Lower Bounds for Subset Cover Based Broadcast Encryption

Per Austrin[*] and Gunnar Kreitz

KTH – Royal Institute of Technology, Stockholm, Sweden
{austrin,gkreitz}@kth.se

Abstract. In this paper, we prove lower bounds for a large class of Subset Cover schemes (including all existing schemes based on pseudo-random sequence generators). In particular, we show that

- For small r, bandwidth is $\Omega(r)$
- For some r, bandwidth is $\Omega(n/\log(s))$
- For large r, bandwidth is $n - r$

where n is the number of users, r is the number of revoked users, and s is the space required per user.

These bounds are all tight in the sense that they match known constructions up to small constants.

Keywords: Broadcast Encryption, Subset Cover, key revocation, lower bounds.

1 Introduction

A Broadcast Encryption scheme is a cryptographic construction allowing a trusted sender to efficiently and securely broadcast information to a dynamically changing group of users over an untrusted network. The area is well studied and there are numerous applications, such as pay-per-view TV, CD/DVD content protection, and secure group communication. For instance, the new Advanced Access Content System (AACS) standard, which is used for content protection with next-generation video disks, employs Broadcast Encryption.

A Broadcast Encryption scheme begins with an initialization phase where every user is given a set of secrets. Depending on the application, a "user" in the scheme could be an individual, a subscriber module for a cable TV receiver, or a model of HD-DVD players. When the initialization is complete, the sender can transmit messages. For each message it wants to transmit, it selects a subset of users to receive the message. We will refer to this subset of intended recipients as *members* (another common name is the privileged set). It then encrypts and broadcasts the message, using the secrets of the members, in such a way that only the members can decrypt the broadcast. Even if all the non-members (or *revoked* users) collude, they should not be able to decrypt the broadcast. The term key revocation scheme is also used for Broadcast Encryption schemes.

[*] Research funded by Swedish Research Council Project Number 50394001.

S. Vaudenay (Ed.): AFRICACRYPT 2008, LNCS 5023, pp. 343–356, 2008.

The performance of a Broadcast Encryption scheme is generally measured in three parameters: bandwidth, space and time. Bandwidth is the size of the transmission *overhead* incurred by the scheme, space is the amount of storage for each user, and time is a measurement of the computation time needed for users to decrypt a message. In this paper, we will focus on the tradeoff between bandwidth and space.

In general, Broadcast Encryption schemes work by distributing a fresh *message key*, so that only the current members can recover the message key. The actual message is then encrypted under the message key and broadcast. This construction means that the bandwidth, i.e., the overhead incurred by the scheme, does not depend on the sizes of the messages the sender wants to transmit.

The problem of Broadcast Encryption was first described by Berkovits in [5], and later Fiat and Naor started a more formal study of the subject [7].

There are two naive schemes solving the broadcast encryption problem. In the first naive scheme, we give each user her own secret key shared with the sender. With this scheme, the space is 1 and the bandwidth is m, where m the number of members. In the second naive scheme, we assign a key to every possible subset of users, and give all users belonging to a subset access to the key for that subset. In this case the space is 2^{n-1}, where n is the number of users, and the bandwidth is 1.

In 2001, the *Subset Cover* framework was introduced by Naor *et al.* [16], along with two schemes, Complete Subtree and Subset Difference. In Subset Cover based schemes, there is a family of subsets of users, where each subset is associated with a key. When the sender wishes to make a broadcast, she finds a *cover* of the members using the subsets and encrypts the message key with each of the subset keys used in the cover. Both naive schemes can be seen as Subset Cover schemes; in the scheme with constant space, the family consists only of singleton subsets, and in the scheme with constant bandwidth, the family consists of all subsets of users. The Subset Cover principle is very general, and most published schemes are Subset Cover schemes.

In most Subset Cover schemes, each user is a member of a large number of subsets, so storing the key for each subset would be expensive in terms of memory. To solve this, the keys are chosen in such a way that users can compute the keys they should have using some smaller set of secrets and a *key derivation algorithm*. Schemes where keys are unrelated are called *information-theoretic*.

The most common key derivation algorithm is a straightforward application of a Pseudo-Random Sequence Generator (PRSG). The first protocol of this type was Subset Difference which has a bandwidth of $\min(2r-1, n-r)$ with a user space $s = \mathcal{O}(\log^2 n)$ where r is the number of revoked users. Many more schemes [11,4,10,13,12] have been proposed using the same kind of key derivation. They all have a bandwidth of $\mathcal{O}(r)$, the same as Subset Difference, and their improvements lie in that some of them have a space of $\mathcal{O}(\log n)$, some offer increased flexibility, and some improve the bandwidth to $c \cdot r$ for $c < 1$ (as opposed to $c = 2$ in the original scheme).

