Authenticated Multicast Immune to Denial-of-Service Attack*

Shouhuai Xu
Laboratory for Information Security Technology
George Mason University
MSN 4A4, University Dr., Fairfax, VA 22030, USA
sxu1@gmu.edu

Ravi Sandhu
SingleSignOn.Net, Inc. Reston, VA, USA
and
George Mason University, Fairfax, VA, USA
sandhu@gmu.edu

ABSTRACT

Authentication of multicast streams has attracted a lot of attention in the last few years. However, two important issues, namely multicast denial-of-service and access control, have been ignored in previous proposals. In this paper, we propose two Internet Protocol multicast authentication schemes by making use of the multicast tree as an essential authentication mechanism. Our schemes are efficient and immune to multicast denial-of-service attack. They allow the receivers to immediately authenticate the packets regardless of the packet loss characteristics of the underlying network.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General–security and protection (e.g., firewalls)

General Terms

Security

Keywords

Multicast authentication, Denial-of-service

1. INTRODUCTION

With the popularity of the Internet, simultaneous transmission of digital information becomes a prevalent model of communication. For this purpose, multicast protocols have been proposed. The basic idea of multicast is that each packet from a source is automatically duplicated and sent to the receivers. In this context, two important topics – multicast authentication and multicast secrecy – have recently attracted a lot of attention. We in this paper focus on Internet Protocol (IP) multicast authentication.

*The is the extended version with the proof of the theorems.

SAC 2002, Madrid, Spain

1.1 Prior Work

Gennaro and Rohatgi [5] introduce some interesting techniques for signing digital streams. However, their solutions are not robust against packet loss. Wong and Lam [16] propose using Merkle signature tree to tolerate packet loss. Although the cost for generating and verifying a signature can be amortized to multiple packets, their solution requires large communication overhead because each packet contains a signature as well as the nodes necessary to compute the root of the tree. Rohatgi [14] proposes a solution based on k-time signature, yet this solution still needs 90 bytes for a 6-time signature public key (which does not include the certificate of the public key) and 300 bytes for each signature.

Perrig et al. [13] propose a solution they call TESLA. The main idea is to let the sender attach to each packet a MAC computed using a key k, and to let the receivers buffer a packet after receiving it. A short while later, the sender discloses k and the receivers are thus able to authenticate the buffered packet. Although some of the shortcomings of TESLA are overcome in their following work [12], a drawback of [13, 12] is the requirement for synchronization among the group members.

They [13] also propose a different solution called EMSS, in which the sender periodically sends signature packets to the receivers. This solution is somewhat comparable to the original proposals in [5, 14]. In order to resist packet loss, EMSS randomly chooses packets for duplicating the hashes of the others. This provides probabilistic guarantee that a packet can be authenticated given an expected amount of packet loss in a stream. Golle et al [6] propose using deterministic redundancy for duplicated hashes to realize optimized resistance against bursty loss. Along this line, Miner et al [8] propose a unified framework considering both bursty and random packet loss, from which many of the constructions in [5, 14, 13, 6] can be instantiated.

Canetti et al [3] propose letting each receiver hold a different subset of s (out of S) MAC keys and the sender send each packet along with S MAC values. A receiver can verify s MAC values with respect to which she has the corresponding MAC keys. The key management guarantees that no coalition of t receivers can fool an honest receiver to accept a bogus packet. Along this line, Boneh et al. [2] prove that one cannot build a short collusion resistant multicast MAC without relying on digital signature, and that the Canetti et al. [3] solution is optimal when the number of colluding receivers is small. As we will see, our Scheme 2 is not only signature-less, but also able to bridge two-party MAC

and multi-party MAC in the sense that a unique MAC key is shared by the source and the receivers. Note that our construction does not contradict their conclusions, because authentication in our Scheme 2 is based on the unique MAC key as well as the multicast tree, whereas authentication in [3, 2] is completely based on the MAC keys.

1.2 Our Contributions

We propose addressing IP multicast authentication by making use of the multicast tree as an essential authentication mechanism, while taking *multicast denial-of-service* and *access control* into consideration. Multicast denial-of-service is an attack based on abusing the automatic duplication functionality of multicast. Access control guarantees that only those who have subscribed to the service can authenticate the data.

