An Analysis of the Naor-Naor-Lotspiech Subset Difference Algorithm (For Possibly Incomplete Binary Trees)

Sanjay Bhattacherjee, Palash Sarkar

An Analysis of the Naor-Naor-Lotspiech Subset Difference Algorithm (For Possibly Incomplete Binary Trees)

Sanjay Bhattacherjee and Palash Sarkar

Applied Statistics Unit, Indian Statistical Institute
203, B.T.Road, Kolkata, India - 700108.
sanjayb_r@isical.ac.in, palash@isical.ac.in

Abstract. The Subset Difference (SD) method is the most popular of Broadcast Encryption schemes due to its use in AACS standard for video discs. The scheme assumes the number of users n to be a power of two. In this paper, we relax this and consider arbitrary values of n. In some applications, this leads to substantial savings in the transmission overhead. Our analysis consists of the following aspects: (1) A recurrence to count $N(n, r, h)$ - the number of revocation patterns for arbitrary values of n and r (number of revoked users) resulting in a header length of h. The recurrence allows us to generate data and hence to completely analyze it for larger n than the brute force method. (2) We do a probabilistic analysis of the subset cover finding algorithm of the SD method and find an expression to evaluate the expected header length $E[X_{n,r}]$ for arbitrary values of n and r. Using this, $E[X_{n,r}]$ can be evaluated in $O(r \log n)$ time using constant space. (3) While concluding, we suggest a similar method for finding $E[X_{n,r}]$ for the Layered Subset Difference (LSD) scheme of Halevy and Shamir. (4) In the SD method, for n being a power two, we find asymptotic values of the expected header length.

Keywords: Broadcast encryption, subset difference, recurrence, expected header length, asymptotic analysis, layered subset difference.

1 Introduction

Broadcast Encryption (BE) is a cryptographic method for a center to efficiently broadcast digital contents to a set of users so that only an intended subset of the users (called privileged users) can access the contents. BE has a wide range of applications in the implementation of digital rights management [DRM] such as Internet or mobile video broadcast, pay-TV or even digital media like CD or DVD to name a few. In a BE scenario, the center pre-distributes key decrypting keys to users during initialization. After initialization is complete, when a data message has to be broadcast to the privileged users, it is encrypted using some session key. This session key in turn is encrypted several times using the key encrypting keys for different sets of privileged users. These encryptions of the

session key are sent along with the encrypted data as the header. Only privileged users will be able to retrieve the session key (using their own pre-distributed keys) from the header. Using the session key, the actual data that had been broadcast can be retrieved. Even if all the other (revoked) users collude, they should not be able to retrieve the session key.

Broadcast Encryption was introduced in [Ber91] followed by [FN93]. Further works on BE were [Sti97], [PGMM03] and [AKI03]. Traitor tracing, an important aspect of BE has been studied in [CFN94], [NP98], etc. The subset cover framework was introduced in [NNL01]. The tree-based subset difference method, which is the scheme that we have primarily analyzed in this paper, was also introduced in [NNL01]. Asymptotic improvements to this scheme were suggested in the LSD scheme of [HS02], which is also analyzed as part of our work. The SD method is the most popular BE scheme due to its use in the Advanced Access Content System (AACS) [AAC] standard used for content protection in video-discs. [LS98] and [PGM04] suggest bounds and tradeoffs between different parameters of BE. One of the efficiency parameters is the transmission overhead of a broadcast. It is determined by the *header length* (the number of encryptions of the session key in the header). Another work on BE is [JHC+05] in which the worst case header length has been brought down to r.