Other forms of key derivation, such as RSA accumulators [2,3,8] and bilinear pairings [6], have also been studied for Subset Cover based broadcast encryption.

There have been attempts to reduce bandwidth by modifying the problem, for instance by allowing some *free-riders* (non-members who can still decrypt the broadcast) [1] or relaxing the security requirements [14].

There has been some analysis of lower bounds for Broadcast Encryption schemes. In 1998 Luby and Staddon [15] showed $s \geq \left(\frac{\binom{n}{r}^{1/b}}{b} - 1 \right) / r$ for Broadcast Encryption without key derivation, using the Sunflower lemma. This bound was sharpened in 2006 by Gentry *et al.* [9] to $s \geq (\binom{n}{r}^{1/b} - 1)/r$. We remark that schemes using key derivation beat these bounds.

1.1 Our Contribution

Table 1. Upper and lower bounds for Subset Cover schemes

Key derivation	Lower bound	Assumption	Upper bound	Space
None	$r \frac{\log(n/r)}{\log(rs)}$	—	$r \log(n/r)$	$s = \mathcal{O}(\log n)$
PRSG, small r	$\Omega(r)$ (**our**)	$s \leq \text{poly}(n)$	$\mathcal{O}(r)$	$s = \mathcal{O}(\log n)$
PRSG, worst r	$\Omega\left(\frac{n}{\log s}\right)$ (**our**)	—	$\mathcal{O}\left(\frac{n}{\log s}\right)$	—
PRSG, large r	$n - r$ (**our**)	$r \geq n - \frac{n}{6s}$	$n - r$	$s = \mathcal{O}(1)$

We present lower bounds for a large class of Subset Cover schemes, including existing schemes based on PRSGs. These lower bounds match known constructions up to a small constant, showing that current PRSG-based schemes are essentially optimal. Table 1 gives a summary of our results.

Our bounds on the bandwidth usage are strong, and show that the early Subset Difference scheme is in fact very close to being optimal. For instance, our bound for small r shows that improving the bandwidth to $o(r)$ would require super-polynomial space, which is unreasonable. In fact, depending on the application, space is generally considered reasonable if it is at most logarithmic (or possibly polylogarithmic) in n.

Our second result implies that, in order to get constant bandwidth b, the space required is exponential. It also implies that, using polylogarithmic space, the worst case bandwidth will be almost linear, $n/\log\log n$.

The third result says that, for a small number of members, the first naive scheme is optimal. With polylogarithmic amount of space, this holds even if the number of members is almost linear, $n/\text{poly}\log n$.

Also, in most current schemes, the decryption time for members is limited to be polylogarithmic in n. Our proofs do not make use of any such restrictions, so allowing longer decryption time than current schemes cannot lower the bandwidth requirements.

1.2 Organization of This Paper

In Section 2 we discuss the structure of Subset Cover based Broadcast Encryption schemes and define the class of schemes, Unique Predecessor schemes, for which we prove lower bounds. In Section 3 we give a proof showing that with polynomial memory in clients, the bandwidth consumption is $\Omega(r)$ for "small" r. In Section 4, we prove a bound for generic r, and in particular show that for $r \approx \frac{n}{e}$, the bandwidth is at least $\frac{n}{1.89 \log s}$. Section 5 shows that for a large number of revoked users, the worst case bandwidth is $n - r$, i.e., the same as for the naive scheme where every user has a single key.

2 Preliminiaries

In this section, we review some preliminaries. The concepts of Broadcast Encryption and Subset Cover schemes are described, and notation will be introduced. We also define a class of Broadcast Encryption schemes called Unique Predecessor (UP) schemes to which our lower bounds apply.

2.1 Broadcast Encryption

In Broadcast Encryption, we have a trusted sender, and a set of users. After some initialization, the sender can securely broadcast messages to some subset of the users in a way which is efficient for both the sender and users. We will refer to the users who are targeted by a broadcast as *members* and the users who are not as *revoked* users. As the name Broadcast Encryption implies, we assume there is a single broadcast medium, so all users see the messages transmitted by the sender.

When evaluating the efficiency, three parameters are measured: bandwidth, space, and time (for decryption). Most Broadcast Encryption schemes transmit encrypted keys, so we will measure the bandwidth in terms of the number of encrypted keys to be transmitted. The sender uses the broadcast encryption to distribute a message key K_m and then encrypts the actual message under K_m, so the bandwidth overhead incurred does not depend on the size of the actual message.

The bandwidth required for a scheme with n users out of which r are revoked can, and generally will, vary, depending on which r users are revoked. We define the bandwidth $b = f(n, r)$ of the Subset Cover scheme as being that of the maximum bandwidth over the choice of the set of revoked users $R \subseteq [n]$ such that $|R| = r$. Thus, when we say that the bandwidth is at least $c_1 r$ for $r \leq n^{c_2}$, we mean that for every such r, there is at least one choice of r revoked users which requires bandwidth at least $c_1 r$.

