrypt the broadcast in order to reduce the overall cost of the system. Such users are called *free riders*.

The particular design of a BE system varies according to the system characteristics, such as the size of the user domain, required security level, available bandwidth, and hardware capabilities. In the traditional setting, the amount of long-term storage is very limited as it has to be tamper resistant, the communication channel is one way, and the devices are stateless in the sense that no additional long-term storage is possible.

Two important performance parameters in evaluating a BE system are the key storage and transmission overheads incurred. The complete subtree (CS) and subset difference (SD) schemes of Naor et al. [20] are among the most well-known BE schemes today. Some of the theoretically most efficient BE schemes are obtained by the SD scheme and its variants [13,12]. The SD scheme has recently gained popularity in applications as well and is included in the next-generation DVD standard [1].

* Corresponding author. Tel.: +90 312 290 1350; fax: +90 312 266 4047.
*E-mail address:* muratak@cs.bilkent.edu.tr (M. Ak).

[1] Current Address: CERFACS, 42 avenue Gaspard Coriolis, Toulouse 31057, France.

0020-0255/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2009.11.024

Despite recent advances in the technology, such as the availability of two-way communication channels, have reduced the pay-per-view TV systems' reliance on BE schemes, new application areas have emerged that greatly benefit from BE, such as content protection [18,24], multicasting promotional material and low cost pay-per-view events [2], multi-certificate revocation/validation [3] and dynamic group key management [25,26,6,7,19].

User profiling is the concept of monitoring data on preferences and interests of the users in the system in order to serve them more effectively. It is broadly used in various areas such as web mining [16] and broadcasting and multicasting [9,15,17].

In the BE literature, traditionally, the users are assumed to be identical in the sense that they are taken to be equally likely to be interested in any particular broadcast. However, in practice every user has a certain type of interest, some being more interested in sport events, some in movies, some in entertainment, etc. If these user profiles are taken into account, they can provide some critical information to optimize the operations of a BE system.

In this paper, we study the problem of achieving a more efficient BE system in the presence of provided user preference information. Our approach works by constructing the subset structure of a CS or SD system according to the given set of subscriber profiles. We first analyze the relationship between the transmission overhead of a BE scheme and the distribution of the user profiles. After proving several key results, we give two optimal algorithms for the CS scheme with one broadcast type. Then we generalize our approach by proposing a similarity metric for the CS and SD schemes with multiple broadcast types. Theoretical and experimental results show that the approach can significantly reduce the transmission overhead of the CS-based and SD-based BE schemes. This reduction can especially be remarkable when the proposed approach is used in conjunction with an optimal free rider assignment [4,22].

The rest of the paper is organized as follows: After summarizing the related work in Section 2, we give an overview of the CS and SD schemes in Section 3. We analyze the average transmission cost of the CS and SD trees according to the user profiles in Section 4 and we prove several results on the optimality conditions in Section 5. We present our optimization algorithms in Section 6 and present the experimental results in Section 7. We discuss the application of user profiling with free riders and present further experimental results for various free rider assignments in Section 8. Section 9 concludes the paper.

## 2. Background

After Berkovits [5] introduced the idea of BE in 1991, Fiat and Naor [11] presented their model which is the first formal work in the area. They introduced the resiliency concept, and defined $k$-resilience to mean being resilient against a coalition of up to $k$ revoked users. Their best scheme required every user to store $O(k \log k \log n)$ keys and the center to broadcast $O(k^2 \log^2 k \log n)$ messages where $n$ is the total number of users.

After these works, Naor et al. proposed two subset–cover schemes, the complete subtree (CS) and subset difference (SD) [20]. In the CS scheme, each user stores $O(\log n)$ long-term keys and the transmission cost is $O(r \log(n/r))$, $r$ denoting the number of revoked users. The SD scheme decreased the transmission overhead to $O(r)$ at the expense of increasing the key storage to $O(\log^2 n)$. It was the most efficient scheme at the time of its proposal, and most of the recently proposed schemes [13,12] are also variations of the SD scheme.

User profiling has been used in a number of different applications. Recent works in broadcasting literature have made use of user profiles in order to increase broadcast efficiency in several aspects [10,17,15]. Similarly, web-user profiles have been heavily studied to serve individual users more effectively [16,21]. User profiling was also used in multicast key management [23] where the key distribution tree is optimized according to the members' expected stay time in a session.

In a recent study that utilizes subscriber profiles for BE efficiency, D'Arco and De Santis [8] proposed a method for efficient key storage, the other important performance metric for a BE system besides the transmission overhead, in presence of non-uniform revocation probabilities. The authors assumed these probabilities to be given and used this information to give fewer keys to users with a higher probability of revocation.

The idea of allowing free riders in a broadcast to get better performance was introduced by Abdalla et al. [2]. They investigated the usage of free riders and developed the basic intuitions about their effective assignment. Ramzan and Woodruff [22] recently proposed an algorithm to optimally choose the set of free riders in a CS scheme to minimize the transmission overhead. Ak et al. [4] extended this work to the SD scheme.

To the best of our knowledge, user profiles have not been used in the BE literature to reduce the transmission overhead despite the fact that the subset–cover framework is by its nature an excellent context for utilizing user profiles.

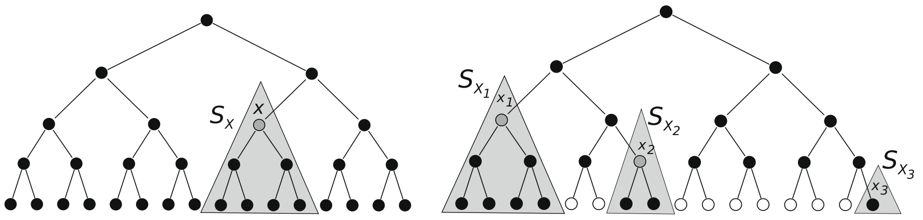
## 3. Subset–cover framework and the CS and SD schemes

A subset–cover BE scheme first generates a collection of subsets from the user set and associates a different long-term key with each subset. Then, every user in the system is installed with the long-term keys of the subsets he is included in.

To broadcast a message to a privileged user set $P$, the sender finds a cover $C$ from the subset collection such that

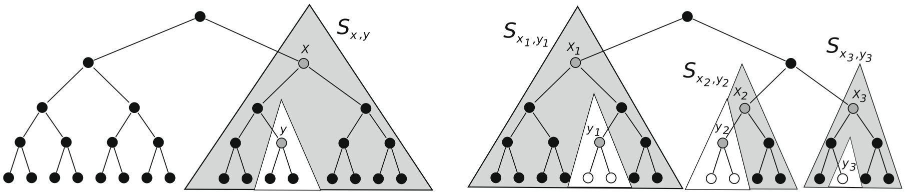
$$P = \cup_{S \in C} S$$

and encrypts the message using the keys of the subsets in $C$. The number of subsets in $C$, i.e., $|C|$, is called the *transmission cost* which is one of the main performance parameters for a BE scheme.



(a) A single CS subset (b) A CS cover

**Fig. 1.** A simple subset and cover of the CS scheme. Revoked users are denoted by white leaves.



(a) An SD subset (b) An SD cover

**Fig. 2.** A simple subset and cover of the SD scheme. Revoked users are shown with white leaves.

Both the CS and SD schemes obtain the user subsets by organizing the users in a binary tree. These schemes differ in the way they define their subsets.

In the CS scheme, the leaves of the subtree rooted at a node $x \in T$ correspond to a subset in the system. That is, for every node $x$, a subset is defined as

$$S_x = \{v | v \text{ is a leaf of } T(x)\},$$

where $T(x)$ denotes the subtree rooted at node $x$. An example subset and an example cover are illustrated in Fig. 1.

