Journal of Discrete Algorithms 34 (2015) 78–107

Contents lists available at ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

Tree based symmetric key broadcast encryption

Sanjay Bhattacherjee *, Palash Sarkar

Applied Statistics Unit, Indian Statistical Institute, 203, B.T. Road, Kolkata, 700108, India

ARTICLE INFO

Article history:
Received 2 September 2014
Received in revised form 11 March 2015
Accepted 27 May 2015
Available online 1 June 2015

Keywords:
Broadcast encryption
Subset difference
Trees
General arity
Probabilistic analysis
Header length
Transmission overhead
Cyclotomic cosets
Layering

ABSTRACT

The most influential broadcast encryption (BE) scheme till date was introduced in 2001 by Naor, Naor and Lotspiech (NNL) and is based on binary trees. This paper generalizes the ideas of NNL to obtain BE schemes based on k-ary trees for any $k \geq 2$. The treatment is uniform across all k and essentially provides a single scheme which is parameterized by the arity of the underlying tree. We perform an extensive analysis of the header length and user storage of the scheme. It is shown that for a k-ary tree with n users out of which r are revoked, the maximum header length is $\min(2r-1, n-r, \lceil n/k \rceil)$. An expression for the expected header length is obtained and it is shown that the expression can be evaluated in $O(r \log n)$ time. Experimental results indicate that for values of r one would expect in applications such as pay TV systems, the average header length decreases as k increases. The number of keys to be stored by any user is shown to be at most $(\chi_k - 2)\ell_0(\ell_0+1)/2$, where $\ell_0 = \lceil \log_k n \rceil$ and χ_k is the number of cyclotomic cosets modulo $2^k - 1$. In particular, when the number of users is more than 1024, we prove that the user storage required for $k = 3$ is less than that of $k = 2$. For higher values of k, the user storage is greater than that for binary trees. The option of choosing the value of k provides a designer of a BE system with a wider range of trade-offs between average header length and user storage. The effect of layering on the k-ary tree SD scheme is also explored.

1. Introduction

Broadcast Encryption (BE) deals with the problem of broadcasting encrypted data. For each transmission (or session), there is a set of *privileged* users who should be able to decrypt the data and a set of *revoked* users who should not be able to do so. In symmetric key BE, there is a center which initially distributes keys to all the users and also broadcasts the encrypted data in each session. In each session, the data to be broadcast is encrypted with a random session key using a symmetric key encryption algorithm. This session key is further encrypted using other keys and the encryptions of the session key are sent as the *header* with the encrypted body. The number of times the session key is encrypted for each session is called the *header length*. Any privileged user will be able to use its secret information to correctly decrypt the session key from the header and hence the message sent in the session. A *fully resilient* scheme ensures that an adversary with the secret information of all the revoked users cannot decrypt the broadcast correctly. Two important efficiency parameters for a BE scheme are the header length and the user storage which is the amount of secret information that each user has to store.

The most popular symmetric key BE scheme was proposed in 2001 by Naor, Naor and Lotspiech (NNL) and was called the Subset Difference (SD) Scheme [17,18]. The NNL-SD scheme assumes the number of users to be a power of two and

* Corresponding author.
E-mail addresses: sanjayb_r@isical.ac.in (S. Bhattacherjee), palash@isical.ac.in (P. Sarkar).

http://dx.doi.org/10.1016/j.jda.2015.05.010
1570-8667/

these users are associated with the leaves of a *full binary tree* $\mathcal{T}^0$. This scheme has a storage requirement of $O(\log^2 n)$ and a worst case header length of $2r-1$, where n is the total number of users and r is the number of revoked users.

The idea of layering introduced by Halevy and Shamir [12] reduced the user storage of the NNL-SD scheme from $O(\log^2 n)$ to $O(\log^{3/2} n)$ at the cost of increased communication overhead. In [7], this layering strategy was generalized to obtain different trade-offs between the user storage and communication overhead. An important optimization was the storage minimal layering that guaranteed minimal storage requirement that could be achieved using layering for a given n.