Similarly, we measure the space as the number of keys, seeds, or other secrets that a user must store to be able to correctly decrypt transmissions she should be able to decrypt. It need not be the case that all users have to store the same amount of secrets, so we let the space of a scheme be the size of the largest amount of secrets any one user must store.

We remark that, in general, the keys and secrets may vary in length, so that our convention of simply counting the number of keys may not measure the exact bandwidth or space. However, such differences are generally small and not taking them into account costs us at most a small constant factor.

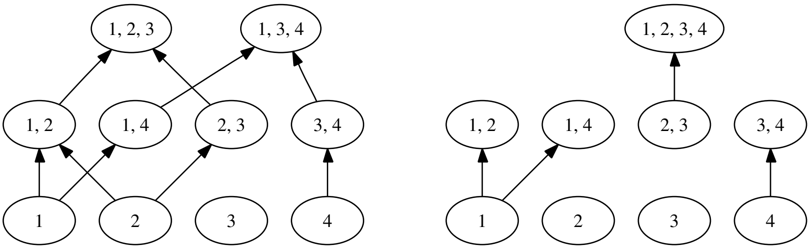
In this paper, we will not concern ourselves with the computational time of the clients. Our only assumption will be the very natural (and necessary) assumption that users cannot derive keys which they should not have access to.

Broadcast Encryption schemes can be classified as either stateful or stateless. In a stateful scheme, a transmission from the sender may update the set of secrets a user uses to decrypt future broadcasts, whereas in the stateless case, the secrets are given to the user at initialization and then remain constant. We focus on the largest family of Broadcast Encryption schemes, Subset Cover schemes, and such schemes are stateless.

2.2 Notation

Throughout the paper, we will use the following notation. We let n denote the total number of users, and identify the set of users with $[n] = \{1, \ldots, n\}$. We let m denote the number of members and r the number of revoked users (so $n = r + m$). The space of a scheme is denoted by s and the bandwidth by b. Note that we are generally interested in the bandwidth as a function of r (or equivalently, of m).

2.3 Subset Cover Schemes



(a) Example of a Subset Cover scheme

(b) Example of a Subset Cover scheme $\mathcal{S}$ with indegree 1

Fig. 1. Illustration of Subset Cover schemes

In this paper, we consider a family of Broadcast Encryption schemes known as Subset Cover schemes, introduced in [16]. In a Subset Cover scheme, the sender starts by creating a family of subsets of users. Each such subset is associated with a key. To make an encrypted broadcast, the sender first computes a *cover* of the current members. A cover is a choice of subsets from the family, so that all members belong to at least one chosen subset, and no revoked user belong to any chosen subset. The message broadcasted will then contain, for each subset in the cover, the message key encrypted under that subset's key.

Without key derivation, each user would have to store the key for each subset of which she is a member. However, when using key derivation, keys of subsets are related in a way that allows a user to derive keys of subsets by applying a suitable function, typically a one way function, to her set of secrets. Thus the space decreases, as one secret can be used to derive multiple keys.

Example 1. Figure 1(a) shows an example of a Subset Cover scheme on $n = 4$ users. In the example, the family of subsets consists of all four singleton subsets, four subsets of size 2, and two subsets of size 3. An edge from S_i to S_j indicates that the secrets used to derive the key for S_i can also be used to derive the key for S_j. Thus, the secret used by user 2 to derive the key for her singleton set $\{2\}$ can also be used to derive the keys for nodes $\{1, 2\}$, $\{2, 3\}$, and $\{1, 2, 3\}$. Without key derivation, she would have had to store four keys, but now she only needs to store one secret.

More formally, a Subset Cover scheme consists of a family of subsets $\mathcal{F} = \{S\} \subseteq 2^{[n]}$ with the property that for every selection of members $M \subseteq [n]$ there is a cover $T \subseteq \mathcal{F}$ such that $\cup_{S \in T} S = M$. There is a set of "secrets" $\mathcal{K}$, and each user $i \in [n]$ is given a subset $P(i)$ of these secrets. Additionally, there is, for each $S \in \mathcal{F}$, a set $K(S) \subseteq \mathcal{K}$ of secrets and a secret key $k(S)$, with the following properties:

- Any user with access to a secret in $K(S)$ can compute $k(S)$.
- For every $S \in \mathcal{F}$ and user $i \in [n]$, $P(i) \cap K(S) \neq \emptyset$ if and only if $i \in S$.
- An adversary with access to all secrets in $\mathcal{K} \backslash K(S)$ cannot compute $k(S)$

To send a message key to the set $M \subseteq [n]$ of members, the cover $T \subseteq \mathcal{F}$ of subsets is chosen in such a way that $\cup_{S \in T} S = M$. The server then broadcasts the message key encrypted using $k(S)$ for each $S \in T$. The bandwidth required for this is $|T|$. We remark that a Subset Cover scheme is required to be able to cover any member set $M \subseteq [n]$.