In the SD scheme, a subset is defined by two nodes $x$ and $y$ where $y$ is a descendant of $x$ in $T$. A subset $S_{x,y}$ is the set of leaves that are descendants of $x$ but not descendants of $y$. More formally, for every non-leaf node $x$, and every descendant $y$ of $x$, a subset is defined as

$$S_{x,y} = \{v | v \text{ is a leaf node, } v \in T(x) \text{ and } v \notin T(y)\}.$$

The total user set is also included as a subset in the SD scheme. An example subset and an example cover for the SD scheme are illustrated in Fig. 2.

Note that every subset in the CS scheme is also a subset in the SD scheme. The SD scheme also has the advantage of covering the leaves of several subtrees at once by a single subset. The increased key storage complexity of the SD scheme is reduced by an intelligent key generation scheme employing a pseudo-random function [20].

## 4. Broadcast encryption with user profiles

As noted in Section 2, the original CS and SD schemes treat the users identically when organizing the key distribution tree. However, if we have information about the user preferences and interests, we can use this information to group similar users together and make the BE scheme more efficient by constructing the subsets in a more clever way.

Consider a system supporting $b$ different types of broadcasts where type $j$ has a broadcast probability of $q_j$ and $\sum_{j=1}^{b} q_j = 1$. Let $p_{u,j}$ denote the probability of user $u$ subscribing to a broadcast of type $j$. We denote the *profile* of user $u$ with the $b$-tuple $(p_{u,1}, p_{u,2}, \ldots, p_{u,b})$.

As described above, both CS and SD schemes use a binary tree $T$ to organize the subsets and construct the cover. For a binary tree $T$, we will use $r_T$ to denote its root and $L_T$ to denote the set of its leaves. For a node $x \in T$, $par(x)$, $sib(x)$, $l(x)$ and $r(x)$ denote the parent, sibling, left child and right child of $x$ in $T$, respectively. For a node $x$, let $p_{x,j}$ denote the probability of all users (leaves) in $T(x)$ subscribing to a type $j$ broadcast, i.e.,

$$p_{x,j} = \prod_{u \in L_{T(x)}} p_{u,j},$$

where $L_{T(x)}$ is the set of leaves in the subtree with root $x$.

For clarity, we will investigate the cases $b = 1$ and $b \geqslant 1$ separately and we will use the terms *unitype* and *multitype* broadcast to refer to these cases, respectively.

### 4.1. Analysis of the CS Scheme with user profiles

We will first investigate the unitype broadcast case. In this case, we will use $p_u$ instead of $p_{u,1}$ to denote the probability of user $u$ being a subscriber. Let $P(S_x)$ be the probability of a CS subset $S_x$ being used in a cover.

**Lemma 4.1.** *In a CS tree, if x is a node other than the root, then*

$$P(S_x) = p_x - p_x p_{sib(x)} = p_x - p_{par(x)}.$$

*If x is the root* $r_T$*, then* $P(S_x) = p_{r_T} = \prod_{u \in L_T} p_u$.

**Proof.** For a node $x$ other than the root, if $S_x$ is in the cover, all the users in $L_{T(x)}$ must be subscribers. Also, there must be at least one non-subscriber in $L_{T(sib(x))}$, because otherwise $S_{par(x)}$ would be in the cover instead of $S_x$.

Note that if $x$ is the root, $S_x$ will be in the cover if and only if each user in $L_T$ is a subscriber, which happens with probability $\prod_{u \in L_T} p_u$. □

Let $E_{CS}(T)$ denote the expected cover size for a CS tree $T$.

**Theorem 4.2.** *For a CS tree T,*

$$E_{CS}(T) = \sum_{x \in L_T} p_x - \sum_{x \notin L_T} p_x. \tag{1}$$

**Proof.** The expected cover size for the CS scheme is equal to the sum of $P(S_x)$ over all $x \in T$. Hence,

$$E_{CS}(T) = \sum_{x \in T} P(S_x) = \sum_{x \in T, x \neq r_T} \left(p_x - p_{par(x)}\right) + p_{r_T}. \tag{2}$$

Note that since $T$ is a binary tree, for each non-leaf $x$, $p_x$ appears three times in the summation where one of them is positive and the other two are negative. And for a leaf $x$, the contribution to the summation is one $p_x$. Hence, (2) is equal to (1). □

Theorem 4.2 can be extended to the multitype case where $b \geqslant 1$:

**Theorem 4.3.** *For a CS scheme with* $b \geqslant 1$ *broadcast types, the expected cover size is*

$$E_{CS}(T) = \sum_{j=1}^{b} q_j \left( \sum_{x \in L_T} p_{x,j} - \sum_{x \notin L_T} p_{x,j} \right). \tag{3}$$

**Proof.** The expected cover size is the weighted average of the expected cover sizes for all broadcast types. Since each type $j$ has probability $q_j, E_{CS}(T)$ is equal to (3). □

### 4.2. Analysis of the SD scheme with user profiles

As in Section 4.1, we begin with an analysis for the unitype SD scheme: Let $P(S_{x,y})$ be the probability of an SD subset $S_{x,y}$ being used in a cover, and let

$$P(S_{*,y}) = \sum_{x \text{ is an ancestor of } y} P(S_{x,y}).$$

**Lemma 4.4.** *For a non-leaf, non-root* $y \in T$,

$$P(S_{*,y}) = p_{sib(y)}(1 - p_{l(y)})(1 - p_{r(y)}) \tag{4}$$

*and for a leaf* $y \in L_T$

$$P(S_{*,y}) = p_{sib(y)}(1 - p_y). \tag{5}$$

**Proof.** If $S_{x,y}$ is used in the cover, for a node $y$ and one of its ancestors $x$, all the users in $L_{T(sib(y))}$ must be subscribers. Furthermore, if $y$ is a non-leaf, non-root node, there must be at least one non-subscriber in both $L_{T(l(y))}$ and $L_{T(r(y))}$.

If $y$ is a leaf node and $S_{x,y}$ is in the cover $sib(y)$ must be a subscriber and $y$ cannot. Hence (4) and (5) follow. □

Let $E_{SD}(T)$ denote the expected cover size for an SD tree.

**Theorem 4.5.** *In an SD tree T,*

$$E_{SD}(T) = \prod_{y \in L_T} p_y + \sum_{y \in L_T} \left(p_{sib(y)}(1 - p_y)\right) + \sum_{\substack{y \notin L_T \\ y \neq r_T}} \left(p_{sib(y)}(1 - p_{l(y)})(1 - p_{r(y)})\right). \tag{6}$$

**Proof.** The expected cover size for the SD scheme, $E_{SD}(T)$, is equal to the sum of $P(S_{*,y})$ for all $y \in T$ except the root $r_T$. Besides, if all of the users subscribe to a broadcast, which happens with probability $\prod_{y\in L_T} p_y$, the cover size will be one. Hence,

$$E_{SD}(T) = \sum_{y\in T-\{r_T\}} P(S_{*,y}) + \prod_{y\in L_T} p_y.$$

By substituting (4) and (5) for $P(S_{*,y})$, (6) follows. □

Theorem 4.5 can be extended to the multitype case:

**Theorem 4.6.** *For an SD scheme with $b \geqslant 1$ broadcast types, the expected cover size is*

$$E_{SD}(T) = \sum_{j=1}^{b} q_j E_{SD}(T,j), \tag{7}$$

*where*

$$E_{SD}(T,j) = \prod_{y\in L_T} p_{y,j} + \sum_{y\in L_T}\left(p_{sib(y),j}(1-p_{y,j})\right) + \sum_{\substack{y\notin L_T\\ y\neq r_T}}\left(p_{sib(y),j}(1-p_{l(y),j})(1-p_{r(y),j})\right)$$

*is the expected cover size for the broadcast type $j$ with probability $q_j$.*

**Proof.** The expected cover size is the weighted average of the expected cover sizes for all broadcast types. Since each type $j$ has probability $q_j$, $E_{SD}(T)$ is equal to (7). □

## 5. Optimal CS tree construction

In this section, we will give two optimal tree construction algorithms for the unitype CS scheme. We will assume that for users $u_1, u_2, \dots, u_n$, the subscription probabilities are $p_{u_1} \geqslant p_{u_2} \geqslant \cdots \geqslant p_{u_n}$; i.e., the users are indexed with respect to their subscription probabilities in decreasing order. We say that a CS tree is *optimal* if it minimizes the expected cover size.

