

A New Approach for Source Authentication of Multicast Data

Jinxin He 1 College of Computer Science and Technology Jilin University Changchun, P.R. China	Gaochao Xu College of Computer Science and Technology Jilin University Changchun, P.R. China	Zhiguo Zhou 1 College of Computer Science and Technology Jilin University Changchun, P.R. China	Guannan Gong College of Computer Science and Technology Jilin University Changchun, P.R. China
2 College of Earth Science Jilin University Changchun, P.R. China		2 College of Computer Northeast Normal University Changchun, P.R. China	

Abstract- *Multicast Security is one of the important problems to solve for the successful deployment of group communication applications, and source authentication of multicast data is a key and difficult problem. To solve it, a new approach is afforded based on the basic tree hashing protocol. The performance evaluation shows there are less computation and communication overheads in it, so which is reasonable and applicable.*

Keywords: Multicast Security; Source Authentication; Tree Hashing

1 Introduction

Satellite TV distribution, software distribution, stock quote streaming, Web caching, and multimedia conferencing are examples of applications that require one-to-many or many-to-many group communication[6]. Multicast enables efficient group communication by allowing the sender to transmit a single copy of data, with network elements such as routers and switches making copies as necessary for the receivers. Thus multicast reduces the computational load at the sender, as well as the number of copies of data on the network. Unfortunately, despite the vast amount of research and development of multicast protocols in the past decade, deployment of multicast applications has been slow. While some attribute this to no “killer applications”, the major factor is, in fact, that multicast services lack support for traffic management, accounting and billing, reliability, and security[3].

We identify multicast security as one of the most important problems to solve for the successful deployment of group communication applications. There are three distinct problem areas to consider in providing multicast security services[4]. First and most important, in secure multicast group members must be able to verify that the data received is indeed sent by an authorized sender. This is called origin authentication, includes group authentication and source authentication. Group authentication is the property that guarantees only that a message was sent (or last modified) by a member of the group. Since a MAC (Message Authentication Code) can be used for group authentication, it is rather inexpensive to authenticate even streaming data in real time. However, in most applications receivers must be able to establish the source of the data, at least for themselves. In other words, we need data source authentication. A stronger version of the above property, referred to as non-repudiation, enables a receiver to prove the origin of data to any impartial third party[5].

But source authentication of multicast data is a difficult problem. The simplest solution is to digitally sign each packet. But signing each packet is computationally expensive, and introduces excessive per packet communication overhead. Several solutions have been documented that amortize the cost of digital signatures over multiple packets, such as tree hashing[3]. However, although tree hashing reduces computation overhead, it increases more communication overhead than signing each packet. So we developed a new approach for source authentication of multicast data based on the basic tree hashing protocol, which can reduce the communication overhead and is more efficient and practical.

2 An overview of Tree Hashing

In the basic tree hashing protocol, the sender first divides the whole data into M blocks and then divides each block into m packets and computes the individual packet hashes. For block hash computation, it associates each individual packet hash with a leaf node of the hash tree. Each internal node's hash is the hash of the concatenation of the children's hashes. As Fig.1, $h_{12} = \text{hash}(h_1, h_2)$ Using this function, the sender recursively computes the root node's hash. With each packet, the sender includes the signed block hash, the packet ID, and the hashes of siblings of all the nodes in the current packet's path to the root[2]. Each receiver first computes the hash of the received packet. It uses the computed hash and the received hashes to compute the root hash. If the computed root hash is identical to the signed block hash, the received packet is authentic.

Authenticity verification of the first received packet of a block consists of a digital signature verification operation, and computation of all hashes in the path from the packet's position in the tree to the root. In all, the receiver needs to compute $O(\log m)$ hashes. Future packet verifications require fewer hash computations and no digital signature verification operations. Each packet carries only $O(\log m)$ hashes along with the signed block hash. Authenticity verification may require as many as $O(\log m)$ hash computations. Caching verified nodes decreases the number of hash computations for subsequent packet verifications of the same block.

Because the computation speed of hash functions(MD5, SHA-1.)is about 1,000 times faster than digital signature(RSA, ECC.)[1], the computation overhead of tree hashing is much less than signing each packet. And tree hashing also supports non-repudiation, since the root hash of each block is signed by the sender. However, per packet communication overhead is even higher than signing each packet. So if the communication overhead of tree hashing can be decreased to a reasonable level, it will be a perfect approach for source authentication of multicast data, especially to unreal-time applications.

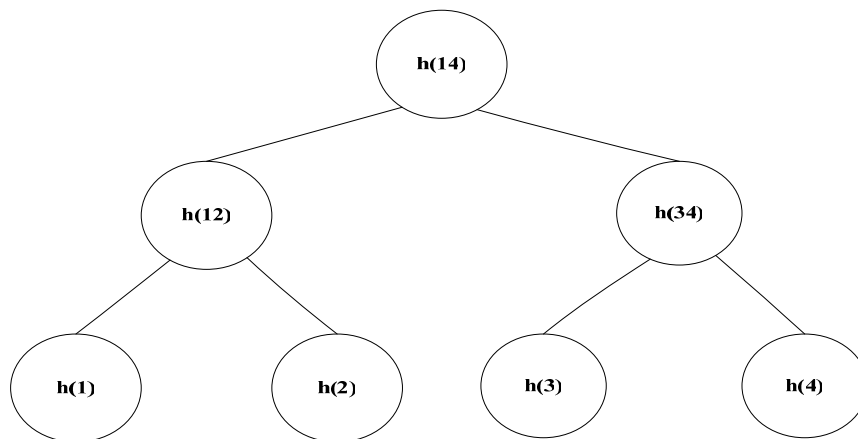


Fig.1 Tree Hashing

3 Our Extensions

In the basic tree hashing protocol, each packet sent by the sender includes the signed block hash, the packet ID, and the hashes of siblings of all the nodes in the current packet's path to the root. But only the first packet to each receiver requires verifying digital signature each block, in other words, if each block is divided into m packets, there are $(m-1)$ packets including useless digital signature per block. More important, digital signature is much longer than hash. For example, the length of data in RSA algorithm is about 128 bytes and SHA-1 is only 20 bytes. So we decided to reduce the communication overhead of tree hashing by making less packets carry digital signatures.