Naturally, a Subset Cover scheme should also include efficient ways of computing $k(S)$ and the cover T, but as we are interested in lower bounds on the tradeoff between space and bandwidth, these computational issues are not relevant to us.

We denote by $B(\mathcal{F})$ the partially ordered set on the elements of $\mathcal{F}$ in which $S_1 \leq S_2$ if $K(S_1) \subseteq K(S_2)$, i.e., if any secret that can be used to deduce $k(S_1)$ can also be used to deduce $k(S_2)$. Note that $S_1 \leq S_2$ implies $S_1 \subseteq S_2$ (since any user $u \in S_1$ will be able to compute $k(S_2)$ and thus has to be an element of S_2). From now on, we will ignore the set of secrets and the keys, and only study the poset $B(\mathcal{F})$, since it captures all information that we need for our lower bounds. In Figure 1 we show Hasse diagrams of $B(\mathcal{F})$ for two toy example Subset Cover schemes.

The number of secrets a user u needs to store, i.e., the space s, is precisely the number of elements S of $B(\mathcal{F})$ such that u occurs in S, but not in any of the predecessors of S.

Lemma 1. *Any Subset Cover scheme will have at least one singleton node for each user.*

Proof. If there is a user which does not occur in a singleton node, the Broadcast Encryption scheme would fail when the sender attempts to broadcast only to that user. □

2.4 Key Derivation Based on a PRSG

The most common type of key derivation uses a Pseudo-Random Sequence Generator (PRSG), or equivalently, a family of hash functions. This type of key derivation was first used in the context of Broadcast Encryption in the Subset Difference scheme [16]. In [4] it is called Sequential Key Derivation Pattern. The key derivation described here is the intuitive way to do key derivation using a PRSG, and all Subset Cover schemes that the authors are aware of that use a PRSG (or a family of hash functions) do have this form of key derivation.

Let ℓ be a security parameter and let $H(x)$ be a pseudo-random sequence generator taking as seed a string x of length ℓ. Let $H_0(x)$ denote the first ℓ bits of output when running $H(x)$, let $H_1(x)$ denote the next ℓ bits, and so on.

Each subset S in the scheme will be assigned a seed $p(S)$ and a key $k(S)$. The key $k(S)$ will be computed as $k(S) = H_0(p(S))$, so from the seed for a subset, one can always compute the key for that subset. All secrets given to users will be seeds, no user is ever given a key directly. The reason for this is that it gives an almost immediate proof of the security of the scheme by giving the keys the property of *key indistinguishability*, which was proved in [16] to be sufficient for the scheme to be secure in a model also defined in [16].

Consider an edge $e = (S_i, S_j)$ in the Hasse diagram of $B(\mathcal{F})$. The edge means that someone with access to the secrets to deduce $k(S_i)$, i.e. $p(S_i)$ should also be able to deduce $p(S_j)$. If we let $p(S_j) = H_c(p(S_i))$ for some $c \geq 1$, anyone with $p(S_i)$ can derive $p(S_j)$. For a node S_i with edges to $S_{j_1}, S_{j_2}, \ldots, S_{j_k}$ we let $p(S_{j_1}) = H_1(p(S_i)), p(S_{j_2}) = H_2(p(S_i)), \ldots p(S_{j_k}) = H_k(p(S_i))$.

This construction cannot support nodes with indegree greater than 1, since that would require $p(S_j) = H_{c_1}(p(S_{i_1})) = H_{c_2}(p(S_{i_2}))$, which, in general, we cannot hope to achieve. This means that the Hasse diagram will be a forest, since all nodes have an indegree of either 0 or 1.

2.5 UP-Schemes

When the Hasse diagram of $B(\mathcal{F})$ is a forest, we say that the Subset Cover scheme is a *Unique Predecessor scheme* (UP-scheme). Schemes using key derivation as described in Subsection 2.4 will always be UP-schemes. Schemes not using any key derivation (there are no edges in $B(\mathcal{F})$) are also UP-schemes, this class of schemes is sometimes referred to as information-theoretic.

Example 2. The scheme in Figure 1(a) is not a UP-scheme, since there are several sets which have multiple incoming edges, for instance the set $\{1, 2\}$. However, the scheme $\mathcal{S}$ in Figure 1(b) is a UP-scheme. In this case, user 1 would have to store two secrets, one for her singleton node, and one for the node $\{1, 2, 3, 4\}$. The keys for nodes $\{1, 2\}$ and $\{1, 4\}$ can be derived from the same secret used to derive the key for her singleton node.