We will consider the optimal CS tree organization problem for two different settings: First, the CS tree has to be a balanced tree, and second, the CS tree is not necessarily balanced. We will refer the former as the balanced setting and the latter as the general setting. Lemma 5.1 below applies to both settings:

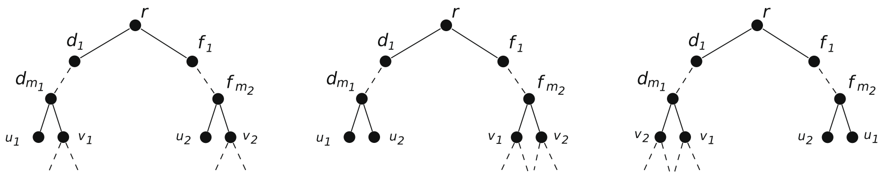
**Lemma 5.1.** *In a CS scheme with unitype broadcast, there exists an optimal tree where $u_1$ and $u_2$, the two users with the highest subscription probabilities, are siblings.*

**Proof.** First recall that for any binary tree $T$, balanced or unbalanced, $E_{CS}(T) = \sum_{x\in L_T} p_x - \sum_{x\notin L_T} p_x$. Let $T$ be an optimal tree with the minimum expected cover size. If $u_1$ and $u_2$ are siblings in $T$ then we are done. Otherwise let $v_1$ and $v_2$ be the siblings of $u_1$ and $u_2$, respectively. Since we are investigating both settings, balanced and general, $v_1$ and $v_2$ may be internal nodes of $T$. Let $r$ be the first common ancestor of $u_1$ and $u_2$ and let $path(r,u_1) = (r, d_1, d_2, \dots, d_{m_1}, u_1)$ and $path(r,u_2) = (r, f_1, f_2, \dots, f_{m_2}, u_2)$ be the paths from $r$ to $u_1$ and $u_2$, respectively, as shown in Fig. 3a.

Note that $p_{u_1}p_{v_1}$ is a factor of each term in $\{p_{d_1}, p_{d_2}, \dots, p_{d_{m_1}}\}$, and $p_{u_2}p_{v_2}$ is a factor of each term in $\{p_{f_1}, p_{f_2}, \dots, p_{f_{m_2}}\}$. Let $D = \sum_{i=1}^{m_1} p_{d_i}/(p_{u_1}p_{v_1})$ and $F = \sum_{i=1}^{m_2} p_{f_i}/(p_{u_2}p_{v_2})$. Let $V(u_1,u_2)$ be the combined set of nodes on $path(d_1, d_{m_1})$ and $path(f_1, f_{m_2})$. The expected cover size can be written as

$$E_{CS}(T) = \sum_{x\in L_T} p_x - \sum_{x\notin L_T\cup V(u_1,u_2)} p_x - \sum_{x\in V(u_1,u_2)} p_x = \sum_{x\in L_T} p_x - \sum_{x\notin L_T\cup V(u_1,u_2)} p_x - (p_{u_1}p_{v_1}D + p_{u_2}p_{v_2}F),$$

where the first two terms do not change if we swap $u_1$ and $v_2$, or $u_2$ and $v_1$, as shown in Fig. 3b and c, respectively. We have two cases:



(a) Before swap (b) Swap $u_1$ and $v_2$ (c) Swap $u_2$ and $v_1$

**Fig. 3.** Structure of $T(r)$ before and after the swap operations.

(1) **D < F**: Let $T'$ be the tree obtained by swapping $u_1$ and $v_2$ as in Fig. 3b. Since we have $p_{u_1} \geqslant p_{v_2}$ and $p_{u_2} \geqslant p_{v_1}$, the difference

$$E_{CS}(T) - E_{CS}(T') = p_{v_1}p_{v_2}D + p_{u_1}p_{u_2}F - p_{u_1}p_{v_1}D - p_{u_2}p_{v_2}F = p_{u_2}F(p_{u_1} - p_{v_2}) - p_{v_1}D(p_{u_1} - p_{v_2})$$

is non-negative. Given that $T$ is optimal, we must have that $p_{u_1} = p_{v_2}$ and swapping $u_1$ and $v_2$ does not change the expected cost.

(2) **D > F**: Let $T'$ be the tree obtained by swapping $v_1$ and $u_2$ as in Fig. 3c. Since we have $p_{u_2} \geqslant p_{v_1}$ and $p_{u_1} \geqslant p_{v_2}$, the difference

$$E_{CS}(T) - E_{CS}(T') = p_{u_1}p_{u_2}D + p_{v_1}p_{v_2}F - p_{u_1}p_{v_1}D - p_{u_2}p_{v_2}F = p_{u_1}D(p_{u_2} - p_{v_1}) - p_{v_2}F(p_{u_2} - p_{v_1})$$

is non-negative. Given that $T$ is optimal, we must have that $p_{u_2} = p_{v_1}$ and swapping $u_2$ and $v_1$ does not change the expected cost.

(3) **D = F**: Let $T'$ be the tree obtained by swapping $u_1$ and $v_2$ as in Fig. 3b. (Note that we could choose to swap $u_2$ and $v_1$, as well.) Since we have $p_{u_1} \geqslant p_{v_2}$ and $p_{u_2} \geqslant p_{v_1}$, the difference

$$E_{CS}(T) - E_{CS}(T') = p_{v_1}p_{v_2}D + p_{u_1}p_{u_2}F - p_{u_1}p_{v_1}D - p_{u_2}p_{v_2}F = p_{u_2}F(p_{u_1} - p_{v_2}) - p_{v_1}D(p_{u_1} - p_{v_2})$$

is non-negative. Given that $T$ is optimal, we must have that $p_{u_2}(p_{u_1} - p_{v_2}) - p_{v_1}(p_{u_1} - p_{v_2}) = 0$ which implies $(p_{u_2} - p_{v_1})(p_{u_1} - p_{v_2}) = 0$. (Here, note that if we had chosen to swap $u_2$ and $v_1$ we would end up with this same equation, by symmetry.) Then, either $p_{u_2} = p_{v_1}$ or $p_{u_1} = p_{v_2}$. If $p_{u_2} = p_{v_1}$, swapping $u_2$ and $v_1$ does not change the expected cost. If $p_{u_1} = p_{v_2}$, in this case, swapping $u_1$ and $v_2$ does not change the expected cost. So in either case, we can come up with an optimal tree where $u_1$ and $u_2$ are siblings.

Hence, for all three cases we can say that the two nodes with maximum subscription probabilities can be paired in a tree that preserves the optimality. □

### 5.1. Optimality for balanced trees

In this section we give the optimal CS tree construction algorithm with the balanced tree constraint. We assume that $n$ is a power of 2 throughout the discussion in this section.

**Lemma 5.2.** *For a unitype CS scheme, there exists an optimal balanced CS tree where the pairs* $(u_1, u_2), (u_3, u_4), \ldots, (u_{n-1}, u_n)$ *are siblings of each other.*

**Proof.** From Lemma 5.1, we know that there exists an optimal balanced tree $T$ such that $(u_1, u_2)$ are siblings. Similar to the proof of Lemma 5.1, starting with $T$, the other users can be paired as siblings by swapping operations by an iterative process that starts with $(u_3, u_4)$. Note that $u_3$ and $u_4$ are the users with the two maximum subscription probabilities excluding $u_1$ and $u_2$; hence the optimality is preserved after the swap operations. Since the tree $T$ is balanced at the beginning, each leaf $T$ will have a leaf sibling at any time. □

Now we are ready to prove the main result for the balanced case.