The SD scheme in [NNL01] and all the subsequent papers like [PB06] assume the number of users to be of the form 2^t while, for practical implementations of the scheme, the number of users can be arbitrary. In Section 4.1 of this paper we show that, if we do away with this assumption it will lead to significant saving on the transmission overhead. The following are our main contributions:

1. For the SD scheme, we find a recurrence $N(n, r, h)$ to count the number of revocation patterns for any given arbitrary n and r resulting in a header length of h. Previous to this work, there was no efficient way to generate this count exhaustively for large n other than the brute force method. For the special case when n is a power of two, we derive from it a generating function for the values of $N(n, r, h)$. It is similar to the generating function stated in [PB06].
2. For arbitrary n and r, we obtain an expression and hence an algorithm to find the expected header length $E[X_{n,r}]$ for the SD scheme where $X_{n,r}$ is a random variable taking its values from the set of possible header lengths. The algorithm to evaluate $E[X_{n,r}]$ takes $O(r \log n)$ time using constant space. Using this algorithm we show that, using the true value of n (instead of rounding it off to a power of two as suggested in the SD scheme) leads to remarkable improvements in the expected header length.
3. Following the same method, we suggest a similar method for computing $E[X_{n,r}]$ for the Layered Subset Difference (LSD) scheme of [HS02].
4. For n of the form 2^t, we find expressions to perform asymptotic analysis of $E[X_{n,r}]$. This explains the empirical observation in [NNL01] that the expected header length is upper bounded by $1.25r$. Our analysis is the first to provide theoretical support to the experimental observations of [NNL01].

Theorem 10 of [PB06] gives a method to find $E[X_{n,r}]$ for n of the power of two. They report their expressions as "complex to compute and hence difficult to get insight from". On the other hand, we see that the complexity of their algorithm is $O(r \log(n-r))$ which for $r \leq \frac{n}{2}$ is asymptotically same as the complexity of our algorithm (since in practice, r is much smaller than n). Our algorithm is much more appropriate for practical applications where n is arbitrary and it is easily computable. Both these methods use constant space.

2 The Subset Difference Method

The subset-cover revocation framework proposed in [NNL01], assumes that a broadcasting system has three parts: (1) an initiation scheme, (2) the broadcast algorithm and (3) the decryption algorithm. Let $\mathcal{N}$ be the set of all users and let $|\mathcal{N}| = n$; $\mathcal{R}$ is the set of revoked users and let $|\mathcal{R}| = r$. The initiation scheme defines a collection of subsets of users $S = \{S_1, S_2, \ldots, S_w\}$ where each $S_j \subseteq \mathcal{N}$. Each subset S_j is assigned a long-lived key L_j; every member u of S_j should be able to deduce L_j as a part of the decryption algorithm, from the secret information it receives during initiation from the center.

During broadcast, given a revoked set $\mathcal{R}$, the broadcast algorithm is used to partition the remaining users $\mathcal{N} \setminus \mathcal{R}$ into h disjoint subsets $S_{i_1}, S_{i_2}, \ldots, S_{i_h}$ so that $\mathcal{N} \setminus \mathcal{R} = \bigcup_{k=1}^{h} S_{i_k}$. This sub-collection of disjoint subsets is called the *cover* S_c for $\mathcal{N} \setminus \mathcal{R}$ ($S_c = \{S_{i_1}, S_{i_2}, \ldots, S_{i_h}\}$). A uniform random session key K is used to encrypt the broadcast message M by some symmetric key encryption algorithm. K is then encrypted h times with $L_{i_1}, L_{i_2}, \ldots, L_{i_h}$ and sent as the header with encrypted M. A privileged user, upon receiving the above broadcast, uses the decryption algorithm to deduce some L_j using its own secret information, and hence decrypt the encrypted session key K. This session key will further be used to decrypt the broadcast data.

For the Subset Difference method, the Subset-Cover Revocation Framework is specified as follows: In the initiation scheme that defines the collection S_{SD} of subsets, the users in $\mathcal{N}$ are viewed as the leaves of a complete binary tree $\mathcal{T}^0$ rooted at node 0. All internal nodes in $\mathcal{T}^0$ are numbered on a top-to-bottom (level-wise) and then left-to-right (at each level) basis. Hence every internal node i (root of subtree $\mathcal{T}^i$) represents a group of users that are leaves of $\mathcal{T}^i$. For each internal node i and a successor node j of i in $\mathcal{T}^0$, a subset $S_{i,j}$ is included into the collection S_{SD} of subsets. Such a subset $S_{i,j}$ has every user that is a successor of the node i but is not a successor of node j (those leaves which are in the subtree $\mathcal{T}^i$ but not in $\mathcal{T}^j$). In other words, $S_{i,j}$ corresponds to the users in $\mathcal{T}^i \setminus \mathcal{T}^j$. A subset corresponding to all users (represented by the full binary tree $\mathcal{T}^0$) is added to S_{SD}.

