Efficient Non-Repudiation Multicast Source Authentication Schemes

LI Xianxian (李先贤) and HUAI Jinpeng (怀进鹏)

School of Computer, Beijing University of Aeronautics and Astronautics, Beijing 100083, P.R. China

E-mail: lixx@cscw.buaa.edu.cn

Received February 21, 2001; revised December 13, 2001.

Abstract In secure multicast communication, packet source authentication is a bottleneck problem due to the dynamic property of the multicast group, unreliability of data transmission and the large number of data packets. This paper proposes a novel authentication scheme called B-MSAS (Balance Multicast Source Authentication Scheme) that can be used to solve this problem, in which a new message authentication technique is introduced. This scheme dramatically reduces the signature size overhead and raises the signature rate. It provides the non-repudiation service, high loss resistance, and can easily be scaled up to potentially millions of receivers, and hence has a sweeping applicability. It should have applications to many practical problems.

Keywords multicast communication, digital signature, non-repudiation, message authentication code

1 Introduction

The popularity of multicast has grown in recent years with the wide use of the Internet. Examples include Internet video transmissions, stock quotes, live multi-party conferencing and long-distance education. Like many unicast applications, most of the multi-party applications listed above will only be successful if the privacy and authenticity of participants can be provided efficiently. Packet source authentication is a fundamental security issue in many multicast protocols. However, maintaining authenticity in multicast protocols is a much more complex problem than that for unicast, and it is a serious bottleneck in multicast security. In the case of unicast, this problem has been solved (e.g., in IPSEC) by the use of message authentication codes (MACs) and public key signatures. However, the MAC approach is inadequate in a multicast setting. This is because an MAC is based on a shared secret among participants and MACs can be both generated and verified by anyone having access to the key. And the public key signature approach is inadequate since it requires significant computational overhead for computing the signature. As an example, even a fast machine (such as 200MHz power PC) can generate only 40 or so 1024 bits RSA signatures per second. Clearly, some multicast applications can require a packet rate far exceeding 40 packets/s. Thus this solution is not practically feasible.

The paper is organized as follows. In Section 2 we review the related work. Section 3 presents some secure issues of multicast source authentication. Section 4 introduces a new concept "Message Authentication Code Matrix" (MACM). Section 5 gives a novel multicast source authentication scheme with low communication overhead (LCO-MSAS) by the use of MASM, and in this section its efficiency and security are analyzed. In Section 6, by improving LCO-MSAS, we propose another efficient scheme B-MSAS (Balance Multicast Source Authentication Scheme) which balances between the computation overhead and the communication overhead, and in this section we compare our work with some related representative schemes. Section 7 concludes the paper with some remarks.

The reader is referred to [1] for concepts and terminologies not explained in this paper.

This work is supported by the National Natural Science Foundation of China (Grant No.60073006) and the National '863' High-Tech Program of China (Grant No.2001AA144150).

2 Related Work

If the reliability of transmission is not an issue, there is an approach known as stream signing[2] that can be used to sign multicast packets efficiently and provide security guarantees associated with digital signatures. In this approach, only one public key signature is transmitted at the beginning of a stream and each packet contains either a hash value of the next packet in the stream or a 1-time public key by which the 1-time signature on the next packet can be verified. This approach is very efficient in reliable Internet protocols (such as those based on TCP/IP). However, this approach cannot tolerate a lost packet, since the information needed to authenticate future packets will be lost, thus it is incompatible with IP multicast. In [3] some different proposals for dealing with packet loss were presented where other chaining schemes were considered. Recently, to achieve robustness against packet loss, [4] proposed two multicast stream authentication schemes, TESLA and EMSS. TESLA, short for Timed Efficient Stream Loss-tolerant Authentication, uses only symmetric cryptographic primitives, and is based on timed release of keys by the sender. This approach offers loss robustness, high scalability, and minimal overhead, at the cost of loose initial time synchronization and delayed authentication. However, TESLA does not provide non-repudiation. Another scheme EMSS (Efficient Multi-chained Stream Signature) in [4] is based on signing a small number of special packets in a data stream; each packet is linked to a signed packet via multiple hash chains. EMSS guarantees that almost all arriving packets can be authenticated, even over highly lossy channels, however, the communication overhead is higher as the packet loss probability is higher.

In [5] another efficient approach to multicast source authentication is proposed where the concept of "asymmetric MACs" is introduced. The basic idea in this approach is that the sender knows several secret MAC keys and these keys are shared with the recipients in a way so as to maintain several properties of the subsets of keys held by the recipients. For example, no collection of w receivers should know all the keys known by any other receiver. When the sender sends a message it authenticates the message by computing MACs using all its keys and appending all the MACs to the message. Each recipient verifies all the MACs which are created using the keys in its subset and if all these MACs are correct then the receiver accepts the message as genuine. From the property of the subsets described above, even w receivers cannot collude to forge the MACs to fool some other recipient. However once there are more than w colluders the security of the scheme will break down[6]. Clearly the number of MACs computed by the sender has to be a linear function of the number of colluders. Therefore this scheme can work well only in scenarios where groups are small and problems of collusion can be controlled. Furthermore, it is also a difficult task for the sender to distribute securely the subsets of the keys to the receivers especially when groups are dynamic, which may cause other secure problems.