**Theorem 5.3.** *In a unitype CS scheme with the balanced tree constraint, sorting the users in the leaf level with respect to their subscription probabilities gives the minimum expected cover size.*

**Proof.** Let $T^{(k)}$ denote an optimal balanced CS tree of depth $k$ whose leaf nodes are grouped as stated in Lemma 5.2 as $(u_1, u_2), (u_3, u_4), \ldots, (u_{n-1}, u_n)$ for a given user set. Let $H^{(k)}$ denote the balanced tree of depth $k$ on the same user set, obtained by ordering the leaves according to the sorted $p_{u_i}$ values. We will use induction on the depth of the tree to prove that $E_{CS}(T^{(k)}) = E_{CS}(H^{(k)})$ for any $k$.

For the basic case, for any set of two nodes, obviously $E_{CS}(T^{(1)}) = E_{CS}(H^{(1)})$. Now assume that the claim is also true for all balanced trees with depth less than $k$. For the tree $T^{(k)}$ for a given user set, let $T'$ denote the subtree of depth $k-1$ which has the paired nodes $u_{2i-1,2i}$ as its leaves, with probabilities $p_{u_{2i-1,2i}} = p_{u_{2i-1}}p_{u_{2i}}$, for $1 \leqslant i \leqslant n/2$. Let $H^{(k-1)}$ denote the balanced tree obtained by sorting the same set of nodes, $\{u_{1,2}, \ldots, u_{n-1,n}\}$. By induction, $E_{CS}(T') \geqslant E_{CS}(H^{(k-1)})$. Also from (1),

$$E_{CS}(T^{(k)}) = E_{CS}(T') + \sum_{i=1}^{n} p_{u_i} - 2\sum_{i=1}^{n/2} p_{u_{(2i-1)(2i)}},$$

$$E_{CS}(H^{(k)}) = E_{CS}(H^{(k-1)}) + \sum_{i=1}^{n} p_{u_i} - 2\sum_{i=1}^{n/2} p_{u_{(2i-1)(2i)}}.$$

Hence, $E_{CS}(T^{(k)}) \geqslant E_{CS}(H^{(k)})$; and since $T^{(k)}$ is optimal, $H^{(k)}$ is also optimal. □

### 5.2. Optimality for the general setting

The optimal construction for the general setting is also based on Eq. (1) and Lemma 5.1, which are true independent of the tree's being balanced.

Let $T_i$ be a tree with one user node $u_i$. Let $T \circ T'$ denote the union of two trees constructed by adding a new root $r$ and connecting $T$ and $T'$ to $r$ as the left and right subtrees. The Uni-Gen Cluster algorithm below takes the subscription probabilities as inputs and constructs a broadcast tree with the minimum expected cover size in a style similar to Huffman trees [14].

**Algorithm 1.** Uni-Gen Cluster

```
$\mathscr{T} \leftarrow \{T_1, T_2, \ldots, T_n\}$, , where $T_i$ is the tree containing just one node $u_i$
while $|\mathscr{T}|$ is not equal to 1 do
    Find the pair $T, T' \in \mathscr{T}$ with maximum $p_{r_T}$ and $p_{r_{T'}}$
    Construct the merged tree $T'' = T \circ T'$
    $\mathscr{T} \leftarrow \mathscr{T} \setminus \{T, T'\}$
    $\mathscr{T} \leftarrow \mathscr{T} \cup T''$
return $\mathscr{T}$
```

The algorithm works in a bottom-up fashion. At each iteration, two trees $T$ and $T'$ with the largest $p_{r_T}$ and $p_{r_{T'}}$ are selected. These trees are extracted from the queue, and a new tree $T'' = T \circ T'$ with a new root $r_{T''}$ is inserted where $p_{r_{T''}} = p_{r_T} p_{r_{T'}}$. The optimality proof of the tree obtained by this algorithm is given in Theorem 5.4:

**Theorem 5.4.** *For a unitype CS scheme, the tree obtained by the* Uni-Gen Cluster *algorithm is optimal with the minimum expected cover size.*

**Proof.** Let $T^{(k)}$ denote an optimal CS tree with $k$ leaves where $u_1$ and $u_2$ are connected as siblings as stated in Lemma 5.1, for a given user set. Let $H^{(k)}$ denote the tree with the same $k$ leaves constructed by the algorithm Uni-Gen Cluster. We will use induction on the number of leaves in the tree to prove that $E_{CS}(T^{(k)}) = E_{CS}(H^{(k)})$ for any $k$.

For the basic case, for any set of two nodes, obviously $E_{CS}(T^{(2)}) = E_{CS}(H^{(2)})$. Now assume that the claim is also true for all trees with $k-1$ or fewer leaves. For the tree $T^{(k)}$ for a given user set, let $T'$ denote the tree with $k-1$ leaves obtained by merging $u_1$ and $u_2$ into a new node $u_{12}$, with probability $p_{u_{12}} = p_{u_1} p_{u_2}$. Let $H^{(k-1)}$ be the tree constructed by the Uni-Gen Cluster algorithm from the same set of leaves. By induction, $E_{CS}(T') \geqslant E_{CS}(H^{(k-1)})$. Also from (1),

$$E_{CS}(T^{(k)}) = E_{CS}(T') + p_{u_1} + p_{u_2} - 2p_{u_{12}},$$

$$E_{CS}(H^{(k)}) = E_{CS}(H^{(k-1)}) + p_{u_1} + p_{u_2} - 2p_{u_{12}}$$

and it follows that $E_{CS}(T^{(k)}) \geqslant E_{CS}(H^{(k)})$. We know $T^{(k)}$ is optimal, therefore $H^{(k)}$ is optimal. □

## 6. The case of multitype broadcasts

In multitype BE schemes, we cannot simply group the users with respect to their subscription probabilities since there are $b$ different subscription probabilities for each user. Nevertheless, if we place similar users closer in the tree, the number of subtrees containing them will increase, hence smaller covers can be obtained. We will first focus on the probability of two users being interested in a common broadcast. If two users' probabilities of being interested in the same broadcast are both high, we will say that these two users are *similar*. We define the similarity of two user profiles as the weighted sum of the products of their probabilities over different broadcast types:

$$\mathrm{Sim}(u, v) = \sum_{j=1}^{b} q_j p_{u,j} p_{v,j}.$$

Assuming that the user subscription decisions are independent, the similarity between two users is the probability of both subscribing to a common broadcast.

Extending the formulation for individual users to groups of users, we define the similarity of groups of users as follows: We call a set of users *similar* if the probability of all users being interested in the same broadcast is high. Let $T$ and $T'$ be two trees containing disjoint sets of users as their leaves. Then the similarity of these trees are