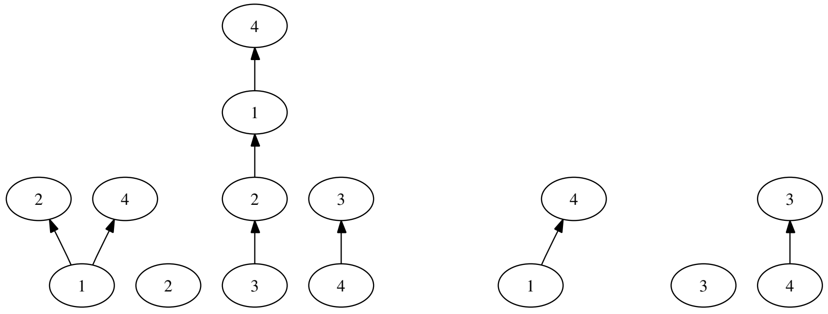
We view a UP-scheme as a rooted forest $\mathcal{S}$, in which each node $V \subseteq [n]$ is labelled with the set of users which are in V, but are not in the parent node. The number of node labels in which a user occurs is the same as the number of secrets that a user will need to store. Thus, when we say that a scheme $\mathcal{S}$ has space s we mean that every user can be used in a label at most s times.

Lemma 2. *Any Unique Predecessor scheme will have at most ns distinct subsets.*

Proof. Adding a new node to a Unique Predecessor scheme means increasing the space for at least one member. Starting from an "empty" scheme, this can be done at most ns times. □

2.6 Normalized UP-Schemes

To simplify the proofs, we will work with *normalized* UP-schemes. We will show that we can perform a simple normalization of a UP-scheme which gives a new scheme with the same set of users, no more space, and at most the same bandwidth. This normalization is similar to the construction of the Flexible SD scheme in [4].



(a) Normalization $\mathcal{S}'$ of the UP-scheme $\mathcal{S}$ in Figure 1(b)

(b) The subscheme $\mathcal{S}'(\{1, 3, 4\})$

Fig. 2. Normalization of UP-schemes

Definition 1. *A UP-scheme $\mathcal{S}$ is* normalized *when every node of $\mathcal{S}$ is labelled with exactly one user and $\mathcal{S}$ has exactly n trees.*

Example 3. The scheme $\mathcal{S}$ from Figure 1(b) is a UP-scheme, but it is not normalized. Two nodes violate the normalization criteria. First, the key for $\{1,2,3,4\}$ can be directly derived from the secrets used for $\{2,3\}$, which adds two new users at the same time. Second, the node $\{2,3\}$ also adds two users at once. In Figure 2(a) shows a normalized scheme $\mathcal{S}'$ which is essentially equivalent with $\mathcal{S}$. The key for $\{2,3\}$ can now be derived from the secret for $\{3\}$, and an extra node $\{1,2,3\}$ was inserted between $\{2,3\}$ and $\{1,2,3,4\}$.

Lemma 3. *Let $\mathcal{S}$ be an arbitrary UP-scheme (on n users) with space s and bandwidth b. Then there exists a normalized UP-scheme $\mathcal{S}'$ (on n users) with space $s' \leq s$ and bandwidth $b' \leq b$.*

Proof. The proof consists of two steps. First we ensure that each node is labelled with exactly one user. Second, we merge identical nodes, which will ensure that $\mathcal{S}'$ has exactly n trees.

Consider a node labelled with a set $U = \{u_1, \ldots, u_k\}$ of users with $k > 1$. Now, split this node into a chain of k nodes, adding one user at a time (in arbitrary order) rather than all k at once. Call the resulting forest $\mathcal{S}_0$. Note that, strictly speaking, it is possible that $\mathcal{S}_0$ is not a UP-scheme as we have defined it, since there may be several nodes representing the same subset S of users. However, it still makes sense to speak of the space and bandwidth of $\mathcal{S}_0$, and we note that the space of $\mathcal{S}_0$ is the same as that of $\mathcal{S}$, as each user occurs in the same number of labels in both. Furthermore, the bandwidth of $\mathcal{S}_0$ is no more than that of $\mathcal{S}$, since all subsets of users present in $\mathcal{S}$ are also present in $\mathcal{S}_0$, and thus, any cover in $\mathcal{S}$ is also valid in $\mathcal{S}_0$.

Next, we describe how to merge nodes representing the same set $S \subseteq [n]$. Given two such nodes v_1 and v_2, attach the children of v_2 as children to v_1, and remove v_2 from the scheme. Note that this operation does not change the bandwidth of the scheme, since only a single node is removed, and this node represents a subset which is still present in the resulting scheme. Also, the space for the resulting scheme will be no larger than the space for the original scheme. The user which was the label for v_2 will now need to store one secret less, whereas the space will be the same for all other users. Let $\mathcal{S}'$ be the result of applying this merging until every set is represented by at most one node.