In [6] Pankaj presents a new scheme which reduces the communication overhead of on-time signatures compared with previously proposed off-line/on-line signature schemes. In the off-line/on-line approach, off-line computation is used to create buffers of 1-time key-pairs and to certify the public 1-time keys using the regular digital signature scheme. When a message needs to be signed then on-line computation is performed to compute a signature of the message using a 1-time private key from the buffer of keys. Since operations on 1-time keys are extremely fast, the most expensive operation is performed off-line, and thus this scheme can achieve a favorable signature rate. Moreover, [6] introduces a k-time signature scheme, which is more space-efficient than the 1-time signature. Despite all advantages, the scheme still uses 90 bytes for a 6-time public key (which does not include the certificate of the public key) and 300 bytes for each signature. Besides, the server requires 350 off-line hash function applications and the client needs 184 hashes on an average to verify the signature.

3 Security Issues of Multicast Source Authentication

In this section we present the multicast source authentication problems. Different from the two-party communication, multicast communication may involve several senders and receivers. Packet source authentication means that each recipient can verify the origin of messages. Multicast source authentication may be classified into two kinds: *multiple sources authentication*, which means there are several senders and each receiver can recognize all senders that have sent messages; and *single* *source authentication*, which means there is only one party that sends messages and it is possible to identify the particular sender within the multicast group, which is commonly referred to as "one to many". In this paper we will discuss mainly the latter.

Here the Multicast Source Authentication Scheme (denoted by MSAS) is always referred to single source authentication unless there is explanation in particular.

Security requirements

Similar to the unicast case, an MSAS includes a signing algorithm and a verifying algorithm. However, there may be many verifying keys corresponding to a signing key in an MSAS, which is different from the unicast case in which a signing key corresponds to only one verifying key. For example, a message signed with one signing key can be verified with multiple distinct verifying keys in the "asymmetric MAC" scheme proposed in [5].

Roughly speaking, *security of MSAS* means that any adversary who does not know the signing key has a negligible probability to generate the right authentication information of any message.

Generally, *non-repudiation* means that message receivers are capable of proving, to third parties, that a message has been transmitted[5]. This implies that an MSAS is non-repudiation if and only if

(1) the MSAS is secure;

(2) the signing key of the message sender is secret and its corresponding verifying keys are certified by the trusted third party.

Examples

Schemes based on a shared secret MAC key among participants are not non-repudiation because the signing key and the verifying key are the same and each authorized receiver is capable of generating the right authentication information of any message besides the message sender.

In the "asymmetric MAC" authentication scheme in [5], a sufficient number of receivers can collude to get the signing key, hence this scheme cannot provide the non-repudiation service.

The public key signature scheme is non-repudiation if the public key of the sender has been certified by the trusted third party.

Performance

Performance is a major concern for multicast security applications. The efficiency of an MSAS is mainly evaluated according to the following overheads:

- *Computation overhead.* It is the work time for computing the authentication information of the packets to be sent and verifying the received data packets.
- *Communication overhead.* It is the bandwidth overhead incurred by inflating data packets via cryptographic transformations.

To achieve a high efficiency, the above two overheads should be minimized as much as possible.

For an MSAS, both security and efficiency are essential. Unfortunately, it seems that they are difficult to be satisfied simultaneously. For example, the MAC authentication scheme enjoys a high efficiency, but its security is weak. The scheme proposed in [6] is non-repudiation, however its efficiency is low.

4 Message Authentication Code Matrix

Our schemes are based on a novel message authentication algorithm. For convenience, we first introduce the authentication algorithm in this section. Firstly, let us give some notions. In the remainder of this paper, M always denotes the set of all possible messages (generally assume $M = \{0,1\}^*$).

Target Collision Resistance Functions[4,6]: An efficiently computable function H: $M \to \{0,1\}^l$ (where l is called the output length of H) is target collision resistance if, for any $m_0 \in M$ chosen at random, the adversary is assumed to be unable (or with negligible probability) to find $m \neq m_0$ such that $H(m) = H(m_0)$. For example, universal one-way functions and MAC functions are target collision resistance.

Pseudo-Random Functions[7]: Pseudo-random functions were introduced by Goldreich, Goldwasser & Micali and were a very well studied subject in cryptography. A distribution of functions is pseudo-random if: (1) this distribution is efficient; (2) it is hard to tell a function sampled according to this distribution from a uniformly distributed function given an adaptive access to the function as a black-box.

Pseudo-random functions have numerous applications. However, in this paper we are interested in the following properties of pseudo-random functions. Suppose $F = \{f_i\}_{i \in A}$ is a pseudo-random function family where each f_i is a pseudo-random function from M to $\{0,1\}^k$ (which is called a k-bit output function). Then the following properties hold.

(1) For a fixed $\sigma \in M$, the probability that any adversary can choose a $\sigma' \in M$ such that $f_i(\sigma) = f_i(\sigma')$ and $\sigma \neq \sigma'$ is

$$\Pr(f(\sigma) = f(\sigma')) < \frac{1}{2^k} + \varepsilon$$

where ε is negligible①.

(2) For any $i \neq j$, the function f_i is independent of f_j.

Notation. Let f be a pseudo-random function. We write $f(x,y) = f(x\|y)$ for $x, y \in M$, where "$\|$" denotes concatenation.