During broadcast, the center starts with the set $\mathcal{R}$ of revoked users. To find the cover S_c, the center first finds the Steiner tree $\mathcal{ST}(\mathcal{R})$ i.e.; the minimal subtree of the full binary tree $\mathcal{T}^0$, all of whose leaves are in $\mathcal{R}$. $\mathcal{T}'$, a copy of $\mathcal{ST}(\mathcal{R})$ is created and the following algorithm is applied to find the cover S_c:

1. Find leaf nodes i and j of $\mathcal{T}'$ such that the subtree rooted at the least-common-ancestor v of these two nodes does not contain any other leaf of $\mathcal{T}'$.
2. Let l and k be the two children of v such that i is a descendant of (or the same node as) l and j is a descendant of (or the same node as) k. If $l \neq i$, add the subset $S_{l,i}$ to the cover S_c. Similarly, if $k \neq j$, add the subset $S_{k,j}$ to the cover S_c.
3. Delete the subtree rooted at v in $\mathcal{T}'$. Hence v becomes a leaf node in $\mathcal{T}'$.
4. Repeat the above three steps iteratively until there is only one leaf remaining. For the last leaf node z, add the set $S_{0,z}$ to the cover S_c.

The cardinality of S_c is called the header length and determines the transmission overhead.

Note 1. The original algorithm in [NNL01] and all subsequent analysis [PB06] [AK08] [MMW09] consider n to be a power of two. In contrast, we work with arbitrary n.

Later, we argue that this can lead to substantial reduction in the transmission overhead.

3 Counting Revocation Patterns

A fixed permutation of the users where some are revoked and others are privileged is called *a revocation pattern*. The algorithm to find the subset cover takes as an input a revocation pattern and outputs the subset cover. The number of subsets h in the subset cover is called *the header length*.

$N(n, r, h)$ is defined as the number of revocation patterns with r revoked users (out of total n users) that are covered by a header of length exactly h.

$T(n, r, h)$ is defined as the number of revocation patterns with r revoked users (out of total n users) that are covered by a header of length h such that there is at least one revoked user in each of the two subtrees of the root node.

For an arbitrary n ($2^{t_0} < n \leq 2^{t_0+1}$ for some t_0), $T(n, r, h)$ and $N(n, r, h)$ are given by the following recurrences:

$$T(n, r, h) = \sum_{r_1=1}^{r-1} \sum_{h_1=0}^{h} N(2^{t_0}, r_1, h_1) \times N(n - 2^{t_0}, r - r_1, h - h_1) \tag{1}$$

$$\begin{aligned} N(n, r, h) = T(n, r, h) &+ \sum_{\ell=1}^{t_0} \left(q_\ell \times T(2^\ell, r, h-1) \right) \\ &+ \sum_{\ell=1}^{t_0} \left(\left\lfloor \frac{\rho_\ell - 1}{2^{\ell-1}} \right\rfloor \times T(\rho_\ell, r, h-1) \right) \end{aligned} \tag{2}$$

where $q_\ell = \lfloor \frac{n}{2^\ell} \rfloor$ and $\rho_\ell = n - (q_\ell \times 2^\ell)$. The base cases to the recurrences $T(n, r, h)$ and $N(n, r, h)$ are as follows:

$T(n,r,h)$	$r<0$	$r=0$	$r=1$	$2 \le r < n$	$r=n$	$r>n$
$h<0$	0	0	0	0	0	0
$h=0$	0	0	0	0	1	0
$h \ge 1$	0	0	0	from (1)	0	0
$N(n,r,h)$	$r<0$	$r=0$	$r=1$	$2 \le r < n$	$r=n$	$r>n$
$h<0$	0	0	0	0	0	0
$h=0$	0	0	0	0	1	0
$h=1$	0	1	n	from (2)	0	0
$h>1$	0	0	0	from (2)	0	0

(3)