1.1. Our contributions

In this work, we extend the ideas of NNL to k-ary trees for any $k \geq 2$. Our treatment is general and unified, i.e., the same approach works for all values of k. Suppose n is a power of k, i.e., $n = k^{\ell_0}$ for some $\ell_0 \geq 1$ and consider the users to be the leaf nodes of a full k-ary tree of height ℓ_0. Let $j_1, \ldots, j_c$, $1 \leq c \leq k$, be a set of sibling nodes in this tree and i is an ancestor of these nodes. Consider the set S of leaf nodes in the subtree formed by taking away the subtrees rooted at $j_1, \ldots, j_c$ from the subtree rooted at i. So, the set S is formed as a subset difference of two sets of users. Subsets of users arising in this manner are called SD sets. The identification of the SD sets is a key aspect of obtaining the k-ary tree scheme. This idea extends the idea of SD sets introduced for binary trees in [17,18].

An intuition behind considering k-ary trees with $k > 2$ is that the number of SD sets grows with increasing k and so the header length may come down at the cost of increasing the user storage. This, however, does not turn out to be entirely true. Working out the details of the scheme and the resulting analysis shows up a rich complexity of behavior which is not apparent at the outset. We provide an extensive analysis of the scheme covering the following points.

Cover generation algorithm. A single cover generation algorithm which works for all k is developed. This is an intuitively simple algorithm which uses just an array as the underlying data structure. Specializing this algorithm for $k = 2$ yields the cover finding algorithm given in [17,18]. The description of the algorithm turns out to be considerably simpler that of [17,18].

Traitor tracing. The NNL paper [17,18] provides a mechanism for tracing traitors. With some modification, this idea also fits the k-ary BE scheme. It turns out that compared to binary trees, for $k \geq 3$, tracing traitors can be done with lesser number of queries.

Header length. For k-ary trees with n users, the maximum header length of a transmission with r revoked users is shown to be $\min(2r-1, n-r, \lceil n/k \rceil)$. Somewhat surprisingly, the first component, i.e., $2r-1$ is not affected by k. We show that the bound of $2r-1$ is indeed achieved for values of k greater than 2. Average case analysis of the header length is done under the assumption that the revoked set of users is distributed uniformly among the set of all users. With this assumption, we derive an expression for the expected header length. The method is to compute the probability that any internal node generates a subset in the header. Summing over all these probabilities provide the expected header length. The expression for the expected header length can be computed in $O(r \log n)$ time and $O(1)$ space. We have implemented the algorithm to compute the expected header length and provide representative values to show the average header lengths for different values of k.

User storage. During the initiation of the scheme, the center provides each user with sufficient information so that it is able to generate any key corresponding to an SD set of which it is a member. This information is measured in terms of the number of m-bit seeds that are required to be stored by any user. Here m is the size of the key of the underlying symmetric cipher. The work of NNL provides a clever way to use a pseudo-random generator so that user storage consists of $1 + \lceil \log_2 n \rceil (\lceil \log_2 n \rceil + 1)/2$ seeds. The direct combination of this idea with the SD sets of a k-ary tree makes the user storage to be $1 + (2^{k-1} - 1)\lceil \log_k n \rceil (\lceil \log_k n \rceil + 1)/2$ seeds. We show that a modification based on the use of cyclotomic cosets modulo $2^k - 1$ reduces the user storage to $1 + (\chi_k - 2)\lceil \log_k n \rceil (\lceil \log_k n \rceil + 1)/2$ seeds, where χ_k is the number of cyclotomic cosets modulo $2^k - 1$.

Tackling arbitrary number of users. When n is not a power of k, we show that a complete k-ary tree structure can be used to construct the BE scheme. This is an analogue of complete binary trees used in data structures. Average header length analysis of such schemes is performed using simulation studies.

Layering. The idea of layering is extended for the k-ary tree generalization of the SD scheme. The choice of the layering strategy determines the user storage of the layered version of the scheme. A dynamic programming algorithm is proposed to compute the layering strategies for which the user storage is minimum. This generalizes the algorithm for $k = 2$ which was given in [7].

Simulation study of the header length. We perform a simulation study of the average header length for $n = 10^x$ $(x = 3, \ldots, 8)$ users and for $k = 2, \ldots, 8$. Experimental results indicate that there is a cut-off value δ_k such that for $r/n > \delta_k$, the average header length of the k-ary scheme is lesser than that of the binary tree based scheme. Further, the value of δ_k decreases as k increases. This suggests that by increasing k, it is possible to reduce the header length for lower values of r. This can be important for applications such as Pay-TV systems. The trade-off is a one-time moderate increase in user storage.

In applications like [1] where the storage for the header is fixed in the standard, there is a threshold for the maximum number of users that the system can accommodate. With decrease in the average header length, the system can accommodate more number of revoked users on an average.