Definition 1. *Suppose that $F = \{f_{ij}\}_{1 \leq i \leq n, 1 \leq j \leq m}$ is a pseudo-random function family, where each f_{ij} is a k-bit output function. For any $x_1, x_2, \ldots, x_n$, $y_1, y_2, \ldots, y_m \in M$, we define a "matrix multiplicative" operation F by the following formula*

$$F\left(\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} (y_1\ y_2\ \cdots\ y_m)\right) = \begin{pmatrix} f_{11}(x_1,y_1) & f_{12}(x_1,y_2) & \cdots & f_{1m}(x_1,y_m) \\ f_{21}(x_2,y_1) & f_{22}(x_2,y_2) & \cdots & f_{2m}(x_2,y_m) \\ \vdots & \vdots & \vdots & \vdots \\ f_{n1}(x_n,y_1) & f_{n2}(x_n,y_2) & \cdots & f_{nm}(x_n,y_m) \end{pmatrix}.$$

Let $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_m)$, then the above formula can be written as $F(X,Y)$.

Theorem 1. *Suppose $F = \{f_{ij}\}_{1 \leq i \leq n, 1 \leq j \leq m}$ is a pseudo-random function family as described above, where each f_{ij} is a k-bit output function. Given an (n,m)-matrix $C = (c_{ij})_{n \times m}$ where each $c_{ij} \in \{0,1\}^k$, any adversary (whose resources are bounded by a polynomial in lk where $l = \min\{n,m\}$) is able to find $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_m)$ (where $x_i, y_j \in M$) such that $F(X,Y) = C$ only with negligible probability.*

Proof. It is stated in detail in the Appendix.

Corollary 1. *The "matrix multiplicative" operation F defined in Definition 1 is target collision resistance. That is, let $F = \{f_{ij}\}_{1 \leq i \leq n, 1 \leq j \leq m}$ be a pseudo-random function family as described above, and let $\alpha = (a_1, a_2, \ldots, a_n)$, $\beta = (b_1, b_2, \ldots, b_m)$ and $C = F(\alpha, \beta)$, then any adversary is capable of finding $X = (x_1, x_2, \ldots, x_n)$, $Y = (y_1, y_2, \ldots, y_m)$ and $(X,Y) \neq (\alpha,\beta)$ such that $F(X,Y) = C$ only with negligible probability.*

Proof. It follows immediately from Theorem 1.

Definition 2. *In Corollary 1 we call the matrix $C = (c_{ij})_{n \times m}$ a Message Authentication Code Matrix (MACM) for the messages $\{a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m\}$.*

To achieve the security of MACM, the natural number $N_0 (= \min\{nk, mk\})$ has to be sufficiently large by Theorem 1, which depends on the capability of the adversary that we assume. Then we call N_0 the *Secure Base Number* for MACM. From the above discussions, it is clear that an MACM function with the secure base number N_0 can offer a level of security of the hash function with an N_0-bit output. Commonly, the security of 80-bits is considered adequate. And it is considered sufficient that $N_0 \geq 128$ for the applications with stronger security requirements.

The MACM possesses even more properties as described below.

Suppose that C is an MACM for message packets $a_1, a_2, \ldots, a_n$, $b_1, b_2, \ldots, b_m$. Assuming that only some packets $\{a_{i_1}, \ldots, a_{i_s}, b_{j_1}, \ldots, b_{j_t}\}$ (where $\{i_1, \ldots, i_s\} \subseteq \{1, 2, \ldots, n\}$ and $\{j_1, \cdots, j_t\} \subseteq \{1, 2, \ldots, m\}$) have been received, it is sufficient to allow the receiver to validate the information if $ki_s \geq N_0$ and $kj_t \geq N_0$. This property is used against packet loss in our schemes.

From the discussions in the Appendix, Theorem 1 is also true when we substitute MAC functions for the pseudo-random functions $\{f_{ij}\}$. In our schemes, MAC functions with a k-bit output are used.

①We say ε is negligible if for every polynomial f and sufficiently large K's $0 \leq \varepsilon < \frac{1}{f(K)}$[7].

5 Low Communication Overhead MSAS (LCO-MSAS)

In this section we propose an MSAS with very low communication overhead (abbreviated as LCO-MSAS) by using the MACM technique. Furthermore, we discuss its security and efficiency.

Our scheme is based on the following primitives.

- Let H be a collision resistant hash function producing a fixed length of output such as 80 bits or 128 bits.
- Assume that A is the sender of messages. A possesses a public key certification, and SK_A and PK_A is its private key and public key, respectively, for example RSA. Sig_{SK_A} is the Public Key Signature algorithm by using A's private key SK_A, and Ver_{PK_A} is the corresponding verifying algorithm by using A's public key PK_A.
- Let $F = \{f_{ij}\}_{1\leq i\leq n, 1\leq j\leq m}$ be a function family, where each $f_{ij} : M \rightarrow \{0,1\}$ is a single bit output MAC function[7]. Suppose that all receivers are aware of F.
- Let N_0 be the secure base number, such as $N_0 = 80$ or 128, depending on the various applications.

5.1 Description of LCO-MSAS

The basic idea is that the multicast sender divides the message packets to be sent into many chains, and each chain is a sequence of message blocks which consist of some packets. In other words, a chain is a message set $M = \bigcup\limits_{i=1}^{l} P_i$, where each $P_i = \{p_{ij}|j = 1,\ldots,n\}$ (called a message block) is a set consisting of some packets. l is called the length of the chain, and n the width of the chain. By aptly choosing the length and width of the chain, it can achieve a high probability that the number of remainder packets in every block is not less than N_0. It depends on the loss probability to choose the length and width of the chain, which is stated in detail in the next section. Here, we first describe the signature generation and the signature verification for a chain.

To simplify the description, here, only the algorithm of the signature is stated and the details of signing performance are not given.