A dynamic programming based algorithm to count $N(n, r, h)$ follows from the recurrence relations (1) and (2). We omit the description of this algorithm and the proof of correctness of the recurrence relations due to space constraints. For a given n_i, r_i and h_i, after the smaller values have been computed, the computation complexity of $N(n_i, r_i, h_i)$ and $T(n_i, r_i, h_i)$ are $O(\log n)$ and $O(rh)$ respectively. Hence, the overall complexity is $O(n \log nrh + nr^2h^2)$. This algorithm will require $O(nrh)$ space. This is a significant improvement over the brute force method which involves enumeration of the $\binom{n}{r}$ revocation patterns, and hence cannot be used for large values of n and r. Moreover, if the complete data for all r and h is needed for a given n, the subset cover finding algorithm has to be run for all the 2^n revocation patterns. The following table contains results of running the dynamic programming based algorithm for some values of n, r and h.

n	r	h	$N(n,r,h)$	n	r	h	$N(n,r,h)$
100	49	30	1.37×10^{28}	115	50	36	1.12×10^{32}
110	49	38	1.06×10^{30}	117	60	37	1.00×10^{33}

(4)

From Recurrences (1) and (2), we find the generating function for the sequence $N(n, r, h)$ *when the number of users is* $n = 2^m$. We omit the proof here due to lack of space.

Theorem 1 *The generating function for the sequence* $N_m(r, h)$ *of numbers is given by* $X_m(x, y)$ *where*

$$X_m(x,y) = \left(X_{m-1}(x,y) - xy^{2^{m-1}}\right)^2 + xy^{2^m} + 2^m x^2 y^{2^m - 1} + \sum_{i=1}^{m-1} \left(2^{m-i} xy^{2^m - 2^i} \times \left(X_{i-1}(x,y) - xy^{2^{i-1}}\right)^2\right). \quad (5)$$

A similar generating function was found by Park and Blake in [PB06]. It was derived based on the structural properties of the tree $\mathcal{T}^0$ used in creating the subsets of the subset difference method. We have taken a different approach of

first finding the recurrence relations for the sequence $N(n, r, h)$ and then have derived the generating function from it. Our generating function is of a slightly different form but gives rise to the same sequence.

4 Expected Header Length For Any n And r

For an arbitrary number of users n, and a fixed r, we find the expected header length. This is done by finding the probability that a node in $\mathcal{T}^0$ contributes a set to the subset cover (and hence the header). Summing up the probabilities for each node in the tree, we evaluate the expected header length. *We consider the random experiment where r out of the n initially un-revoked leaves of the tree $\mathcal{T}^0$ are chosen one-by-one uniformly at random without replacement and revoked.* This gives rise to a revocation pattern and hence a corresponding subset cover S_c and its header length h. Let $X_{n,r}$ be the random variable taking the value of the header length (cover size). Let $X^t_{n,r} \in \{0, 1\}$ be a random variable associated with node t of $\mathcal{T}^0$. $X^t_{n,r} = 1$ denotes the event that the cover contains a subset of the form $\mathcal{T}^t \setminus \mathcal{T}^{t'}$ where t' is some node in the subtree $\mathcal{T}^t$. Similarly, $X^t_{n,r} = 0$ denotes the event that there is no subset in the cover of the form $\mathcal{T}^t \setminus \mathcal{T}^{t'}$. If node t is the k^{th} node of level ℓ of $\mathcal{T}^0$, then $X^t_{n,r}$ will also be written as $X^{\ell,k}_{n,r}$. There are $n - 1$ internal nodes in $\mathcal{T}^0$. Then for a revocation pattern, we can write: $X_{n,r} = X^0_{n,r} + X^1_{n,r} + \ldots + X^{n-2}_{n,r}$. By linearity of expectation, we can write:

$$E[X_{n,r}] = E[X^0_{n,r}] + E[X^1_{n,r}] + \ldots + E[X^{n-2}_{n,r}]. \tag{6}$$

Since all the random variables $X^t_{n,r}$ follow Bernoulli distribution with probability $\Pr[X^t_{n,r} = 1]$, we get:

$$E[X_{n,r}] = \Pr[X^0_{n,r} = 1] + \Pr[X^1_{n,r} = 1] + \ldots + \Pr[X^{n-2}_{n,r} = 1]. \tag{7}$$

So the problem boils down to finding $\Pr[X^t_{n,r} = 1]$ for a given internal node t.