Specifically, we present two schemes that can be easily integrated into the multicast channel model proposed in [7] and the multicast secrecy frameworks proposed in [1, 9, 10, 15]. In Scheme 1, the source authenticates each packet with a different MAC key, which is transmitted simultaneously along with the MAC value. In addition to being immune to multicast denial-of-service attack, it has the desirable property that dynamic maintenance (due to dynamic access control) is confined within the corresponding leaf domains (as in [10]). Scheme 2 is indeed a multi-party counterpart of two-party MAC in the sense that there is a unique MAC key shared between the source and the receivers. The comparison between our schemes and TESLA [13] and the "benchmark" solution (i.e., each packet is independently signed) is presented in Table 1, where PRF denotes pseudorandom function, and h is the maximal number of cryptographic routers on the paths from the root to the receivers.

Outline. In section 2, we introduce the cryptographic primitives as well as the atomicity of Ethernet broadcast. In section 3, we present the model and goals. In section 4, two concrete schemes are presented. We conclude in section 5.

2. PRELIMINARY

Target Collision Resistance. We use a pseudorandom function (PRF) family $\{f_k\}$ parameterized by a secret value k with the following target collision resistance [11]. An adversary $\mathcal{A}$ can win in the following game with only negligible probability: a key k is chosen at random; $\mathcal{A}$ is given $f_k(0)$; $\mathcal{A}$ manages to find $k' \neq k$ such that $f_{k'}(0) = f_k(0)$.

Forward-Secure Pseudorandom Generators (FSPRGs). We use a key chain with the property that exposure of the key at a given time period reveals no efficiently computable information about the key at any future time period. A key chain $k^{(0)}, k^{(1)}, \cdots, k^{(w)}$ can be constructed as $k^{(i-1)} = f_{k^{(i)}}(0)$ for $1 \leq i \leq w$, where $k^{(w)}$ is firstly chosen and $\{f_k\}$ is a PRF family. For convenience, we also denote $f^n(x) = f_{f^{n-1}(x)}(0)$, where $f^0(x) = x$.

IP multicast over Ethernet. In most real-world systems, the receiver computers attached to the same router are within the same broadcast domain of the underlying Ethernet Local Area Network, since IP multicast is finally mapped to Ethernet broadcast. As specified in [4], all IGMP (Internet Group Management Protocol) messages used by IP hosts to report their multicast group membership to routers are sent with IP TTL (Time To Live) 1. Due to the properties of Ethernet CSMA/CD, we assert the following fact.

> **Fact** (atomicity of Ethernet broadcast). Within the same Ethernet broadcast domain, either all computers receive a broadcast message, or none of them receive it. That is, no dishonest receiver can receive a broadcast message while preventing an honest receiver from receiving it.

3. MODEL AND GOALS

The Model. In a typical real-world IP multicast system, a source/sender computer (called *source*) multicasts digital packets to a set of receiver computers (called *receivers*) via a multicast tree consisting of routers. The routers with cryptographic processing capability are called *nodes*. The node closest to the source is called *root node*. A node to which there are receivers attached/connected is called *leaf node*. The receivers attached to the same leaf node make up a *leaf*, which is treated as a single entity in the multicast tree. For example, in Figure 1, N_0 is the source, N_1 is the root node, N_4, N_6, and N_8 are the leaf nodes, and the receivers are grouped into three leaves lf_1, lf_2, and lf_3.

The Adversarial Model. The adversary, a probabilistic polynomial-time Turing machine, has access to a network that is faster than the network the honest receivers have access to, and has total control over the communication lines (i.e., it can arbitrarily discard and disorder messages). The adversary may corrupt arbitrarily many receivers. The adversary manages to do the following. First, it attempts to convince an honest receiver to accept a bogus packet as if the packet is from the genuine source. Second, it attempts to impose denial-of-service attack by convincing a node to forward a bogus packet through the multicast tree.

The Goals. We consider the following properties for a multicast authentication scheme.

- Security. A multicast authentication scheme is secure, if an honest receiver will only accept packets from the genuine source.
- Real-time authentication. A receiver can authenticate any packet immediately.
- Efficiency. A scheme should be efficient in both computation and communication.
- Immunity to multicast denial-of-service (DOS) attack. A bogus packet should be filtered as early as possible.
- Dynamic maintenance. Access control should be maintained.

4. THE SCHEMES

Suppose a multicast stream is encapsulated into w packets, $M^{(1)}, M^{(2)}, \cdots, M^{(w)}$, the encryption functions are derived from PRF family $\{g_k\}$, and the key chain is derived from PRF family $\{f_k\}$.

4.1 Scheme 1

For simplicity and concreteness, we take the multicast tree in Figure 1 as an example. Suppose Alice is among the receivers attached to leaf node N_8.

Table 1: Comparison of the proposals

Scheme	Computation (PRFs/packet)	Communication (bytes/packet)	Need synchronized clock	Immune to Multicast DOS attack
Sign each packet	(1 exponentiation)	(128 for RSA 1024)	NO	NO
TESLA	1	24	YES	NO
Our scheme 1	$4h$+9	34	YES	YES
Our scheme 2	$2h$+7	24/34	NO	YES

4.1.1 The Construction

Initialization. The participants execute as follows.