It remains to show that $\mathcal{S}'$ has exactly n trees. By Lemma 1, there must be at least n trees in $\mathcal{S}'$. Because of the first step every root of a tree will represent a singleton set, and because of the second step every singleton set can be present at most once, implying that there are at most n trees. □

We will, without loss of generality, from now on assume UP-schemes we deal with are normalized. See Figure 2(a) for an example of a normalized UP-scheme. We would like to remark that while normalization can only improve bandwidth and space, it does so at the cost of time. Thus, when applied to improve the performance of practical schemes, one has to take into account the computation time of users, as discussed in [4].

We remark that, in general, normalization will introduce key derivation, even if the original UP-scheme had completely independent keys.

Definition 2. *Given a UP-scheme $\mathcal{S}$ and a set $X \subseteq [n]$ of users, the* subscheme of $\mathcal{S}$ induced by X*, denoted $\mathcal{S}(X)$, is defined as follows: for every user $y \notin X$, we remove all nodes of $\mathcal{S}$ labelled with y, and their subtrees.*

In other words, $\mathcal{S}(X)$ contains the nodes (and thus subsets) which are still usable when $[n] \setminus X$ have been revoked. See Figure 2(b) for an example.

3 Few Revoked Users

We prove that when the number of revoked users r is small, any UP-scheme using at most polynomial space will require bandwidth $\Omega(r)$.

As noted in the introduction, the requirement that the space is polynomial is *very* generous. Anything beyond polylogarithmic space per user is generally considered impractical.

Theorem 1. *Let $c \geq 0$ and $0 \leq \delta < 1$. Then, any UP-scheme with n users and space $s \leq n^c$ will, when the number of revoked users $r \leq n^\delta$, require bandwidth*

$$b \geq \frac{1-\delta}{c+1} \cdot r \tag{1}$$

Proof. Let $\mathcal{S}$ be an arbitrary UP-scheme with $s \leq n^c$ and let $r \leq n^\delta$. An upper bound on the number t of sets of users that can be handled using bandwidth at most b is given by the number of sets of nodes of $\mathcal{S}$ of cardinality at most b. Since $\mathcal{S}$ contains at most ns nodes, this is upper-bounded by

$$\sum_{i=1}^{b} \binom{ns}{i} \leq (ns)^b \leq n^{(c+1)b} \tag{2}$$

In order for $\mathcal{S}$ to be able to handle every set of revoked users of size r, we need t to be at least $\binom{n}{r}$, giving

$$n^{(c+1)b} \geq \binom{n}{r} \geq (n/r)^r \geq n^{(1-\delta)r} \tag{3}$$

and the theorem follows. □

Theorem 1 comes very close to matching many of the previous works, for instance Subset Difference [16] with $s = \mathcal{O}(\log^2 n)$ and $b = \min(2r-1, n-r)$. For $r \leq \sqrt{n}$, our bound gives $b \geq \frac{r}{2(1+c)}$ which is within a factor $4 + o(1)$.

As mentioned in the introduction, [9] has shown a stronger bound, roughly $r\frac{\log(n/r)}{\log(rs)}$, using the Sunflower lemma. However, their bound applies only to Subset Cover schemes without key derivation, and is in fact stronger than existing schemes using key derivation – e.g. the Subset Difference scheme mentioned above for $r < n^{1/3}$.

4 Arbitrarily Many Revoked Users

In this section we show that, for a certain choice of r, any UP-scheme has to use bandwidth at least $\frac{n}{1.89 \log s}$. We start with Theorem 2, which gives a lower bound on the bandwidth as a function of m/n. Plugging in a suitable value of m/n in Corollary 1 will then give the desired result.

Theorem 2. *Let $\delta \in (0,1]$ and $\epsilon > 0$. Then for every UP-scheme $\mathcal{S}$ with $n > \frac{2\delta(1-\delta)}{\epsilon^2}$ there exists a set of users M of size $\delta - 3\epsilon \leq |M|/n \leq \delta + \epsilon$ which requires bandwidth $b \geq |M| \frac{\log(1/\delta)}{\log(s/\epsilon)}$*

Proof. Pick $M_0 \subseteq [n]$ randomly where every element is chosen with probability δ, independently.

Set $d = \log_\delta(\epsilon/s)$ and let X be the set of users which occur at depth exactly d in $\mathcal{S}(M_0)$ (where the roots are considered to be at depth 1). Let $M = M_0 \setminus X$. Since each node can cover at most d users of M, the bandwidth required for M is at least

$$\frac{|M|}{d} = |M| \frac{\log(1/\delta)}{\log(s/\epsilon)}$$

It remains to show that there is a positive probability (over the random choice of M_0) that M ends up having the required size, as this implies that such an M exists.