Here, we define $\eta_r(i, j)$ as the probability of choosing r elements from a set of i elements such that some fixed j elements are not chosen. So, if $j \geq i - r + 1$, then $\eta_r(i, j) = 0$ and for $0 < j < i - r + 1$,

$$\eta_r(i,j) = \left(1 - \frac{j}{i}\right)\left(1 - \frac{j}{i-1}\right)\left(1 - \frac{j}{i-2}\right)\cdots\left(1 - \frac{j}{i-r+1}\right) = \frac{(i-j)_r}{(i)_r}. \tag{8}$$

For an internal node t (the k^{th} node of level ℓ), let p be the predecessor and let s be the other child node of p (the sibling of node t). For a subset of the form $\mathcal{T}^t \setminus \mathcal{T}^{t'}$ to occur in the cover (where t' is some node in the subtree $\mathcal{T}^t$), there should be at least one revoked user in exactly one of the subtrees of the node t while there should be no revoked user in the other subtree of t. The subtree rooted at s should also have at least one revoked user. Let $R^{\ell,k}_{left}$ ($R^{\ell,k}_{right}$) be the random variable taking up values to indicate the number of revoked users in the

left (right) subtree of $\mathcal{T}^t$. Also, let $R^{\ell,k}_{sibling}$ be the random variable taking up values to indicate the number of revoked users in $\mathcal{T}^s$. Then we have the following lemma:

Lemma 2 *When $r > 0$, the probability that the cover S_c contains a set of the form $\mathcal{T}^t \setminus \mathcal{T}^{t'}$ is given by $\Pr[X^{\ell,k}_{n,r} = 1]$ where*

$$\begin{aligned}\Pr[X^{\ell,k}_{n,r} = 1] = \Pr[R^{\ell,k}_{sibling} > 0 \wedge R^{\ell,k}_{right} > 0 \wedge R^{\ell,k}_{left} = 0] \\ + \Pr[R^{\ell,k}_{sibling} > 0 \wedge R^{\ell,k}_{left} > 0 \wedge R^{\ell,k}_{right} = 0].\end{aligned} \tag{9}$$

Let A be the event $R^{\ell,k}_{sibling} > 0$. Similarly, let B and C be the events $R^{\ell,k}_{right} > 0$ and $R^{\ell,k}_{left} > 0$ respectively. Then, $\Pr[X^{\ell,k}_{n,r} = 1] = \Pr[A \wedge B \wedge \overline{C}] + \Pr[A \wedge \overline{B} \wedge C]$. For the moment, assume $\Pr[\overline{C}] > 0$ and $\Pr[\overline{B} \wedge \overline{C}] > 0$. Then,

$$\begin{aligned}\Pr[A \wedge B \wedge \overline{C}] &= (1 - \Pr[\overline{A} \vee \overline{B} | \overline{C}]) \times \Pr[\overline{C}] \\ &= \Pr[\overline{C}] - \Pr[\overline{A} \wedge \overline{C}] - \Pr[\overline{B} \wedge \overline{C}] + \Pr[\overline{A} \wedge \overline{B} \wedge \overline{C}].\end{aligned} \tag{10}$$

It can be verified that (10) holds even if $\Pr[\overline{C}] = 0$. Similarly, $\Pr[A \wedge \overline{B} \wedge C] = \Pr[\overline{B}] - \Pr[\overline{A} \wedge \overline{B}] - \Pr[\overline{B} \wedge \overline{C}] + \Pr[\overline{A} \wedge \overline{B} \wedge \overline{C}]$. Hence,

$$\begin{aligned}\Pr[X^{\ell,k}_{n,r} = 1] = \Pr[\overline{B}] + \Pr[\overline{C}] - \Pr[\overline{A} \wedge \overline{B}] - \Pr[\overline{A} \wedge \overline{C}] - 2\Pr[\overline{B} \wedge \overline{C}] \\ + 2\Pr[\overline{A} \wedge \overline{B} \wedge \overline{C}]\end{aligned} \tag{11}$$