- The source, N_0, generates and stores a key chain $k^{(0)}$, $k^{(1)}$, $\cdots$, $k^{(w)}$ where $k^{(i-1)} = f_{k^{(i)}}(0)$ for $1 \leq i \leq w$. It sends $k^{(0)}$ to the legitimate receivers via private channels. It also chooses and sends a pair of random secrets (t_0, v_0) to the *root node*, N_1, such that t_0 will be used to encrypt the MAC keys and v_0 will be used to authenticate the packets. Note that N_0 will authenticate $M^{(i)}$ by using $u^{(i)} = f_{k^{(i)}}(1)$ as the MAC key.
- A node N_j $(1 \leq j \leq 8)$ chooses and sends a pair of random secrets (t_j, v_j) to its children such that t_j will be used to encrypt the MAC keys and v_j will be used to authenticate the packets. N_j sets $i' \leftarrow 0$, where i' is the index for the last packet that has been successfully processed.
- Each receiver sets $i' \leftarrow 0$ and $k^{(i')} \leftarrow k^{(0)}$, where i' is the index for the last packet having been successfully processed and $k^{(i')}$ is the corresponding key on the key chain from which the MAC keys are derived.

Runtime. We show how the i^{th} $(1 \leq i \leq w)$ packet, $M^{(i)}$, flows through the third path in Figure 1. Recall that N_0 keeps $(w, k^{(0)}, \cdots, k^{(w)}, t_0, v_0)$, N_1 keeps $(i' = 0, t_0, v_0, t_1, v_1)$, N_7 keeps $(i' = 0, t_1, v_1, t_7, v_7)$, N_8 keeps $(i' = 0, t_7, v_7, t_8, v_8)$, and Alice keeps $(i' = 0, k^{(0)}, t_8, v_8)$.

1. The source, N_0, computes $u^{(i)} = f_{k^{(i)}}(1)$, $tag^{(i)} = MAC_{u^{(i)}}(i, M^{(i)})$, $t_0^{(i)} = g_{t_0}(i)$, $\alpha = k^{(i)} \oplus t_0^{(i)}$, and $\beta = MAC_{v_0}(i, M^{(i)}, tag^{(i)}, \alpha)$. Then, it sends $(i, M^{(i)}, tag^{(i)}, \alpha, \beta)$ to the root node N_1.
2. N_1 receives $(i, M^{(i)}, tag^{(i)}, \alpha, \beta)$. N_1 discards it if $i \leq i'$ or $MAC_{v_0}(i, M^{(i)}, tag^{(i)}, \alpha) \neq \beta$. Otherwise, N_1
   (a) computes $t_0^{(i)} = g_{t_0}(i)$, $t_1^{(i)} = g_{t_1}(i)$, and $\alpha^* = \alpha \oplus t_0^{(i)} \oplus t_1^{(i)} = k^{(i)} \oplus t_1^{(i)}$;
   (b) computes $\beta^* = MAC_{v_1}(i, M^{(i)}, tag^{(i)}, \alpha^*)$ and sends $(i, M^{(i)}, tag^{(i)}, \alpha^*, \beta^*)$ to N_2 and N_7;
   (c) sets $i' \leftarrow i$.
3. N_7 receives $(i, M^{(i)}, tag^{(i)}, \alpha, \beta)$. N_7 discards it if $i \leq i'$ or $MAC_{v_1}(i, M^{(i)}, tag^{(i)}, \alpha) \neq \beta$. Otherwise, N_7
   (a) computes $t_1^{(i)} = g_{t_1}(i)$, $t_7^{(i)} = g_{t_7}(i)$, and $\alpha^* = \alpha \oplus t_1^{(i)} \oplus t_7^{(i)} = k^{(i)} \oplus t_7^{(i)}$;
   (b) computes $\beta^* = MAC_{v_7}(i, M^{(i)}, tag^{(i)}, \alpha^*)$ and sends $(i, M^{(i)}, tag^{(i)}, \alpha^*, \beta^*)$ to N_8;
   (c) sets $i' \leftarrow i$.
4. N_8 receives $(i, M^{(i)}, tag^{(i)}, \alpha, \beta)$. N_8 discards it if $i \leq i'$ or $MAC_{v_7}(i, M^{(i)}, tag^{(i)}, \alpha) \neq \beta$. Otherwise, N_8
   (a) computes $t_7^{(i)} = g_{t_7}(i)$, $t_8^{(i)} = g_{t_8}(i)$, and $\alpha^* = \alpha \oplus t_7^{(i)} \oplus t_8^{(i)} = k^{(i)} \oplus t_8^{(i)}$;
   (b) computes $\beta^* = MAC_{v_8}(i, M^{(i)}, tag^{(i)}, \alpha^*)$ and broadcasts $(i, M^{(i)}, tag^{(i)}, \alpha^*, \beta^*)$;
   (c) sets $i' \leftarrow i$.
5. Alice receives $(i, M^{(i)}, tag^{(i)}, \alpha, \beta)$. She discards it if $i \leq i'$ or $MAC_{v_8}(i, M^{(i)}, tag^{(i)}, \alpha) \neq \beta$. Otherwise,
   (a) she obtains $k^{(i)}$ by computing $t_8^{(i)} = g_{t_8}(i)$ and $k^{(i)} = \alpha \oplus t_8^{(i)}$;
   (b) if $k^{(i')} \neq f^{i-i'}(k^{(i)})$, she discards the packet; otherwise,
      i. she computes $u^{(i)} = f_{k^{(i)}}(1)$;
      ii. if $MAC_{u^{(i)}}(i, M^{(i)}) = tag^{(i)}$, she accepts the packet and sets $i' \leftarrow i$ as well as $k^{(i')} \leftarrow k^{(i)}$.