In a chain $M = \bigcup\limits_{i=1}^{l} P_i$, where $P_i = \{p_{ij}|j = 1,\ldots,n\}$, each packet p_{ij} is numbered (i,j).

Signature Generation Process

1) Firstly, let $k_{lj} = H(p_{lj})$ for every $p_{lj} \in P_l$, where $j = 1,\ldots,n$. Then k_{lj} is the authentication information of p_{lj}, and it is appended to p_{lj}.

2) For $1 \leq i \leq l-1$ and $1 \leq j \leq n$, compute $c^i_{rj} = f_{rj}\ (k_{i+1,r}, p_{ij})$ where $p_{ij} \in P_i$ and $r = 1,2,\ldots,n$, and let $k_{ij} = (c^i_{1j}\|c^i_{2j}\|\ldots\|c^i_{nj}) \in \{0,1\}^n$. Then k_{ij} is the authentication information of p_{ij}.

3) Compute the signature $s_{1j} = Sig_{SK_A}(k_{1j})$ of k_{1j} using A's private key SK_A for $j = 1,\ldots,n$, and it is sent with the packet p_{1j}.

The process of signing can also be illustrated by Fig.1.

Signature Verification Process

Assume that the set of packets that have been received is $M' = \bigcup\limits_{i=1}^{l} P'_i$, where $P'_1 = \{p'_{1j} = (p_{1j}, k_{1j}, s_{1j})\}$ and $P'_i = \{p'_{ij} = (p_{ij}, k_{ij})\}$ $(i = 2,\ldots,l)$ in which the subscript (i,j) is the serial number of a corresponding packet (remember that each packet signed has been numbered), k_{ij} is the authentication information of p_{ij}, and s_{1j} is the signature of k_{1j}. To verify these packets, first check the number of packets in every block P_i. If there is a block P_i such that $|P_i| < N_0$, it is considered that the verification fails, otherwise do the following:

1) Verify whether $Ver_{PK_A}(k_{1j}, s_{1j}) = T$ for all j using A's public key PK_A.

2) Let $J_i = \{j|$ there exists $p_{ij} \in P'_i\}$ for $1 \leq i < l$. Then compute and verify whether

$$(k_{ij})_r = f_{rj}(k_{i+1,r}, p_{ij})$$

for all $j \in J_i$ and $r \in J_{i+1}$, where $(k_{ij})_r$ denotes the r-th bit of k_{ij}.

3) Compute $k_{lj} = H(p_{lj})$ for all $j \in J_l$.

If the above verifying processes are legitimate, then the recipient accepts all the packets as truth.

Correctness

Note that the probability that the remainder packets are sufficient to be verified is very high. In this case, it is clear that every correct packet can be accepted by any authorized recipient.

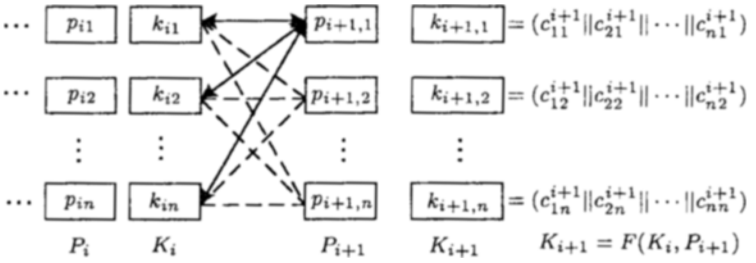

$\cdots$ p_{i1}
k_{i1}
$p_{i+1,1}$
$k_{i+1,1}$ $= (c_{11}^{i+1} \| c_{21}^{i+1} \| \cdots \| c_{n1}^{i+1})$
$\cdots$ p_{i2}
k_{i2}
$p_{i+1,2}$
$k_{i+1,2}$ $= (c_{12}^{i+1} \| c_{22}^{i+1} \| \cdots \| c_{n2}^{i+1})$
$\cdots$ p_{in}
k_{in}
$p_{i+1,n}$
$k_{i+1,n}$ $= (c_{1n}^{i+1} \| c_{2n}^{i+1} \| \cdots \| c_{nn}^{i+1})$
P_i
K_i
P_{i+1}
K_{i+1}
$K_{i+1} = F(K_i, P_{i+1})$

Fig.1

5.2 Security Analysis of LCO-MSAS

5.2.1 *Proof of Security*

Theorem 2. *Assume that the public key signature system used in LCO-MSAS is secure, then LCO-MSAS is a secure MSAS.*

Proof. The security of this scheme can be roughly explained by the following statements:

From the description of LCO-MSAS, $K_{i+1} = F(K_i, P_{i+1})$ for $1 \leq i \leq l-1$, then the collision resistance of F implies that K_i and P_{i+1} are true only if K_{i+1} is true. Finally, the truth of K_l is guaranteed by the public key signature system.

Its formal proof is presented in detail in the Appendix. □

5.2.2 *Non-Repudiation*

In LCO-MSAS, the signing key is SK_A and the verifying key is PK_A. Note that the verifying key PK_A has been certified by a third party (CA), and hence the security of LCO-MSAS implies that it is non-repudiation.

5.3 Performance and Overhead Analysis

Now we discuss how to choose the length and width of a chain.

To simplify the following discussion, we firstly assume that the packet loss is independent. Then we have the following lemma.

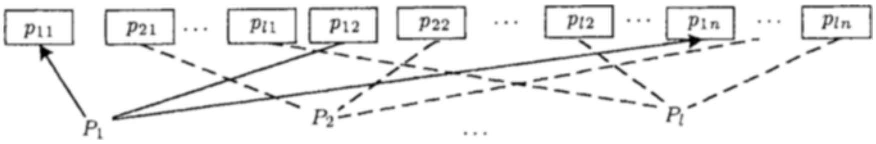
Notation. Let A be a set, then $|A|$ denotes the number of elements in A.