$$\mathrm{Sim}(T, T') = \sum_{j=1}^{b} q_j p_{r_T,j} p_{r_{T'},j},$$

where

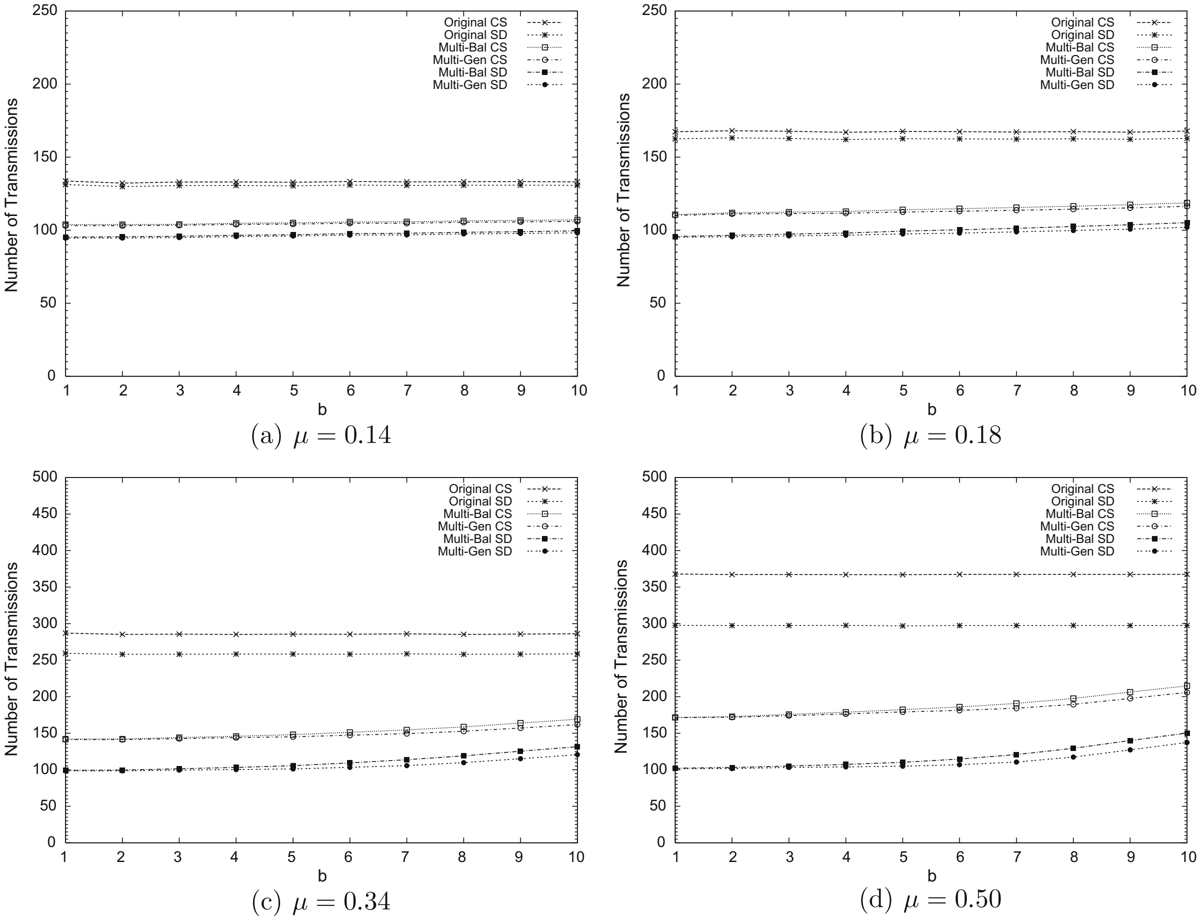
$$p_{r_T,j} = \prod_{u \in L_T} p_{u,j}.$$

### 6.1. The balanced tree algorithm

The Multi-Bal Cluster algorithm below clusters the set of users according to the Sim metric and organizes them as the leaves of a balanced binary tree. It works by arranging the tree in levels. It starts with the bottom level by organizing the most similar users in pairs. Then, at every level, pairs of nodes/subsets are matched and clustered according to their similarities.

**Algorithm 2.** Multi-Bal Cluster

```
$\mathcal{T} \leftarrow \{T_1, T_2, \ldots, T_n\}$, where $T_i$ is the tree containing just one node $u_i$
$\mathcal{S} \leftarrow \{\}$
while $|\mathcal{T}|$ is not equal to 1 do
  while $\mathcal{T}$ is not empty do
  Find the pair $T, T' \in \mathcal{T}$ with maximum $\mathrm{Sim}(T, T')$
    Construct the merged tree $T'' = T \circ T'$
    $\mathcal{T} \leftarrow \mathcal{T} \setminus \{T, T'\}$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{T''\}$
  $\mathcal{T} \leftarrow \mathcal{S}$
  $\mathcal{S} \leftarrow \{\}$
return $\mathcal{T}$
```



(a) $\mu = 0.14$

(b) $\mu = 0.18$

(c) $\mu = 0.34$

(d) $\mu = 0.50$

**Fig. 4.** Transmission costs of the CS and SD schemes in their basic form and with subscriber profiling. Four different plots are given for four different values of the interested user density, 5%, 10%, 30% and 50%, making the population mean 0.14, 0.18, 0.34 and 0.5, respectively. The results indicate that significant reductions are possible over the basic CS and SD schemes by the proposed algorithms. On the other hand, there is only a slight difference between the balanced-tree algorithms and their generalized counterparts.

The algorithm works in a bottom-up fashion; in the first iteration, it clusters the pairs of leaves starting with the most similar pair. The pairs in these clusters will be the siblings in the resulting tree. In the next iteration, these clusters are paired and this process continues until just one cluster remains and the tree is constructed. Note that the algorithm constructs a balanced binary tree since the list $\mathscr{T}$ always contains trees of the same depth. For $b = 1$, the Multi-Bal Cluster algorithm sorts the users with respect to their subscription probabilities, which we know to give the optimal CS tree for $b = 1$.

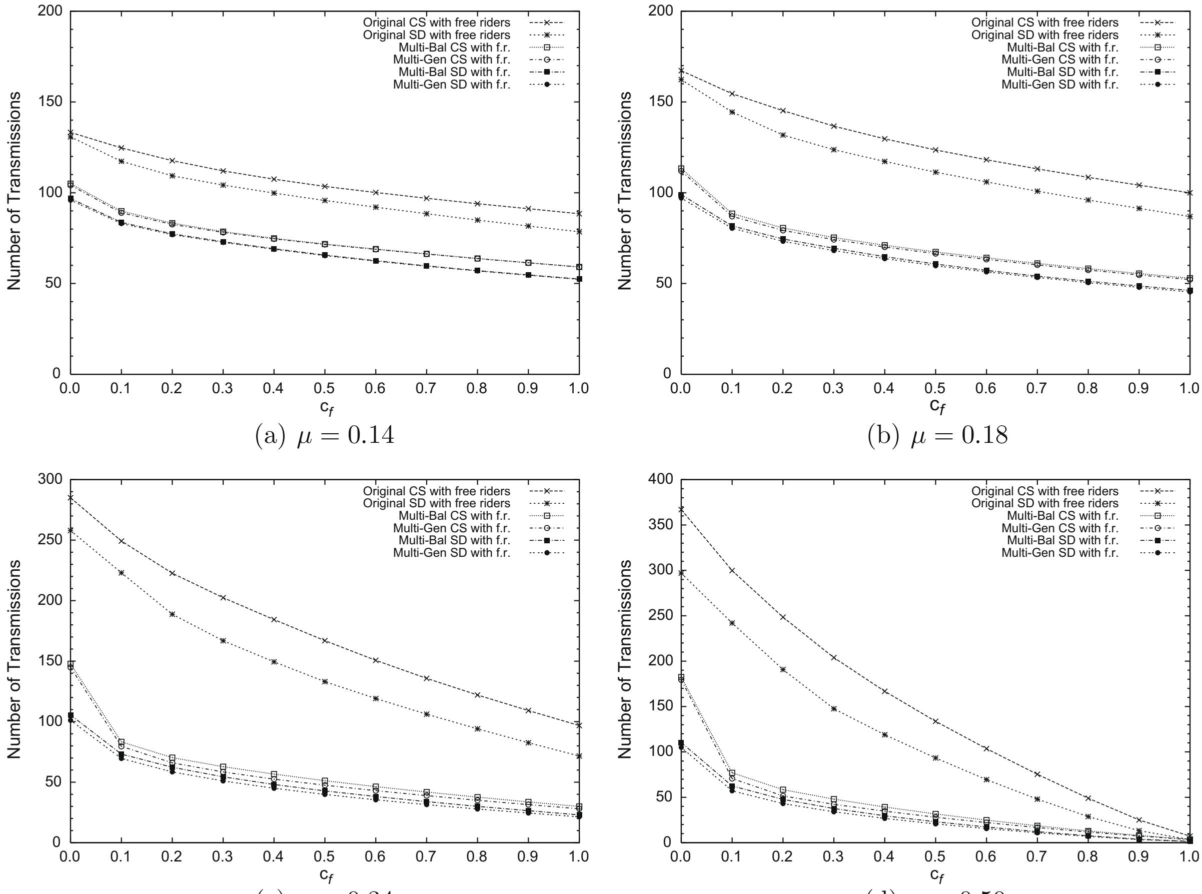
### 6.2. The general algorithm

The similarity approach can also be used for the general setting where the CS and SD trees need not be balanced.