4.1.2 Properties of the Scheme

- Security. Suppose that the nodes are trusted as well as secure and that there are no colluding receivers attached to different leaf nodes, this scheme is provably secure (see Theorem 1 in the appendix).
- Real-time authentication. A receiver can immediately authenticate a packet regardless of the packet loss characteristics.
- The computational overhead is at most $4h + 9$ PRFs per packet, where h is the maximum number of nodes among all the paths. If we use 80-bit MAC for both the MAC key and the MAC value, the communication overload is 34 bytes per packet.
- Immunity to multicast DOS attack. This is proved as Theorem 2 in the appendix.
- Dynamic maintenance. Dynamic maintenance of membership is confined within the corresponding leaf domains. Suppose N_8 is involved in a dynamic change of membership. Both join and leave can be dealt with by replacing (t_8, v_8) with a new pair of (T_8, V_8).

Remark 1. We assume the nodes (i.e., routers with cryptographic processing capability) are trusted and secure. The trust assumption can be addressed by letting a node N_j choose a unique authentication key v_{js} for its child N_s.

2. The MAC keys have to be encrypted while transmitting over the multicast tree. For the same reason, it is insecure

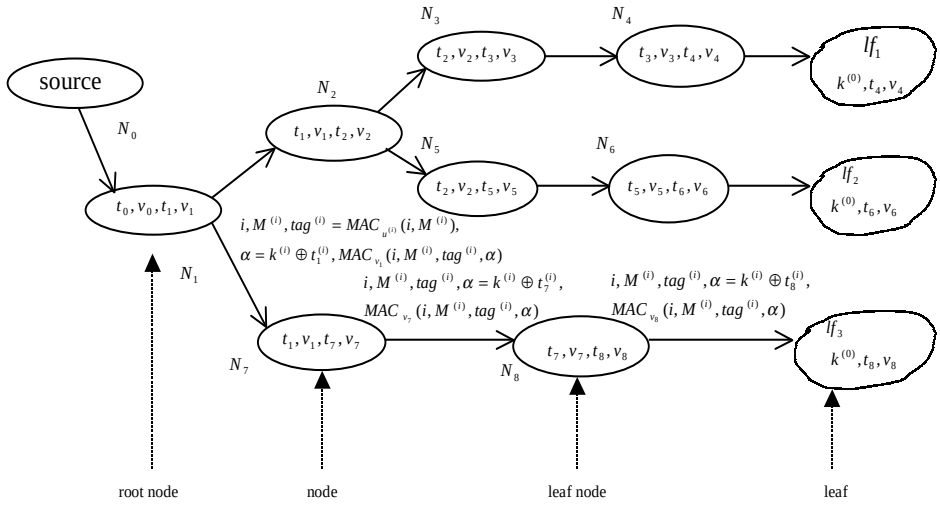

Figure 1: The i^{th} packet $M^{(i)}$ flows through the example multicast tree in Scheme 1

to adopt an end-to-end encryption for the MAC keys. On the other hand, encryption of MAC keys enables us to implement access control.

3. Hop-by-hop authentication of multicast packets is for implementing immunity to multicast DOS attack, whereas authentication of broadcast packets is for blocking the attack by an adversary attached to a different leaf node.