Lemma 1. *Let $P = \{p_i | i = 1, \ldots, n\}$ be a set consisting of n packets. Suppose that P is transmitted on an unreliable network where the loss probability of each packet is w $(0 \leq w < 1)$, and that P' is the set of remainder packets received. Let N_0 $(< n)$ be a natural number, then the probability that $|P'| < N_0$ is $\sum\limits_{k=n-N_0+1}^{n} C_n^k w^k (1-w)^{n-k}$. Therefore, if a message chain $M = \bigcup\limits_{i=1}^{l} P_i$ (where $P_i = \{p_{ij} | j = 1, \ldots, n\}$) is transmitted on such a network, then the probability, that there is at least one block P_i such that the number of its remainder packets is less than N_0, is $q = 1-(1-u)^l \leq lu$ where $u = \sum\limits_{k=n-N_0+1}^{n} C_n^k w^k (1-w)^{n-k}$.*

Proof. Note that the packet loss probability is of binomial distribution, then it follows easily. □

In our scheme LCO-MSAS, q is just the failing verification probability.

However, the assumption of independent packet loss does not hold in the Internet. Many studies show that packet loss is correlated, which means that the probability of loss is much higher if the previous packet is lost. To achieve a uniform loss probability in every block, instead of transmitting these packets by the sequence: $p_{11}, p_{12}, \ldots, p_{1n}, p_{21}, \ldots, p_{2n}, \ldots, p_{ln}$, we transmit these packets by the following sequence:



where $P_i = \{p_{i1}, p_{i2}, \ldots, p_{in}\}$. Then the packet loss probability of every block can be considered as uniform. Hence Lemma 1 is practical in our schemes.

Let $M = \bigcup_{i=1}^{l} P_i$ be a message chain, where $P_i = \{p_{ij} | j = 1, \ldots, n\}$. Table 1 shows the relations among the loss probability w, the length of chain l, width n and the failing verification probability q.

Table 1. Evaluated by Lemma 1

N_0	w	n	l	q
100	0.4	256	1000	4.85×10^{-9}
128	0.3	232	1000	6.86×10^{-4}
128	0.2	200	1000	4.72×10^{-5}

Recall the notations we use in LCO-MSAS. $F = \{f_{ij}\}$ is a function family consisting of single bit output MAC functions, where f_{ij} has a very fast operational rate which is similar to the hash function. Now let a denote the running time required to generate an MAC or a hash value. For example, on a computer (200MHz power PC) it is considered about $a = 1/500,000$ second. Let b and d denote the running time required for the public signature system (such as RSA) to generate a signature and to verify a signature, respectively. Consider, for example, RSA signatures with a 1024-bit modulus. Experiments indicate that a 200MHz power PC signature takes $b = 1/40$ second and the verification time is $d = 1/30,000$ second.

Now we come to analyze the overhead of LCO-MSAS. Firstly recall that every packet has to be numbered. Table 1 shows that it is adequate that $n \le 2^8$ and $l \le 2^{10}$, which means that the 18-bit space overhead is sufficient for all packets to be numbered. Then the scheme LCO-MSAS has the following overheads.

The per-packet communication overhead is $C = n + 18$ bits.

The average signature rate is $R_S = \frac{nl}{n^2(l-1)a+na+nb} = \frac{l}{n(l-1)a+a+b}$ packets/second.

The average verification rate is $R_V = \frac{l}{n(l-1)a+a+d}$ packets/second.

Since a is much less than b, by the above formulae, the smaller the n, the higher the computational rate is. On the other hand, the smaller the n, the lower the verification probability. So it is necessary to balance between the performance efficiency and the verification probability. Here, Table 2 shows several performance parameters of LCO-MSAS which is supposed running on a 200MHz power PC.

Table 2. Performance Parameters of LCO-MSAS
(where the verification probability for four cases is evaluated by Lemma 1)

Packet loss probability w	0.3	0.3	0.2	0.1
The width of chain n	160	200	128	112
The length of chain l	500	500	500	500
Secure Base Number N_0	80	100	80	80
Verification probability q	$1 - 1.97 \times 10^{-5}$	$1 - 5.45 \times 10^{-7}$	$1 - 7.2 \times 10^{-4}$	$1 - 4.1 \times 10^{-6}$
Communication overhead C	178 bits	218 bits	146 bits	130 bits
The average signature rate (packets/s) R_S	≈ 2700	≈ 2380	≈ 3270	≈ 3655
The average verification rate (packets/s) R_V	≈ 3100	≈ 2500	≈ 3910	≈ 4470

6 Balanced MSAS (B-MSAS)

6.1 Description of B-MSAS

The scheme LCO-MSAS has almost no redundancy, and its communication overhead is much lower. However, since a message chain includes too many packets, and the receiver-sides have to buffer these packets for signature verification, LCO-MSAS does not work so well in practical applications. The scheme LCO-MSAS can be modified in a way to reduce the number of messages in a chain. That is, instead of using single-bit output MAC functions in LCO-MSAS, the improved scheme is based on using the functions with the output of a fitting number of bits, and so increases a little communication overhead. Then the scheme achieves a balance between the communication overhead and the practicability. So we call it *Balanced MSAS* (shortly denoted by B-MSAS).