As a result, we get the following theorem:

Theorem 3 *Let n_ℓ and n_r be the number of leaf nodes in the left and right subtrees respectively of the k^{th} node of level ℓ and let n_s be the number of leaf nodes in its sibling subtree $\mathcal{T}^s$. Then,*

$$\begin{aligned}\Pr[X^{\ell,k}_{n,r} = 1] = \eta_r(n, n_\ell) + \eta_r(n, n_r) - \eta_r(n, n_s + n_\ell) - \eta_r(n, n_s + n_r) \\ - 2\eta_r(n, n_\ell + n_r) + 2\eta_r(n, n_s + n_\ell + n_r)\end{aligned} \tag{12}$$

Equation (7) can be rewritten as follows:

$$E[X_{n,r}] = \sum_{\ell=1}^{t_0+1} \sum_{k=1}^{q_\ell} \Pr[X^{\ell,k}_{n,r} = 1] + \sum_{\ell=1}^{t_0} \left\lfloor \frac{\rho_\ell - 1}{2^{\ell-1}} \right\rfloor \times \Pr[X^{\ell,q_\ell+1}_{n,r} = 1]. \tag{13}$$

Here, level $t_0 + 1$ has only one node (the root node) for which $\Pr[X^0_{n,r} = 1] = \eta_r(n, 2^{t_0}) + \eta_r(n, n - 2^{t_0})$ which is a special case of (12). Computing $\Pr[X^{\ell,k}_{n,r} = 1]$ requires evaluating the η function which requires r multiplications. At level ℓ, $\Pr[X^{\ell,k}_{n,r} = 1]$ is same for $k = 1, \ldots, q_\ell - 1$ and needs to be computed once. Hence, for these $q_\ell - 1$ nodes for all the ℓ levels $O(r \log n)$ multiplications are required. Contribution from the q^{th}_ℓ and $(q_\ell + 1)^{th}$ nodes at the level ℓ can be computed using $O(r \log n)$ multiplications and so the overall complexity is $O(r \log n)$ multiplications. The space complexity is $O(1)$. By running this algorithm, we make an interesting observation that for $r = 2$, $n = (2^{20} - 1)$ and 2^{20}, $\frac{E[X_{n,r}]}{r} = 1.249$. For $n = (2^{20} + 1)$, $\frac{E[X_{n,r}]}{r} = 1.499$. So, there is a sharp change in the value of $\frac{E[X_{n,r}]}{r}$ as n increases from $2^{20} - 1$, 2^{20} to $2^{20} + 1$. This also happens for other similar values of n.

4.1 Power-of-two Anomaly

The SD scheme of [NNL01] and all subsequent work on it, assume n to be a power of 2. For most practical implementations, n will not be a power of 2. Hence, to be able to use the scheme, additional dummy users have to be assumed and added to the collection of users to make n a power of 2. If n is of the form $2^{t_0}+1$, then the next high power is 2^{t_0+1} and hence the number of dummy users added is almost equal to n. These dummy users can either be assumed to be revoked or privileged. Unless the implementation requires redundancy in the number of privileged users, assuming the dummy users to be privileged will lead to useless increase in key storage and may lead to increase in header length. This may be less of a problem for implementations in DVD-s, but it will definitely be a serious issue with implementations like pay-TV where user storage as well as transmission overhead are crucial efficiency parameters. The storage space required by each user is $\frac{1}{2}\log^2 n + \frac{1}{2}\log n + 1$ in the SD method. If the number of dummy users is almost equal to n, then the useless storage is around $\log 2 \log n + \frac{1}{2}\log^2 2 + \frac{1}{2}\log 2$ which is significant compared to the total storage size of the user. Hence, it is more reasonable to assume the dummy users to be revoked. But such an assumption adversely affects the efficiency of the scheme. The following table enlists the true values of n and r, the corresponding assumed values of n' (of the form 2^{t_0+1}) and r' and their respective expected header lengths.