4. This scheme is not secure against colluding receivers that are attached to different leaf nodes. To avoid this, we can assume that there is a synchronization clock (as in [13]). Note that TESLA is not secure without a synchronization clock, even if there are no colluding receivers.

4.2 Scheme 2

We assume the same multicast tree as in Scheme 1.

4.2.1 The Construction

Initialization. The participants execute as follows.

- The source, N_0, generates and sends a MAC key k to the legitimate receivers via private channels. It also chooses and sends a random secret v_0 to N_1.
- A non-leaf node N_j ($j \in \{1, 2, 3, 5, 7\}$) chooses and sends a random secret v_j to its children. N_j sets $i' \leftarrow 0$.
- A leaf node N_l ($l \in \{4, 6, 8\}$) chooses a key chain $q_l^{(0)}$, $q_l^{(1)}$, $\cdots$, $q_l^{(w)}$, where $q_l^{(i-1)} = f_{q_l^{(i)}}(0)$ for $1 \leq l \leq w$. N_l sets $i' \leftarrow 0$.
- Each receiver sets $i' \leftarrow 0$ and $q_l^{(i')} \leftarrow q_l^{(0)}$.

Runtime. We show how the i^{th} ($1 \leq i \leq w$) packet, $M^{(i)}$, flows through the third path in Figure 2. Recall that N_0 keeps (w, k, v_0), N_1 keeps $(i' = 0, v_0, v_1)$, N_7 keeps $(i' = 0, v_1, v_7)$, N_8 keeps $(i' = 0, v_7, q_8^{(0)}, \cdots, q_8^{(w)})$, and Alice keeps $(i' = 0, k, q_8^{(0)})$.

1. The source, N_0, computes $tag^{(i)} = MAC_k(i, M^{(i)})$ and $\beta = MAC_{v_0}(i, M^{(i)}, tag^{(i)})$. Then, it sends $(i, M^{(i)}, tag^{(i)}, \beta)$ to the root node N_1.
2. N_1 receives $(i, M^{(i)}, tag^{(i)}, \beta)$. N_1 discards it if $i \leq i'$ or $MAC_{v_0}(i, M^{(i)}, tag^{(i)}) \neq \beta$. Otherwise, N_1
   (a) computes $\beta^* = MAC_{v_1}(i, M^{(i)}, tag^{(i)})$ and sends $(i, M^{(i)}, tag^{(i)}, \beta^*)$ to N_2 and N_7;
   (b) sets $i' \leftarrow i$.
3. N_7 receives $(i, M^{(i)}, tag^{(i)}, \beta)$. N_7 discards it if $i \leq i'$ or $MAC_{v_1}(i, M^{(i)}, tag^{(i)}) \neq \beta$. Otherwise, N_7
   (a) computes $\beta^* = MAC_{v_7}(i, M^{(i)}, tag^{(i)})$ and sends $(i, M^{(i)}, tag^{(i)}, \beta^*)$ to N_8;
   (b) sets $i' \leftarrow i$.
4. N_8 receives $(i, M^{(i)}, tag^{(i)}, \beta)$. N_8 discards it if $i \leq i'$ or $MAC_{v_7}(i, M^{(i)}, tag^{(i)}) \neq \beta$. Otherwise, N_8
   (a) computes $v_8^{(i)} = f_{q_8^{(i)}}(1)$ and $\beta^* = MAC_{v_8^{(i)}}(i, M^{(i)}, tag^{(i)})$;
   (b) broadcasts $(i, M^{(i)}, tag^{(i)}, \beta^*, q_8^{(i)})$;
   (c) sets $i' \leftarrow i$.
5. Alice receives $(i, M^{(i)}, tag^{(i)}, \beta, q_8^{(i)})$. She discards it if $i \leq i'$ or $q_8^{(i')} \neq f^{i-i'}(q_8^{(i)})$ or $MAC_{v_8^{(i)}}(i, M^{(i)}, tag^{(i)}) \neq \beta$, where $v_8^{(i)} = f_{q_8^{(i)}}(1)$. She also discards it if $tag^{(i)} \neq MAC_k(i, M^{(i)})$; otherwise, she accepts the packet and sets $i' \leftarrow i$ as well as $q_8^{(i')} \leftarrow q_8^{(i)}$.

4.2.2 Properties of the Scheme

- Security. Suppose the nodes are trusted and secure, this scheme is provably secure (see Theorem 3 in the appendix).
- Real-time authentication. A receiver can immediately authenticate any packet regardless of the packet loss characteristics of the underlying network.

