Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Networks 47 (2005) 429–441
COMPUTER NETWORKS
www.elsevier.com/locate/comnet

A lower bound for multicast key distribution

Jack Snoeyink [a], Subhash Suri [b], George Varghese [c,*]

[a] Department of Computer Science, UNC-Chapel Hill, Chapel Hill, NC 27599-3175, USA
[b] Department of Computer Science, University of California, Santa Barbara, CA 93106, USA
[c] Computer Science and Engineering Department, University of California, 9500 Gilman Drive, La Jolla, CA 92093-0114, USA

Received 2 September 2002; received in revised form 3 March 2004; accepted 7 September 2004
Available online 19 October 2004

Reponsible Editor: S. Lam

Abstract

With the rapidly growing importance of multicast in the Internet, several schemes for scalable key distribution have been proposed. These schemes require the broadcast of $\Theta(\log n)$ encrypted messages to update the group key when the nth user joins or leaves the group. In this paper, we establish a matching lower bound (Independently, and concurrently, Richard Yang and Simon Lam discovered a similar bound with slightly different properties and proofs. An earlier version of our paper appeared in Infocom 2001 while their result appears in [R. Yang, S. Lam, A secure group key management communication lower bound, Technical Report TR-00-24, Department of Computer Sciences, UT Austin, July 2000, revised September 2000].), thus showing that $\Theta(\log n)$ encrypted messages are necessary for a general class of key distribution schemes and under different assumptions on user capabilities. While key distribution schemes can exercise some tradeoff between the costs of adding or deleting a user, our main result shows that for any scheme there is a sequence of $2n$ insertion and deletions whose *total* cost is $\Omega(n \log n)$. Thus, any key distribution scheme has a worst-case cost of $\Omega(\log n)$ either for adding or for deleting a user.
© 2004 Elsevier B.V. All rights reserved.

Keywords: Multicast; Security; Protocol analysis

1. Introduction

Many distributed applications—such as interactive games, teleconferencing, and chat rooms—use a *group* paradigm. While such applications using groups can be implemented over point-to-point communication links, there are advantages to using multicast or broadcast as the underlying communications primitive.

A broadcast channel such as a satellite allows a sender to communicate with every user that can listen to the channel using a *single* broadcast

* Corresponding author. Tel.: +1 858 822 0424.
E-mail addresses: snoeyink@cs.unc.edu (J. Snoeyink), suri@cs.ucsb.edu (S. Suri), varghese@cs.ucsd.edu (G. Varghese).

1389-1286/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2004.09.001

message. With n users, broadcast can be n times cheaper than sending n separate unicast messages. The notion of broadcasting extends to a network of point-to-point links, such as the Internet where the routers can make extra copies of a message for all downstream links to which the message is intended. Further, broadcast generalizes to *multicast* where a message can be sent to a *subset* of all the Internet nodes.

Multicast is easily accomplished by assigning separate multicast addresses for each subset that wishes to communicate, and creating a separate Steiner tree [7] for each such subset. Despite the slowness of initial deployment, Internet multicast [7] is likely to become an important and well-used Internet paradigm.

The original Internet protocols paid little attention to secure communication, but commercial success has lead to many proposals for Internet security (e.g., IPsec [18]) that allow unicast messages to travel encrypted. Security concerns for IP multicast are even greater, due to the nature and distribution of the traffic. When a multicast message is sent to one station on say a satellite link, other stations in the range of the satellite can listen to the packets. If the listening stations have not paid for the service, then cryptographic techniques must be used to prevent unauthorized listeners from using the service.

This paper is about the problem of maintaining secrecy for multicast communication using any multicast or broadcast communication primitive, including the Internet multicast protocols as an important special case. Although there are proposals for group security that use sophisticated cryptographic techniques [16], we concentrate on secrecy by encrypted communication using simple and efficient private key techniques (e.g., DES) for group data encryption. Since secret key techniques are well studied and widely deployed, the main problem is key distribution: sending keys to all the group recipients in a *scalable* fashion.

The scalable key distribution problem is interesting because a number of applications can use *large* multicast groups that are also highly *dynamic* (i.e., users can be added or deleted frequently). For example, distributed war gaming and teleconferencing [19], applications can have thousands of users at any time with ten percent of the users changing over a period of one minute, and a constraint that users be added or dropped within a second.

Simple extensions of unicast key distribution protocols (e.g., [10]) take linear time to add or remove a user, which would be problematic for the dynamic scenarios described above. Recent proposals [20,6,19] introduced a *Key Graph* scheme for scalable key distribution that takes $O(\log n)$ messages to add to or delete from a group of n users. We describe the Key Graph scheme in the next section. The main question that we investigate in this paper is whether the Key Graph scheme is optimal for scalable, multicast key distribution. For this, we must define the security requirements for key distribution.

1.1. Security requirements

Intuitively, the main requirement is *confidentiality*: only valid users should be able to decrypt the multicast data even if the data is broadcast to the entire network. We assume in what follows that data is encrypted to ensure confidentiality using a symmetric cryptosystem such as DES. Thus, the confidentiality requirement can be translated into the following four requirements on key distribution:

Non-group confidentiality: Users that were never part of the group should not have access to any key that can decrypt any multicast data sent to the group.
Future confidentiality: Users deleted from the group at time t do not have access to any keys used to encrypt data after t unless they are added back to the group.
Collusion freedom: No set of deleted users should be able to pool the keys they had before deletion to decrypt future communication.
Past Confidentiality: A user added at time t should not have access to any keys used to encrypt data before t while the user was not part of the group.

The last requirement is debatable. It protects against an attack in which an unsubscribed user could record encrypted broadcasts for a long per-

Download English Version:

https://daneshyari.com/en/article/10339608

Download Persian Version:

https://daneshyari.com/article/10339608

Daneshyari.com