**Algorithm 3.** Multi-Gen Cluster

```
$\mathscr{T} \leftarrow \{T_1, T_2, \ldots, T_n\}$, where $T_i$ is the tree containing just one node $u_i$
while while $|\mathscr{T}|$ is not equal to 1 do
    Find the pair $T, T' \in \mathscr{T}$ with maximum $\mathrm{Sim}(T, T')$
    Construct the merged tree $T'' = T \circ T'$
    $\mathscr{T} \leftarrow \mathscr{T} \setminus \{T, T'\}$
    $\mathscr{T} \leftarrow \mathscr{T} \cup \{T''\}$
return $\mathscr{T}$
```

As in the balanced setting, the Multi-Gen Cluster algorithm constructs the tree in a bottom-up fashion. Similar to its uni-type counterpart Uni-Gen Cluster, at each iteration the algorithm chooses and merges the most similar pair.



(a) $\mu = 0.14$

(b) $\mu = 0.18$

(c) $\mu = 0.34$

(d) $\mu = 0.50$

**Fig. 5.** Transmission costs of the CS and SD schemes with free riders, in their basic form and with user profiling, where the number of broadcast types is $b = 5$. The results indicate that a sharp decrease in the transmission cost is possible by allowing a limited number of free riders, especially for higher values of $\mu$.

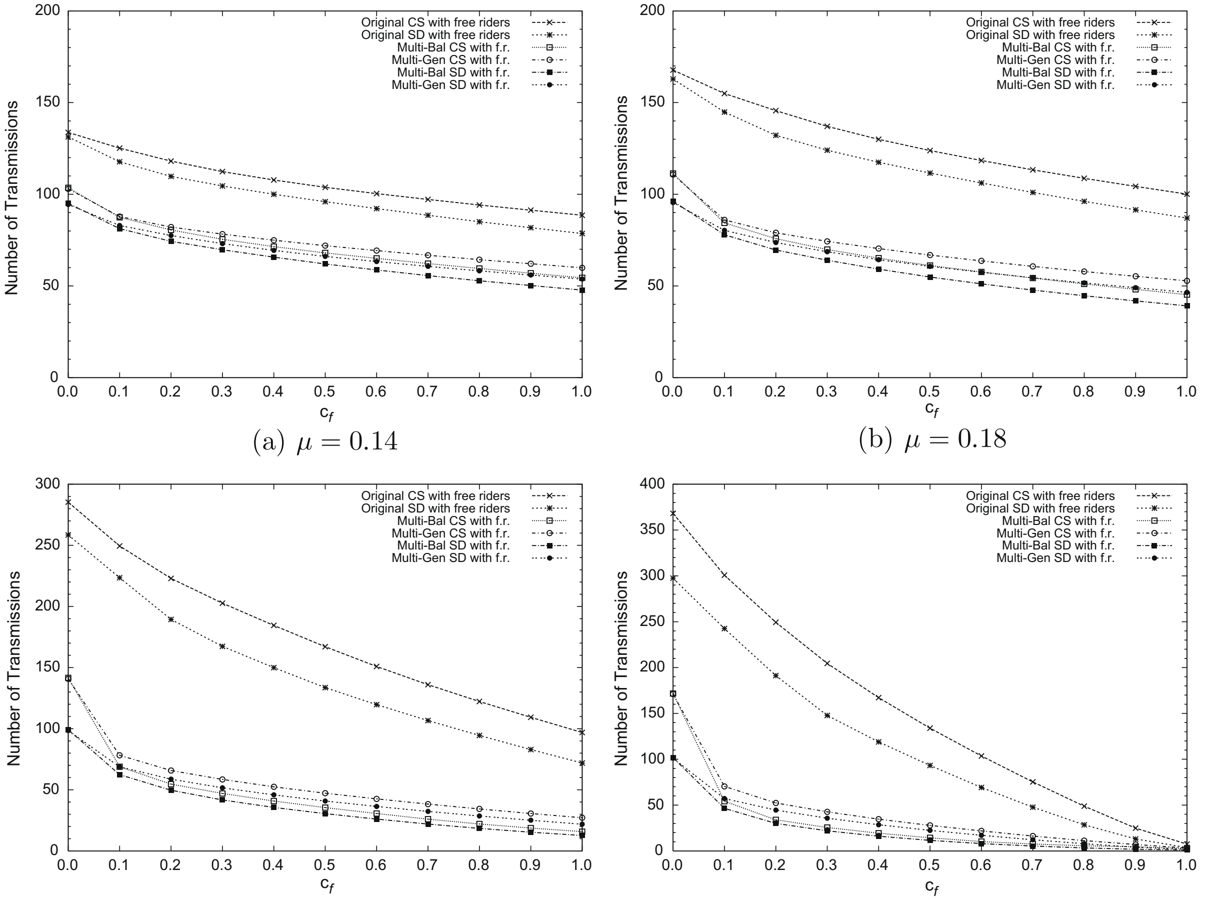
## 7. Experimental results

We tested the performance of the proposed algorithms against the standard BE approach by running a large number of experiments on synthetically generated user profiles. The user profiles were carefully generated with various characteristics to be representatives of a wide variety of applications.

We experimented with a population of $n = 1024$ users. Each user profile contains $b$ subscription probabilities for some $1 \leqslant b \leqslant 10$. For each broadcast type $j$, the subscription probabilities $p_{i,j}$ are randomly generated by using a bimodal density function based on two uniform distributions with respective means of $\mu_1 = 0.9$ and $\mu_2 = 0.1$ to represent the interested and uninterested user populations, respectively. The overall population mean, $\mu$, is determined according to the weight of the interested users in the population. For each set of experiments, we compared the average transmission costs of the basic CS and SD schemes with those obtained by subscriber profiling. In the experiments, the broadcast types are taken to be equally likely with a probability of $q_j = 1/b$ for each $1 \leqslant j \leqslant b$.

The experimental results are summarized in Fig. 4 where the transmission costs of the basic and similarity-based CS and SD schemes are compared. The results show that utilizing the user profiles with the given similarity metric can reduce the transmission cost significantly. For the balanced-tree CS scheme, the reduction rate is about 20–45% for larger values of $b$ and more than 20–50% for smaller values of $b$. The improvements are even more significant for the balanced-tree SD scheme, with 25–55% improvement for larger values of $b$ and 25–65% for smaller $b$ values. The cost reduction rates get higher with larger population means.

The improvement rates for the generalized (unbalanced) algorithm are only slightly better than those of the balanced tree algorithm for smaller values of $b$ and the population mean; however as the value of $b$ gets larger and the population mean increases, the generalized algorithm provides better improvement rates that allow up to an additional 5% reduction in the transmission costs.