n	r	$E[X_{n,r}]$	n'	r'	$E[X_{n',r'}]$
$2^{16}+1$	2^7	160.03	2^{17}	$2^7+(2^{16}-1)$	38065.75
$2^{16}+8$	2^7	160.01	2^{17}	$2^7+(2^{16}-8)$	38066.14
$2^{16}+2^{15}$	2^7	159.15	2^{17}	$2^7+(2^{15})$	29771.38
$2^{16}+1$	2^{10}	1253.96	2^{17}	$2^{10}+(2^{16}-1)$	38008.61
$2^{16}+8$	2^{10}	1253.85	2^{17}	$2^{10}+(2^{16}-8)$	38009.11
$2^{16}+2^{15}$	2^{10}	1260.33	2^{17}	$2^{10}+(2^{15})$	30275.03

(14)

The above table shows that if n is rounded off to n' (a power of 2) as suggested by [NNL01] and all subsequent work, the expected header length suffers an enormous increase. This clearly brings out the significance of our contribution.

4.2 Complete Tree: Asymptotic Analysis of Expected Header Length

For $n = 2^{t_0+1}$, $n_\ell = n_r$ and hence for a node t (the k^{th} node of level ℓ), (12) can be written as:

$$\Pr[X_{n,r}^{\ell,k} = 1] = 2\left(\eta_r(n, 2^{\ell-1}) - \eta_r(n, 2^\ell) - \eta_r(n, 3\times 2^{\ell-1}) + \eta_r(n, 2^{\ell+1})\right). \quad (15)$$

This probability is independent of k and hence, is equal for all nodes at level ℓ which we denote by $B_{n,r}^{(\ell)}$. We define the expected header length $H_{n,r}$ as: $H_{n,r} = E[X_{n,r}] = \sum_{\ell=1}^{t_0+1} 2^{t_0+1-\ell} B_{n,r}^{(\ell)}$. We define $D_{n,r} = H_{n,r} - H_{n,r-1}$ so that $H_{n,r} =$

$H_{n,r-1} + D_{n,r} = 1 + \sum_{i=2}^{r} D_{n,i}$. Expanding the expressions for $B_{n,r+1}^{(\ell)}$ and $B_{n,r}^{(\ell)}$, and simplifying, we obtain the following:

$$D_{n,r+1} = \frac{n}{n-r}\left[-\eta_r(n,1) + \eta_r(n,2) + 3\eta_r(n,3)\right] - \frac{3n}{n-r}\left[\sum_{\ell=1}^{t_0}\left(\eta_r(n, 2\times 2^{\ell} - \eta_r(n, 3\times 2^{\ell})\right)\right]. \tag{16}$$

For the asymptotic analysis, we define $H_r = \lim_{n\to\infty} H_{n,r}$, $D_r = \lim_{n\to\infty} D_{n,r}$ and so, $H_r = 1 + \sum_{i=2}^{r} D_i$. Using algebraic manipulations, we get the following result:

Theorem 4 *$D_{r+1} \uparrow 3 - 3K_r$ where $K_r = \left(-\frac{1}{2}\right)^r + \sum_{j=1}^{r}(-1)^j \binom{r}{j} \frac{(2^j-3^j)}{(2^j-1)}$.*

r	2	3	4	5	6
D_r	$\frac{3}{2}$	$\frac{5}{4}$	$\frac{69}{56}$	$\frac{417}{336}$	$\frac{25953}{20832}$
$\frac{H_r}{r}$	1.25	1.25	1.24	1.24	1.24

(17)

The above table lists the values of D_r and $\frac{H_r}{r}$ for small values of r. This table shows the ratio $\frac{H_r}{r}$ is 1.25 for $r = 2$ and decreases with increasing r. This explains the empirical observation in [NNL01] that the expected header length is $1.25r$.

5 Analysis of the Layered Subset Difference Scheme

The basic LSD Scheme of [HS02] causes an asymptotic improvement in the number of keys stored per user over the SD scheme of [NNL01], with a compromise on the header length. The amount of information that a user has to store in the SD method, improves from $O(\log^2(n))$ to $O(\log^{3/2}(n))$. It modifies the SD scheme by dividing the levels of nodes of the tree $\mathcal{T}^0$ into $\sqrt{\log n}$ layers with each layer containing $\sqrt{\log n}$ levels of $\mathcal{T}^0$. A level at the border of two adjacent layers is said to be *special* and is included in both the layers (above and below it). To get the collection of subsets S_{LSD} for the LSD scheme, a subset $S_{i,j}$ in the collection S_{SD} of the SD method is divided into two subsets $S_{i,k}$ and $S_{k,j}$ where node k is either in the same layer as node i or is at a special level.