The signature generation and verification in B-MSAS are almost the same as LCO-MSAS, except for using the multiple-bit output MAC functions instead of the single-bit output MAC functions, and here we do not describe them repeatedly. In B-MSAS, suppose $F = \{f_{ij}\}$ is a function family consisting of k-bit output MAC functions. Let $M = \bigcup_{i=1}^{l} P_i$ (where $|P_i| = n$) be a message chain and N_0 the secure base number, then the message chain can be verified if the number of remainder packets for every block P_i is not less than N_0/k. The analyses for the performance overhead and security of B-MSAS are similar to the corresponding description for LCO-MSAS in Section 4, and here we do not repeat these statements any more. The natural number k depends on the concrete application environments such as the packet loss probability, the packet delay degree on the network and so on. Here, in Table 3 we present the performance parameters of B-MSAS, in which each MAC function has a 10-bit output or $k = 10$.

Table 3 shows that the space overhead of 12 bits is sufficient for all packets to be numbered in B-MSAS, hence the per-packet communication overhead is $C = n + 12$ bits, and other parameters are evaluated similar to the case in LCO-MSAS.

Table 3. Performance Parameters of B-MSAS (where 10-bit output MAC functions are used)

Packet loss probability w	0.3	0.2	0.1
The width of chain n	24	20	16
The length of chain l	100	100	100
Secure Base Number N_0	80	80	80
Verification probability q	$1 - 4.39 \times 10^{-3}$	$1 - 1.52 \times 10^{-3}$	$1 - 5.92 \times 10^{-4}$
Communication overhead C	252 bits	212 bits	172 bits
The average signature rate (packets/s) R_S	≈ 3360	≈ 3450	≈ 3550
The average verification rate (packets/s) R_V	≈ 20800	≈ 25000	≈ 31200

From the comparison between Table 2 and Table 3, it shows that, except for the verification probability and the communication overhead, the other performance parameters of B-MSAS are much superior to those of LCO-MSAS. Furthermore B-MSAS has much less cost of delayed verification.

6.2 Advantages of Our Schemes and Comparison with the Previous Work

Our schemes LCO-BSAS and B-MSAS are suitable for any size of receivers since the verification key is public. Furthermore they are independent of the group key management and the reliable transmittal, and hence can be widely used for various applications such as the authentication of video and audio data. The scheme B-MSAS proposed in this paper is more efficient, which is shown in Table 4.

Table 4

	The scheme in [5]	The scheme in [6]	EMSS in [4][2]	B-MSAS
Communication overhead	760 bits	2240 bits	480 bits	252 bits
Signature rate	660	410	400	3360
Verification rate	6600	410	69767	20800
Number of receivers	Small	Any size	Any size	Any size
Secure model	Semi-perfect security	Perfect security	Perfect security	Perfect security
Non-repudiability	N	Y	Y	Y
Delayed verification	N	N	Y	Y

Note: The performances of four schemes are evaluated according to the performance parameters on a 200MHz power PC described above. The signature and verification rates are in packets per second. Here all schemes are supposed to provide a security of 80 bits. The performances of both EMSS and B-MSAS are supposed to be based on a network with the packet loss probability of 0.3, and they provide a similar verification probability (99.27% and 99.56%, respectively). However, the time of the delayed verification of EMSS is slightly less than that of B-MSAS.

The scheme B-MSAS proposed in this paper is somewhat similar to the extended scheme of EMSS in [4], in which the hash of each packet splits into n chunks, and then is reconstructed by using Rabin's Information Dispersal Algorithm (IDA)[8]. But the computational overhead of IDA is higher. Besides, its performance overhead and security are not analyzed in [4].

[2] Here the performance of EMSS is evaluated by the algorithm described in [4].

7 Conclusion

In this paper, we first present problems of the Multicast Source Authentication Scheme (MSAS). To solve the multicast source authentication problem, we introduce a novel algorithm called MACM, and present two efficient schemes, LCO-MSAS and B-MSAS. LCO-MSAS offers much lower communication overhead. And B-MSAS provides minimal communication and computational overhead, strong loss robustness, and non-repudiation. They can be used for the data source authentication in the case where the data transmitted has a high packet loss probability, and they can be easily scaled up to potentially millions of receivers.

References

[1] Bruce Schneier. Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C. John Wiley & Sons, Inc., 1996.

[2] Gennaro R, Rohatgi P. How to sign digital streams. In *Advances in Cryptology — CRYPTO'99, Lecture Notes in Computer Science 1294*, Springer-Verlag, 1997, pp.180–197.

[3] Wong C K, Lam S. Digital signatures for flows and multicasts. Technical Report, TR-98-15, Department of Computer Science, The University of Texas at Austin, 1998.

[4] Perrig A, Canetti R, Tygar D, Song D. Efficient authentication and signing of multicast streams over lossy channels. In *2000 IEEE Symposium on Security and Privacy* (*S&P* 2000), Berkeley, California, 2000, pp.14–17.

[5] Caneti R, Garay J, Itkis G *et al.* Multicast security: A taxonomy and some efficient constructions. In *Proc. the IEEE INFOCOM'99*, New York: IEEE Communication Society, 1999, pp.708–716.

[6] Pankaj Rohatgi. A compact and fast hybrid signature scheme for multicast packet authentication. In *Proc. the 6th ACM Computer and Communications Security Conference*, ACM Press, 1999, pp.93–100. http://www.acm.org/pubs/articles/proceedings.