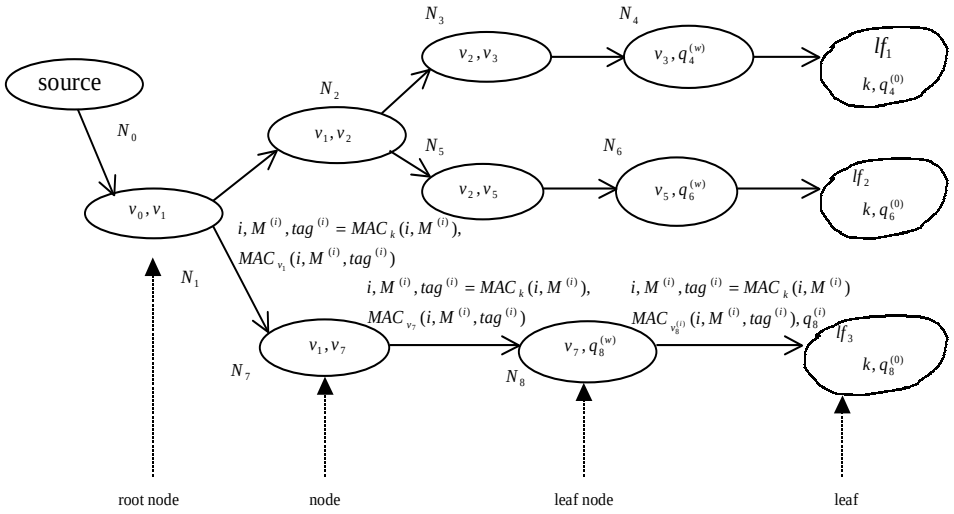

Figure 2: The i^{th} packet $M^{(i)}$ flows through the example multicast tree in Scheme 2

- The computational overhead is at most $2h + 7$ PRFs per packet, where h is the maximum number of nodes among all the paths. When we use 80-bit MAC for both the MAC key and the MAC value, the communication overload is at most 34 bytes per packet.
- Immunity to multicast DOS attack. This is the same as in Scheme 1.
- Dynamic maintenance. The source distributes a new MAC key K in place of the original MAC key k.

5. CONCLUSION

We have presented two IP multicast authentication schemes by making use of the multicast tree as an essential authentication mechanism. Our schemes are efficient and immune to multicast denial-of-service attack. They allow the receivers to immediately authenticate the packets regardless of the packet loss characteristics of the underlying network. Scheme 1 has the nice property that dynamic maintenance can be confined within the corresponding leaf domains. Scheme 2 is indeed a multi-party counterpart of two-party MAC in the sense that there is a unique MAC key shared among the source and the receivers, while the multicast tree guarantees that no honest receiver will be fooled to accept a bogus packet.

6. ACKNOWLEDGMENTS

We thank the anonymous reviewers for useful comments.

7. REFERENCES

[1] A. Ballardi. Scalable multicast key distribution. In *Request for Comment (RFC) 1949*. IETF, 1996.

[2] D. Boneh, G. Durfee, and M. Franklin. Lower bounds for multicast message authentication. In *Proceedings of Eurocrypt'01*, 2001.

[3] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas. Multicast security: A taxonomy and some efficient constructions. In *Proceedings of INFOCOM'99*, pages 708–716. IEEE, 1999.

[4] W. Fenner. Internet group management protocol, version 2. In *Request for Comment (RFC) 2236*. IETF, 1997.

[5] R. Gennaro and P. Rohatgi. How to sign digital streams. In *Proceedings of Crypto'97*, 1997.

[6] P. Golle and N. Modadugu. Authentication schemes for online streams. In *Proceedings of NDSS*, 2001.

[7] H. W. Holbrook and D. R. Cheriton. Ip multicast channels: EXPRESS support for large-scale single-source applications. In *Proceedings of SIGCOMM*, pages 65–78. ACM, 1999.

[8] S. Miner and J. Staddon. Graph-based authentication of digital streams. In *Proceedings of Security and Privacy*, pages 277–288. IEEE, 2001.

[9] S. Mittra. Iolus: A framework for scalable secure multicasting. In *Proceedings of SIGCOMM*, pages 277–288. ACM, 1997.

[10] R. Molva and A. Pannetrat. Scalable multicast security in dynamic groups. *ACM Tran. on Info. and Sys. Sec.*, 3(3):136–160, 2000.

[11] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of STOC*, pages 33–43. ACM, 1989.

[12] A. Perrig, R. Canetti, D. Song, and J. D. Tygar. Efficient and secure source authentication for multicast. In *Proceedings of NDSS*. ISOC, 2001.

[13] A. Perrig, R. Canetti, J. D. Tygar, and D. Song. Efficient authentication and signing of multicast streams over lossy channels. In *Proceedings of Security and Privacy*, pages 56–73. IEEE, 2000.