The main idea of our extensions to tree hashing is to separate digital signatures from data packets. First, the sender only sends per block's digital signature to each receiver, and the receiver needs to authenticate the source of received packets and buffer them. Since the digital signature of each block is important to receiver, the sender calculating a hash of whole data packets and signing it once is secure and reasonable. Second, the sender sends the real data to each receiver signed with tree hashing, but each packet includes block ID instead of block signatures. Fig. 2 depicts the relationship between these two sorts of packets. For each receiver, the first packet of per block is authenticated by verifying the buffered block signature with block ID, and the authentications of other packets in the same block only need to calculate hashes. Because all block signatures were buffered on the receiver's side in advance and real data packets need not carry any block signatures but some hashes, the communication overhead of our approach is much less than the basic tree hashing protocol.

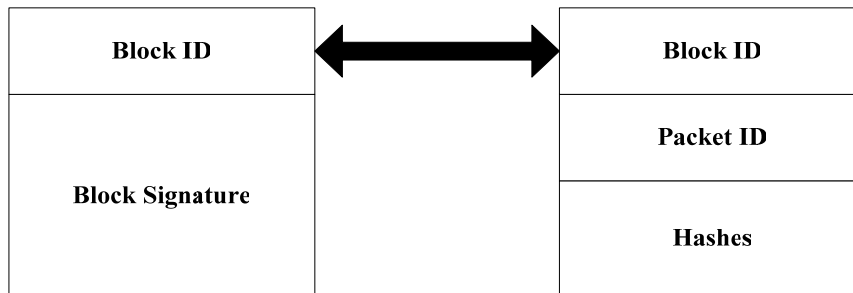


Fig.2 Packet Definition

4 Performance Evaluation

As mentioned above, both computation and communication overheads are two most important keys to source authentication of multicast data. In our approach, the computation overhead is as small as the basic tree hashing protocol. Furthermore, if the size of each block is reasonable, the communication overhead of ours is much less than signing each packet. For example, if we adopt RSA as the algorithm of digital signature and hash function is SHA-1 (the length of data in RSA is about 128 bytes and SHA-1 is typically 20 bytes), for each packet carries only $O(\log m)$ hashes and no block signature in our approach, the size of each packet is reduced unless $m \geq 2^{128/20}$. Even in a very large group, m should not be too large as buffer considered. So both computation and communication overheads of our approach are reasonable and acceptable.

5 Conclusion and Future work

In this paper, we have presented an extension to the basic tree hashing protocol which provides a new approach for source authentication of multicast data. Compared to some documented protocols, such as TESLA (Timed Efficient Stream Loss-tolerant Authentication) and hash chaining, TESLA that based MAC and delayed key chains requires the loose synchronization between the sender and receiver, and Non-repudiation is not supported[7,8]; hash chaining can not tolerance packet loss even disorder[3], Our approach provides some benefits as below:

- ◆ Both computation and communication overheads are reasonable and acceptable;
- ◆ Each packet is authenticated separately, which is loss-tolerant;
- ◆ Non-repudiation is supported since the hash is signed by the sender.

So our approach is efficient and secure for source authentication of multicast data. But it requires the receiver's buffer to hold all block signatures besides the sender's side, which is prone to DoS(Denial of Service) attacks and will occupy more system resource when the number of blocks is very large. So some related work should be continued to solve this problem.

6 References

- [1] M.Baughner, R.Canetti, L.Dondeti and F.Lindholm. "Multicast Security(MESC) Group Key Management Architecture". IETF RFC 4046, April 2005.
- [2] Yacine Challal, Abdelmadjid Bouabdallah and Yoann Hinard. "Efficient multicast source authentication using layered hash-chaining scheme". Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks(LCN'04).
- [3] Thomas Hardjono, Laksmiath R. Dondeti. "Multicast and Group Security". Artech House Inc, 2003.
- [4] T. Hardjono, B. Weis. "The Multicast Group Security Architecture". IETF RFC 3740, March 2004.
- [5] Xianxian Li, Jinpeng Huai. "Efficient Non-Repudiation Multicast Source Authentication Schemes". J. Comput. Sci. & Technol. pp.820-829, 17(6), 2002.
- [6] Sanjoy Paul. "Multicasting on the internet and its applications". Kluwer Academic Publishers, 1998.
- [7] A. Perrig, R. Canetti, D. Song, and J. D. Tygar. "Efficient and secure source authentication for multicast". Network and Distributed System Security Symposium, pp.35-46, February 2001.
- [8] A.Perrig, D.Song, R.Canetti, J.D.Tygar and B.Briscoe. "Timed Efficient Stream Loss-Tolerant Authentication(TESLA): Multicast Source Authentication Transform Introduction". IETF RFC 4082, June 2005.