(a) $\mu = 0.14$

(b) $\mu = 0.18$

(c) $\mu = 0.34$

(d) $\mu = 0.50$

**Fig. A.1.** $b = 1$.

## 8. Using similarity approach with free riders

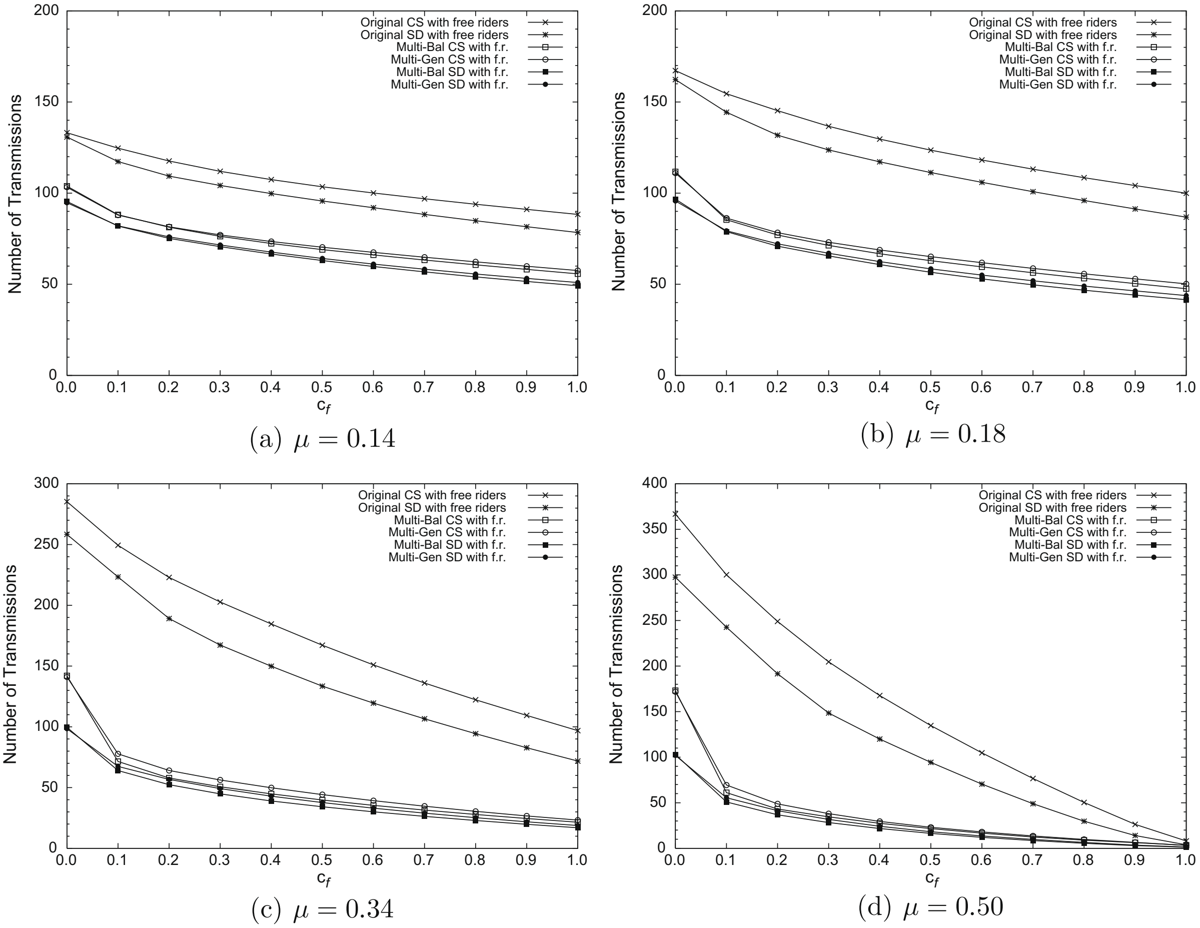
*Free riders* are the users who are able to decrypt a broadcast session although they are not subscribed to it. Some free riders can be allowed in a BE system in order to lower the transmission cost by relaxing the restriction that the cover must exactly match the privileged user set. Free riders must be assigned carefully in order to reduce the cost effectively. Optimal free rider assignment algorithms for the CS and SD schemes have recently been given by Ramzan and Woodruff [22] and Ak et al. [4], respectively.

Our proposed similarity-based organization algorithms can be expected to be even more effective when a few free riders can be tolerated. Our approach aims to obtain large subsets by taking a set of consecutive users as subscribers. Hence, if a few remaining non-subscribers can be tolerated as free riders in such a sequence of subscribers, a larger and fully privileged subset can be obtained, leading to more compact covers.

Let $f$ denote the number of free riders that can be allowed, and let $c_f$ denote the *free rider ratio*, $f/(n-r)$, where $n$ and $r$ are the total number of users and the number of revoked users, respectively. We tested the performance of our algorithms with a given number of free riders by a large number of simulation experiments with $n = 1024$ and $0.1 \leqslant c_f \leqslant 1.0$, where the user profiles are generated with the same parameters used for the experiments with no free riders in Section 7.

Fig. 5 shows the results for the basic and the similarity-based CS and SD schemes with free riders for $b = 5$ broadcast types. Additional plots for different values of $b$ are provided in Appendix A, which turn out to be parallel to the plots given here for $b = 5$. The plots demonstrate the improvements in the transmission cost according to the free rider ratio $c_f$. The results show that significant savings can be achieved by using the similarity approach and allowing a very limited number of free riders. A sharp decrease in the transmission cost can be obtained by using the similarity approach with a free rider ratio of just 10%, while the improvement rates of the basic CS and SD schemes appear to be linear with $c_f$.

The experiments show that allowing a free rider ratio of 10% reduces the transmission cost of the similarity-based CS scheme by 40–70% and the similarity-based SD scheme by 35–55%, whereas the transmission cost of the original schemes are only reduced by 20%. As a result, the similarity-based CS scheme has 65–85% lower cost than the original CS scheme and



(a) $\mu = 0.14$

(b) $\mu = 0.18$

(c) $\mu = 0.34$

(d) $\mu = 0.50$

**Fig. A.2.** $b = 2$.

the similarity-based SD scheme has 60–80% lower cost than the original SD scheme when a free rider ratio of 10% is allowed. The similarity approach becomes more effective at smaller values of $b$ and at greater values of $\mu$, which is consistent with the previous experiments with no free riders.

The balanced-tree and the generalized algorithms have similar transmission costs for a given number of free riders, while the generalized algorithms have a slight cost advantage over their balanced-tree counterparts.