The probability that a node at depth d of $\mathcal{S}$ remains in $\mathcal{S}(M_0)$ is $\delta^d = \epsilon/s$. The total number of nodes at depth d in $\mathcal{S}$ is upper-bounded by ns, and thus, the expected number of nodes at depth d in $\mathcal{S}(M_0)$, i.e. the expected size of X, is at most $\delta^d ns = \epsilon n$. By Chebyshev's inequality, we have $\Pr\left[\left|\frac{|M_0|}{n} - \delta\right| \geq \epsilon\right] \leq \frac{\delta(1-\delta)}{n\epsilon^2} < 1/2$. By Markov's inequality, we have $\Pr\left[\frac{|X|}{n} \geq 2\epsilon\right] \leq 1/2$. The union bound then gives that $\Pr[\delta - 3\epsilon \leq |M|/n \leq \delta + \epsilon] > 0$. Thus, there exists some choice of M_0 such that $|M|$ falls within this range. □

As a corollary, we have:

Corollary 1. *For any $\epsilon > 0$ there exist n_0 and s_0 such that any UP-scheme $\mathcal{S}$ with $n \geq n_0$ and $s \geq s_0$ uses bandwidth at least*

$$\frac{n}{(e \ln(2) + \epsilon) \log_2(s)} \approx \frac{n}{1.89 \log_2(s)} \tag{4}$$

Proof. Let $\delta = 1/e$. Invoking Theorem 2 with parameters δ and ϵ' (the value of which will be addressed momentarily), we get a set M of size at least $(\delta - 3\epsilon')n$ requiring bandwidth at least

$$n\frac{\delta - 3\epsilon'}{\ln(s/\epsilon')} = n\frac{1 - 3e\epsilon'}{e\ln(2)\log_2(s) + e\ln(1/\epsilon')} \tag{5}$$

Pick ϵ' small enough so that

$$\frac{e\ln(2)}{1 - 3e\epsilon'} \leq e\ln(2) + \epsilon/2.$$

Then Equation (5) is lower-bounded by Equation (4) for any s satisfying

$$\frac{e\ln(1/\epsilon')}{1-3e\epsilon'} \leq \frac{\epsilon}{2}\log_2(s)$$
$$\log_2(s) \geq \frac{2e\ln(1/\epsilon')}{\epsilon(1-3e\epsilon')},$$

and we are done. □

We remark that Corollary 1 is tight up to the small constant 1.89, as seen by the following theorem.

Theorem 3. *There exists a UP-scheme $\mathcal{S}$ using bandwidth at most $\left\lceil \frac{n}{\log_2(s)} \right\rceil$.*

Proof. Partition the users into $\lceil n/\log_2(s) \rceil$ blocks of size $\leq \log_2(s)$. Then, in each block, use the naive scheme with exponential space and bandwidth 1, independently of the other blocks. □

5 Bandwidth is $n - r$ for Large r

We show that when the number of revoked users gets very large, all UP-schemes will have a bandwith of $n-r$, e.g. one encryption per member. Exactly how large r has to be for this bound to apply depends on s. This is the same bandwidth as is achieved by the naive solution of just giving each user her own private key.

Theorem 4. *For any UP-scheme $\mathcal{S}$ and $m \leq \frac{n}{6s}$, there is a member set M of size $|M| = m$ requiring bandwidth $b = |M|$.*

Proof. We will build a sequence $M_0 \subseteq M_1 \subseteq M_2 \subseteq \ldots \subseteq M_m$ of sets of members with the properties that $|M_i| = i$, and that the bandwidth required for M_i is i.

The initial set M_0 is the empty set. To construct M_{i+1} from M_i, we pick a user $u \notin M_i$ satisfying:

- There is no $v \in M_i$ such that some node labelled with u occurs as the parent of some node labelled with v
- There is no $v \in M_i$ such that the root node labelled with v occurs as the parent of some node labelled with u
- The root node labelled u has outdegree $\leq 2s$

We then set $M_{i+1} = M_i \cup \{u\}$. Clearly $|M_i| = i$, so there are two claims which remain to be proved. First, that the required bandwidth of M_i is i. Second, that the process can be repeated at least m times.

To compute the bandwidth of M_i, we prove that the only way to cover M_i is to pick the singleton sets of every $u \in M_i$. To see this, assume for contradiction that there exists some set S with $|S| > 1$ that can be used when constructing a cover. This corresponds to a node x at depth $|S|$ of some tree, and S is given by the labels of all nodes from x up to the root. In order for us to be able to use

S when constructing a cover, all these nodes need to belong to M_i. However, the first two criterions in the selection of u above guarantee that not all of these nodes can belong to M_i. The first criterion states that, once we have added a node, we can never add its parent. The second criterion states that, once we have added a root, we can never add any of its children. This shows that there can be no such S.