[7] Moni Naor, Omer Reingold. From unpredictability to indistinguishability: A simple construction of pseudo-random functions from MACs. In *Advances in Cryptology — Crypto'98*, Krawczyk H (ed.), Berlin: Springer-Verlag, 1998, pp.267–282.

[8] Rabin M. Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of the Association for Computing Machinery*, 1989, 36(2): 335–348.

LI Xianxian (male) was born in 1969.9. He is a Ph.D. student in School of Computer, Beijing University of Aeronautics and Astronautics. He received the MS degree in Mathematics from Guangxi Normal University in 1997. His research interests include information security and computer science.

HUAI Jinpeng (male) was born in 1962.9. He is a professor and doctoral supervisor of School of Computer, Beijing University of Aeronautics and Astronautics. He received the Ph.D. degree in computer science from Beijing University of Aeronautics and Astronautics. His research interests include computer software technology, network security and computer supported cooperative work.

Appendix

A.1 The Proof of Theorem 1

Proof. By the properties of F, to find X and Y such that $F(X,Y) = C$, any efficient adversary has to choose $X^i = (x^i_1, x^i_2, \ldots, x^i_n)$ and $Y^i = (y^i_1, y^i_2, \ldots, y^i_m)$ in M and compute $F(X^i, Y^i)$ $(i = 1, 2, \cdots)$ until he succeeds, and hence he has to operate F many times. Without loss of generality, we assume that $n \le m$, then it is sufficient to prove that the probability that an adversary is capable of finding X and Y such that $F(X,Y) = C$ when it has operated F for l times, is less than lw^n, where $w = 1/2^k + \varepsilon$.

Let A_s denote the following event: The adversary succeeded at the s-th operation, or he has found $X^s = (x^s_1, x^s_2, \ldots, x^s_n)$ and $Y^s = (y^s_1, y^s_2, \ldots, y^s_m)$ such that $F(X^s, Y^s) = C$. Then the probability that the adversary succeeded in finding X and Y such that $F(X,Y) = C$ within l operations is

$$q = \sum_{s=2}^{l} Pr(A_s | \neg A_{s-1}, \ldots, \neg A_1) + Pr(A_1),$$

where $Pr(A)$ denotes the probability of the event A's occurrence. Clearly $Pr(A_1) = w^{mn}$. Therefore it is sufficient to prove that $Pr(A_s | \neg A_{s-1}, \ldots, \neg A_1) \le w^n$ for any s.

Note that the event A_s is not independent of the former events, hence we have to consider the relations between A_s and $\neg A_{s-1}, \ldots, \neg A_1$. By the properties of f_{ij}, if $f_{ij}(x^s_i, y^s_j)$ has not been computed in former operations, that is, $(x_i^s, y_j^s) \neq (x_i^t, y_j^t)$ for $t < s$, then the probability that $f_{ij}(x_i^s, y_j^s) = c_{ij}$ holds is w, which is denoted by $Pr(f_{ij}(x_i^s, y_j^s) = c_{ij}) = w$. Now we evaluate the value of $Pr(A_s|\neg A_{s-1}, \ldots, \neg A_1)$ from the following two cases.

(1) Firstly, if there is an x_i^s (or y_j^s) in $x_1^s, \ldots, x_n^s, y_1^s, \ldots, y_m^s$ such that $x_i^s \neq x_i^t$ (or $y_j^s \neq y_j^t$) for any $t < s$, then $f_{i1}(x_i^s, y_1^s), \ldots,$ and $f_{im}(x_i^s, y_m^s)$ have not been calculated in the former operations, hence $Pr(f_{i1}(x_i^s, y_1^s) = c_{i1}) = w, \ldots, Pr(f_{im}(x_i^s, y_m^s) = c_{im}) = w$. Then we obtain that

$$Pr(A_s|\neg A_{s-1}, \ldots, \neg A_1) \leq Pr(f_{i1}(x_i^s, y_1^s) = c_{i1}) \ldots Pr(f_{im}(x_i^s, y_m^s) = c_{im}) = w^m \leq w^n.$$

The case for y_j^s is similar.

(2) Otherwise, let $x_1^s = x_1^{t_1}, \ldots, x_n^s = x_n^{t_n}$, $y_1^s = y_1^{k_1}, \ldots$ and $y_m^s = y_m^{k_m}$, where t_i, $k_j < s$. We claim that these numbers $t_1, \ldots, t_n$, $k_1, \ldots, k_m$ cannot be all the same. If so, assume that they are equal to t, then $X^s = X^t$ and $Y^s = Y^t$, thus $A_s = A_t$, which contradicts $\neg A_t$.

a) If $k_1 = k_2 = \cdots = k_m$, then there is at least a t_i such that $t_i \neq k_1, \ldots, t_i \neq k_m$, hence $(x_i^s, y_1^s) = (x_i^{t_i}, y_1^{k_1}), \ldots$ and $(x_i^s, y_m^s) = (x_i^{t_i}, y_m^{k_m})$ were not computed in the previous operations. Then it follows from the properties of $\{f_{ij}\}$ that

$$Pr(A_s|\neg A_{s-1}, \ldots, \neg A_1) \leq Pr(f_{i1}(x_i^s, y_1^s) = c_{i1}) \ldots Pr(f_{im}(x_i^s, y_m^s) = c_{im}) = w^m \leq w^n.$$