For a node t, let ℓ_t denote the level of t and k_t be its index in that level. Let $\mathsf{NSDesc}(t)$ be the set of all non-special descendant nodes of node t which are not at the same layer as t. Similar to the technique used for the SD method in Section 4, finding the expected header length for LSD boils down to finding $\Pr[X_{n,r}^{\ell_i,k_i} = 1 \wedge X_{n,r}^{\ell_j,k_j} = 1]$ for a pair of nodes (i,j) such that j is a descendant node of i in $\mathcal{T}^0$ and $j \in \mathsf{NSDesc}(i)$. The event $X_{n,r}^{\ell_i,k_i} = 1 \wedge X_{n,r}^{\ell_j,k_j} = 1$ occurs when there is at least one revoked user in each of the following subtrees: the subtree rooted at the sibling node of i and each of the two child subtrees of j. Using this characterization, we find the expected header length.

References

[AAC] AACS. Advanced access content system, http://www.aacsla.com. [AAC].

[AK08] Per Austrin and Gunnar Kreitz. Lower bounds for subset cover based broadcast encryption. In Serge Vaudenay, editor, *AFRICACRYPT*, volume 5023 of *Lecture Notes in Computer Science*, pages 343–356. Springer, 2008.

[AKI03] Nuttapong Attrapadung, Kazukuni Kobara, and Hideki Imai. Sequential key derivation patterns for broadcast encryption and key predistribution schemes. In Chi-Sung Laih, editor, *ASIACRYPT*, volume 2894 of *Lecture Notes in Computer Science*, pages 374–391. Springer, 2003.

[Ber91] Shimshon Berkovits. How to broadcast a secret. In *EUROCRYPT*, pages 535–541, 1991.

[CFN94] Benny Chor, Amos Fiat, and Moni Naor. Tracing traitors. In Yvo Desmedt, editor, *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 257–270. Springer, 1994.

[DRM] DRM. http://en.wikipedia.org/wiki/drm_(computing). In *Digital Rights Management* [DRM].

[FN93] Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer, 1993.

[HS02] Dani Halevy and Adi Shamir. The LSD broadcast encryption scheme. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 47–60. Springer, 2002.

[JHC+05] Nam-Su Jho, Jung Yeon Hwang, Jung Hee Cheon, Myung-Hwan Kim, Dong Hoon Lee, and Eun Sun Yoo. One-way chain based broadcast encryption schemes. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 559–574. Springer, 2005.

[LS98] Michael Luby and Jessica Staddon. Combinatorial bounds for broadcast encryption. In *EUROCRYPT*, pages 512–526, 1998.

[MMW09] Thomas Martin, Keith M. Martin, and Peter R. Wild. Establishing the broadcast efficiency of the subset difference revocation scheme. *Des. Codes Cryptography*, 51(3):315–334, 2009.

[NNL01] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62. Springer, 2001.

[NP98] Moni Naor and Benny Pinkas. Threshold traitor tracing. In Hugo Krawczyk, editor, *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, pages 502–517. Springer, 1998.

[PB06] E. C. Park and Ian F. Blake. On the mean number of encryptions for tree-based broadcast encryption schemes. *J. Discrete Algorithms*, 4(2):215–238, 2006.

[PGM04] Carles Padró, Ignacio Gracia, and Sebastià Martín Molleví. Improving the trade-off between storage and communication in broadcast encryption schemes. *Discrete Applied Mathematics*, 143(1-3):213–220, 2004.

[PGMM03] Carles Padró, Ignacio Gracia, Sebastià Martín Molleví, and Paz Morillo. Linear broadcast encryption schemes. *Discrete Applied Mathematics*, 128(1):223–238, 2003.

[Sti97] Douglas R. Stinson. On some methods for unconditionally secure key distribution and broadcast encryption. *Des. Codes Cryptography*, 12(3):215–243, 1997.