## 9. Conclusion

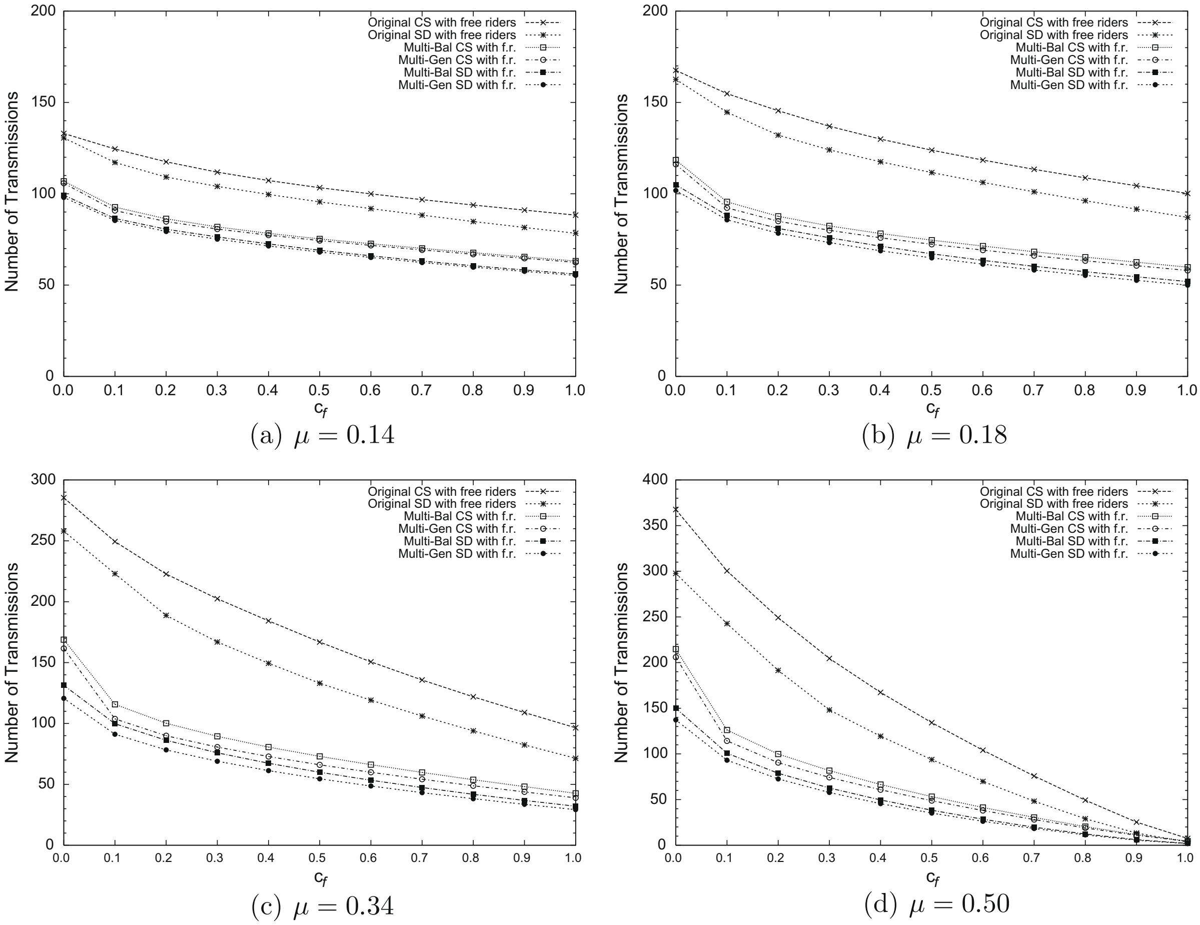
In this paper, we analyzed the problem of reducing the transmission costs of subset–cover based BE schemes of CS and SD by utilizing information about user interests. We gave optimal algorithms for the CS scheme when only one type of broadcast exists. For the multitype case, we proposed a similarity approach which can be used in both CS and SD schemes. The simulation experiments showed that the proposed algorithms are effective and can provide significant reductions in the transmission complexity of a BE system. The gains obtained by the proposed algorithms turn out to be even more significant when a limited number of free riders can be tolerated in the system.

## Acknowledgement

This work is supported in part by the Turkish Scientific and Technological Research Agency (TUBİTAK), under Grant No. 108E150.

## Appendix A. Simulation results

In this section, we provide further simulation experiment results for the performance of the proposed optimization algorithms with free riders, for different values of the number of broadcast types, $b$. The results turn out to be mostly parallel to those presented in Section 8. See Figs. A.1–A.3.



(a) $\mu = 0.14$

(b) $\mu = 0.18$

(c) $\mu = 0.34$

(d) $\mu = 0.50$

**Fig. A.3.** $b = 10$.

## References

[1] AACS-Advanced Access Content System, 2007. http://www.aacsla.com.

[2] M. Abdalla, Y. Shavitt, A. Wool, Key management for restricted multicast using broadcast encryption, IEEE/ACM Transactions on Networking 8 (4) (2000) 443–454.

[3] W. Aiello, S. Lodha, R. Ostrovsky, Fast digital identity revocation, in: CRYPTO'98, LNCS, vol. 1462, Springer-Verlag, 1998, pp. 137–152.

[4] M. Ak, K. Kaya, A.A. Selçuk, Optimal subset-difference broadcast encryption with free riders, Information Sciences 179 (20) (2009) 3673–3684.

[5] S. Berkovits. How to broadcast a secret, in: EUROCRYPT'91, LNCS, vol. 547, Springer-Verlag, 1991, pp. 535–541.

[6] C. Blundo, A. Cresti, Unconditional secure conference key distribution schemes with disenrollment capability, Information Sciences 120 (1-4) (1999) 113–130.

[7] J.-T. Chung, C.-M. Li, T. Hwang, All-in-one group-oriented cryptosystem based on bilinear pairing, Information Sciences 177 (24) (2007) 5651–5663.

[8] P. D'Arco, A. De Santis, Optimizing SD and LSD in presence of non-uniform probabilities of revocation, in: Proc. of International Conference on Information Theoretic Security (ICITS), 2007.

[9] E. David, S. Kraus, Agents for information broadcasting, in: 6th International Workshop on Intelligent Agents VI, Agent Theories, Architectures, and Languages (ATAL'99), London, UK, 2000, Springer-Verlag, pp. 91–105.

[10] E. Dees, Decentralized advertisement recommendation on IPTV, Vrije Universiteit, Amsterdam, 2007.

[11] A. Fiat, M. Naor, Broadcast encryption, in: CRYPTO'93, LNCS, vol. 773, Springer-Verlag, 1993, pp. 480–491.

[12] M.T. Goodrich, J.Z. Sun, R. Tamassia, Efficient tree based revocation in groups of low-state devices, in: CRYPTO'04, LNCS, vol. 3152, Springer-Verlag, 2004, pp. 511–527.

[13] D. Halevy, A. Shamir, The LSD broadcast encryption scheme. in: CRYPTO'02, LNCS, vol. 2442, Springer-Verlag, London, UK, 2002, pp. 47–60.

[14] D. Huffman, A method for the construction of minimum redundancy codes, Proceedings of the Institute of Radio Engineers 40 (9) (1952) 1098–1101.

[15] M. Kim, S. Kang, M. Kim, J. Kim, Target advertisement service using TV viewers profile inference, in: Advances in Multimedia Information Processing – Pacific Rim Conference on Multimedia 2005, Springer, Berlin, Germany, 2005, pp. 202–211.

[16] R. Kosala, H. Blockeel, Web mining research: a survey. ACM SIGKDD Explorations, 2, 2000.

[17] J. Lim, M. Kim, B. Lee, M. Kim, H. Lee, H. Lee, A target advertisement system based on TV viewer's profile reasoning, Multimedia Tools and Applications 2 (2007).

[18] J. Lotspiech, S. Nusser, F. Pestoni, Broadcast encryption's bright future, Computer 35 (2002) 57–63.

[19] J. Nam, J. Paik, U.M. Kim, D. Won, Resource-aware protocols for authenticated group key exchange in integrated wired and wireless networks, Information Sciences 177 (23) (2007) 5441–5467. Including: Mathematics of Uncertainty, A selection of the very best extended papers of the IMS-2004 held at Sakarya University in Turkey.

[20] D. Naor, M. Naor, J. Lotspiech, Revocation and tracing schemes for stateless receivers, in: CRYPTO'01, LNCS, vol. 2139, Springer-Verlag, 2001, pp. 41–62.

[21] O. Nasraoui, World wide web personalization, in: J. Wang (Ed.), Encyclopedia of Data Mining and Data Warehousing, Idea Group, 2005 (invited chapter).

[22] Z. Ramzan, D. Woodruff, Fast algorithms for the free riders problem in broadcast encryption, in: CRYPTO'06, LNCS, vol. 4117, Springer-Verlag, 2006, pp. 308–325.

[23] A.A. Selçuk, D. Sidhu, Probabilistic optimization techniques for multicast key management, Computer Networks 40 (2) (2002) 219–234.

[24] C.B.S. Traw, Protecting digital content within the home, Computer 34 (2001) 42–47.

[25] D.M. Wallner, E.J. Harder, R.C. Agee, Key Management for Multicast: Issues and Architectures, Internet Draft, 1999.

[26] C.K. Wong, M. Gouda, S.S. Lam, Secure group communication using key graphs, in: SIGCOMM'98, September 1998, pp. 68–79.