b) If $k_1, k_2, \ldots$ and k_m are not all equal, then, for each t_i, there is a $k_{r_i} \in \{k_1, \ldots, k_m\}$ such that $t_i \neq k_{r_i}$, where $r_i \in \{1, \ldots, m\}$ and $i = 1, \ldots, n$, therefore $(x_1^s, y_{r_1}^s) = (x_1^{t_1}, y_{r_1}^{k_{r_1}}), \ldots, (x_n^s, y_{r_n}^s) = (x_n^{t_n}, y_{r_n}^{k_{r_n}})$ were not computed in the previous trials, and hence $Pr(f_{1r_1}(x_1^s, y_{r_1}^s) = c_{1r_1}) = w, \ldots$ and $Pr(f_{nr_n}(x_n^s, y_{r_n}^s) = c_{nr_n}) = w$. So we obtain that

$$Pr(A_s|\neg A_{s-1}, \ldots, A_1) \leq Pr(f_{1r_1}(x_1^s, y_{r_1}^s) = c_{1r_1}) \cdots Pr(f_{nr_n}(x_n^s, y_{r_n}^s) = c_{nr_n}) = w^n.$$

From the statements above, whatever is the case, we obtain that $Pr(A_s|\neg A_{s-1}, \ldots, \neg A_1) \leq w^n$, then we have proved the theorem. □

A.2 The Proof of Theorem 2

Notation. Let $A = (a_{ij})_{n \times m}$ be an $n \times m$ matrix, the set $I = \{i_1, i_2, \ldots, i_s\} \subseteq \{1, 2, \ldots, n\}$ and $J = \{j_1, j_2, \ldots, j_t\} \subseteq \{1, 2, \ldots, m\}$. We use $A_{I,J}$ to denote the $s \times t$ matrix whose element in the r-th row and v-th column is a_{i_r, j_v}, or $A_{I,J} = (a_{ij})_{i \in I, j \in J}$.

Proof. In the LOC-MSAS, the signing key is SK_A, hence it is sufficient to prove that any adversary who does not know SK_A has a negligible probability to generate a legitimate signature for any $m \notin M$, where $M = \{p_{ij} | j = 1, \ldots, n,\ i = 1, \ldots, l\}$ is described in LOC-MSAS.

Here, we use the notation in Subsection 5.1, and let $J_i = \{j_1^{(i)}, j_2^{(i)}, \ldots, j_{N_i}^{(i)}\}$ for $i = 1, \ldots, l$, where $|J_i| = N_i \geq N_0$. Denote $\alpha_i = (p_{i,j_1^{(i)}}, p_{i,j_2^{(i)}}, \ldots, p_{i,j_{N_i}^{(i)}})$ and $\kappa_i = (k_{i,j_1^{(i)}}, k_{i,j_2^{(i)}}, \ldots, k_{i,j_{N_i}^{(i)}})$, where each $k_{i,j_r^{(i)}}$ is the authentication information of $p_{i,j_r^{(i)}}$ attached within the packet. Let

$$\beta_i = \begin{pmatrix} (k_{i,j_1^{(i)}})_{j_1^{(i+1)}} & (k_{i,j_2^{(i)}})_{j_1^{(i+1)}} & \cdots & (k_{i,j_{N_i}^{(i)}})_{j_1^{(i+1)}} \\ (k_{i,j_1^{(i)}})_{j_2^{(i+1)}} & (k_{i,j_2^{(i)}})_{j_2^{(i+1)}} & \cdots & (k_{i,j_{N_i}^{(i)}})_{j_2^{(i+1)}} \\ \vdots & \vdots & \vdots & \vdots \\ (k_{i,j_1^{(i)}})_{j_{N_{i+1}}^{(i+1)}} & (k_{i,j_2^{(i)}})_{j_1^{(i+1)}} & \cdots & (k_{i,j_{N_i}^{(i)}})_{j_{N_{i+1}}^{(i+1)}} \end{pmatrix}$$

be an $N_{i+1} \times N_i$ matrix, where $(k_{i,j_r^{(i)}})_{j_r^{(i+1)}}$ denotes the $j_r^{(i+1)}$-th bit of $k_{i,j_r^{(i)}}$. Then the legitimacy of the signature verification described in Subsection 5.1 implies that

$$\beta_1 = F_{J_2,J_1}(\kappa_2, \alpha_1),\ \beta_2 = F_{J_3,J_2}(\kappa_3, \alpha_2), \ldots,\ \beta_{l-1} = F_{J_l,J_{l-1}}(\kappa_l, \alpha_{l-1}),$$

where β_i is an MACM for (κ_{i+1}, α_i). By noting that β_i is a sub-matrix of κ_i, it is fixed by κ_i. By the security of the public key signature, $Ver_{PK_A}(k_{1,j}, s_{1j}) = T$ implies that the probability for $k_{1,j_r^{(1)}}$ to be forged is negligible, and hence κ_1 is forged with a negligible probability, and so is β_1. Since $|J_2|$, $|J_1| \geq N_0$, F_{J_2,J_1} is target collision resistance by Corollary 1, therefore the probability that α_1, κ_2 are forged is negligible. But β_2 is fixed by κ_2, then the process is continued. Consequently, the probability that $\alpha_2, \ldots, \alpha_{l-1}$ and κ_l are forged is negligible. Since $\kappa_l = (H(p_{l,j_1^{(l)}}), \ldots, H(p_{l,j_{N_l}^{(l)}}))$, the collision resistibility of the Hash Function H ensures that each $p_{l,j_r^{(l)}}$ has a negligible probability of being forged. Then we have shown that each packet being verified successfully is forged only with a negligible probability.

So we have proved that LCO-MSAS is secure by the above statements. □