[14] P. Rohatgi. A compact and fast hybrid signature scheme for multicast packet authentication. In *Proceedings of CCCS*, pages 93–100. ACM, 1999.

[15] C. K. Wong, M. Gouda, and S. S. Lam. Secure group communications using key graphs. In *Proceedings of SIGCOMM*, pages 68–79. ACM, 1998.

[16] C. K. Wong and S. S. Lam. Digital signatures for flows and multicasts. In *Proceedings of ICNP*, pages 198–209. IEEE, 1998.

APPENDIX

THEOREM 1. *Suppose that the nodes are trusted and secure, and that there are no colluding receivers attached to different leaf nodes. Then, no honest receiver can be fooled to accept a bogus packet.*

PROOF. (sketch) In a real-world system, the MAC keys are derived from PRF family $\{f_k\}$. These keys are encrypted (while flowing over the multicast tree) with functions from PRF family $\{g_k\}$. Suppose an adversary $\mathcal{A}$ succeeds (in fooling an honest receiver to accept a bogus packet) with non-negligible probability ε. We first construct a simulation SIM-1 which is the same as a real-world system, except that all the encryption functions are replaced with random functions (RFs). Denote τ the probability that $\mathcal{A}$ succeeds in SIM-1. If $\tau \geq \varepsilon/2$, we construct a simulation SIM-2 whereby we distinguish a PRF from a RF with probability at least $\varepsilon/2w$, where w is the number of packets. If $\tau < \varepsilon/2$, we construct a simulation SIM-3 whereby we distinguish a PRF from a RF with probability at least $\varepsilon/2z$, where z is the number of nodes in the multicast tree. In either case, the success probability for distinguishing a PRF from a RF is non-negligible.

SIM-1

1. Let the simulator choose the secrets on behalf of the participants as in the real-world, except that encryption functions of the source and the nodes in $\{N_0, \cdots, N_{j-1}, N_{j+1}, \cdots, N_z\}$ are instantiated from RFs, where N_j is the leaf node to which $\mathcal{A}$ is attached.
2. Let the simulator run the system as in the real-world, except that all the encryptions $\alpha = k^{(i)} \oplus t_l^{(i)}$ are replaced with encryptions using RFs, where $0 \leq l \leq z$ and $l \neq j$. All the public transcripts are presented to $\mathcal{A}$.

$\mathcal{A}$ succeeding in SIM-1 means, due to the atomicity of Ethernet broadcast, that $\mathcal{A}$ must present a MAC value valid with respect to a MAC key on the key chain before N_j broadcasts the (encrypted) MAC key. If $\mathcal{A}$ succeeds in SIM-1 probability $\tau \geq \varepsilon/2$, then we construct a simulation SIM-2 whereby we distinguish a PRF from a RF with probability at least $\varepsilon/2w$.

SIM-2

1. Suppose the simulator is given black-box access to F – either a PRF or a RF with equal probability. Let the simulator choose $b \in_R \{1, \cdots, w\}$ and bet that $\mathcal{A}$ will succeed in fooling an honest receiver to accept a bogus packet instead of the b^{th} packet. Let the simulator choose $k^{(b-1)} = F(0)$ and define $k^{(j-1)} = f_{k^{(j)}}(0)$ for $1 \leq j \leq b-1$. All the other system parameters are generated in the same way as in SIM-1.
2. For the first $b-1$ packets, the simulator executes as in SIM-1.
3. When the b^{th} packet, $M^{(b)}$, is generated, it should be associated with $tag^{(b)} = MAC_{u^{(b)}}(i, M^{(b)})$ where $u^{(b)} = f_{k^{(b)}}(1)$. Instead, the simulator lets $tag^{(b)} = F(i, M^{(i)})$. If $\mathcal{A}$ succeeds in presenting a tag valid with respect to oracle $F(\cdot)$, the simulator returns 1 (i.e., F is a PRF), and 0 otherwise.

If F is a random function, $\mathcal{A}$ succeeds with only negligible probability. Since $\mathcal{A}$ succeeds in SIM-1 with probability $\tau \geq \varepsilon/2$, we can distinguish a PRF from a RF with probability at least $\varepsilon/2w$.

Now, suppose $\mathcal{A}$ succeeds in SIM-1 with probability $\tau < \varepsilon/2$. We construct another simulation SIM-3 whereby we distinguish a PRF from a RF with probability at least $\varepsilon/2z$.