To see how many steps the process can be repeated, let r_i be the total number of nodes which are "disqualified" after having constructed M_i. Then, M_{i+1} can be constructed if and only if $r_i < n$. First, r_0 equals the number of roots which have degree $> 2s$. Since the total number of nodes is at most ns, this number is at most $r_0 \leq n/2$. When going from M_i to M_{i+1}, the total number of new disqualified nodes can be at most $3s$ – the node added, the parents of the at most $s-1$ non-root occurrences of u, and the at most $2s$ children of the root labelled with u. Thus, we have that $r_i \leq n/2 + 3si$, which is less then n if $i < \frac{n}{6s}$. □

The lower bound of Theorem 4 is tight up to a small constant in the following sense.

Theorem 5. *There exists a UP-scheme $\mathcal{S}$ such that for any set M of $|M| > \lceil \frac{n}{s} \rceil$ members, the bandwidth is $b < |M|$.*

Proof. Partition the set of users into $B = \lceil \frac{n}{s} \rceil$ blocks of size $\leq s$, and let each user share a key with each of the $s-1$ other users in her block. Then, given a set M of size $|M| > B$, there must be two users $i, j \in M$ belonging to the same block. Using the key shared by i and j to cover both them both, we see that the bandwidth of M is at most $b \leq |M| - 1$. □

6 Conclusion

In this paper, we have shown lower bounds for a large class of Subset Cover based Broadcast Encryption schemes. This type of scheme is probably the most explored class of schemes today, with many constructions. Our proofs are in a model with very relaxed constraints compared to what is considered practical, so it would not help to simply relax requirements slightly (e.g. allowing more space or time). The lower bounds shown in this paper match known constructions very well.

In particular, our bounds show that it will be impossible to get a bandwidth of $o(r)$ without increasing the space requirements to unreasonable levels or using some new form of key derivation. We do not have any lower bounds on the memory needed for $\mathcal{O}(r)$ bandwidth, an open question is thus if it is possible to get $\mathcal{O}(r)$ bandwidth with space $o(\log n)$.

Acknowledgements. The authors are grateful to Johan Håstad for many useful comments and discussions.

References

1. Adelsbach, A., Greveler, U.: A broadcast encryption scheme with free-riders but unconditional security. In: Safavi-Naini, R., Yung, M. (eds.) DRMTICS 2005. LNCS, vol. 3919, pp. 246–257. Springer, Heidelberg (2006)
2. Asano, T.: A revocation scheme with minimal storage at receivers. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 433–450. Springer, Heidelberg (2002)
3. Asano, T.: Reducing Storage at Receivers in SD and LSD Broadcast Encryption Schemes. In: Chae, K.-J., Yung, M. (eds.) WISA 2003. LNCS, vol. 2908, pp. 317–332. Springer, Heidelberg (2004)
4. Attrapadung, N., Kobara, K., Imai, H.: Sequential key derivation patterns for broadcast encryption and key predistribution schemes. In: Laih, C.S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 374–391. Springer, Heidelberg (2003)
5. Berkovits, S.: How to broadcast a secret. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 535–541. Springer, Heidelberg (1991)
6. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
7. Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)
8. Gentry, C., Ramzan, Z.: RSA accumulator based broadcast encryption. In: Zhang, K., Zheng, Y. (eds.) ISC 2004. LNCS, vol. 3225, pp. 73–86. Springer, Heidelberg (2004)
9. Gentry, C., Ramzan, Z., Woodruff, D.P.: Explicit exclusive set systems with applications to broadcast encryption. In: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), pp. 27–38. IEEE Computer Society, Washington (2006)
10. Goodrich, M.T., Sun, J.Z., Tamassia, R.: Efficient tree-based revocation in groups of low-state devices. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 511–527. Springer, Heidelberg (2004)
11. Halevy, D., Shamir, A.: The LSD broadcast encryption scheme. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 47–60. Springer, Heidelberg (2002)
12. Hwang, J.Y., Lee, D.H., Lim, J.: Generic transformation for scalable broadcast encryption schemes. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 276–292. Springer, Heidelberg (2005)
13. Jho, N.S., Hwang, J.Y., Cheon, J.H., Kim, M.H., Lee, D.H., Yoo, E.S.: One-way chain based broadcast encryption schemes. In: Cramer, R.J.F. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 559–574. Springer, Heidelberg (2005)
14. Johansson, M., Kreitz, G., Lindholm, F.: Stateful subset cover. In: Zhou, J., Yung, M., Bao, F. (eds.) ACNS 2006. LNCS, vol. 3989, pp. 178–193. Springer, Heidelberg (2006)
15. Luby, M., Staddon, J.: Combinatorial bounds for broadcast encryption. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 512–526. Springer, Heidelberg (1998)
16. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001)