In order to use the hybrid argument, we define a total order (e.g., depth first) $N_1^* < N_2^* < \cdots < N_z^*$ over the set of the source and the nodes, except the leaf node N_j to which $\mathcal{A}$ is attached. Define $\mathcal{EXPT}_l$ the experiment that the first l encryption functions in the order are instantiated from PRFs and the rest are instantiated from RFs. Denote p_l the probability that the adversary succeeds in fooling an honest receiver to accept a bogus packet in $\mathcal{EXPT}_l$. Since $\mathcal{EXPT}_0$ is exactly SIM-1 and $\mathcal{EXPT}_z$ is exactly the real-world system, we have $p_0 < \varepsilon/2$ and $p_z \geq \varepsilon$. For $l \in_R \{0, \cdots, z-1\}$, the probability that the adversary can be used to distinguish a PRF from a RF is $\frac{1}{z}\sum_{i=0}^{z-1}(p_{i+1} - p_i) \geq \varepsilon/2z$. Now, we present the simulation SIM-3.

SIM-3

1. Let the simulator choose $N_c \in_R \{N_1^*, \cdots, N_z^*\}$ and embed the challenge F (i.e., either a PRF or a RF with equal probability) as the encryption function of node N_c. Let the encryption function of $N^* < N_c$ be instantiated from a PRF, and the encryption function of $N^* > N_c$ be instantiated from a RF. The encryption and decryption operations of N_c are done via having oracle access to F.
2. Let the simulator run the system as in SIM-1.

If the adversary succeeds in SIM-3, the simulator returns 1 (i.e., F is a PRF), and 0 otherwise. Therefore, we can distinguish a PRF from a RF with probability at least $\varepsilon/2z$.

Note that the above argument fails, if the adversary finds $K^{(i)}$ such that $K^{(i)} \neq k^{(i)}$ yet $f_{K^{(i)}}(0) = f_{k^{(i)}}(0)$ and $f_{K^{(i)}}(1) = f_{k^{(i)}}(1)$, for some $0 < i \leq w$. However, this breaks the assumption that $\{f_k\}$ is target collision resistant. □

THEOREM 2. *Suppose the nodes are trusted and secure. Then, the scheme is immune to multicast DOS attack.*

PROOF. (sketch) Suppose an adversary succeeds in breaking the "immunity to multicast DOS attack" with non-negligible probability ε. We can simply guess a random node N with respect to which the adversary succeeds in fooling its children to forward a bogus packet. Then, we can simulate the real-world system by embedding the challenge F (i.e., either a PRF or a RF with equal probability) as N's MAC function, while choosing all the other system parameters as in the real-world. If F is a random function, the adversary can succeed in fooling N 's children to forward a bogus packet with only negligible probability. Thus, we can distinguish a PRF from a RF with probability at least $\varepsilon/(z+1)$, where z is the number of nodes in the multicast tree. □

THEOREM 3. *Suppose the nodes are trusted and secure. Then, no honest receiver can be fooled to accept a bogus packet.*

PROOF. (sketch) Due to the atomicity of Ethernet broadcast, there are only two cases in which an adversary can fool an honest receiver to accept a bogus packet.

- The bogus packet is broadcast by the corresponding leaf node, which means that the leaf node is fooled to accept the bogus packet. This happens with only negligible probability; otherwise, immunity to multicast DOS attack is broken.
- The bogus packet is broadcast by the adversary, which means that the adversary presents with non-negligible probability ε a MAC value that is valid with respect to a key on the key chain $q_j^{(i)}$ chosen by leaf node N_j. We construct a simulation whereby we distinguish a PRF from a RF with non-negligible probability.

Suppose the simulator is given a black-box access to F – either a PRF or a RF with equal probability.

1. Let the simulator choose $b \in_R \{1, \cdots, w\}$ and bet that the adversary will succeed in fooling an honest receiver to accept a bogus packet instead of the b^{th} packet. Let the simulator define $q_j^{(b-1)} = F(0)$ and $q_j^{(i-1)} = f_{q_j^{(i)}}(0)$ for $1 \leq i \leq b-1$. All the other system parameters are generated as in the real-world.
2. For the first $b-1$ packets, the simulator executes as in the real-world system.
3. When the b^{th} packet $M^{(b)}$ is generated, it should be associated with $MAC_{v_j^{(b)}}(i, M^{(b)}, tag^{(i)})$ where $v_j^{(b)} = f_{q_j^{(b)}}(1)$. Instead, the simulator substitutes it with $F(i, M^{(b)}, tag^{(i)})$. If the adversary succeeds, the simulator returns 1 (i.e., F is a PRF), and 0 otherwise.

If F is a random function, the adversary succeeds with only negligible probability. Therefore, we can distinguish a PRF from a RF with probability at least ε/wL, where L is the number of leaf nodes. □