International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:4, No:2, 2010

Signing the First Packet in Amortization Scheme for Multicast Stream Authentication

Mohammed Shatnawi, Qusai Abuein, and Susumu Shibusawa

Abstract—Signature amortization schemes have been introduced for authenticating multicast streams, in which, a single signature is amortized over several packets. The hash value of each packet is computed, some hash values are appended to other packets, forming what is known as hash chain. These schemes divide the stream into blocks, each block is a number of packets, the signature packet in these schemes is either the first or the last packet of the block. Amortization schemes are efficient solutions in terms of computation and communication overhead, specially in real-time environment. The main effictive factor of amortization schemes is it's hash chain construction. Some studies show that signing the first packet of each block reduces the receiver's delay and prevents DoS attacks, other studies show that signing the last packet reduces the sender's delay. To our knowledge, there is no studies that show which is better, to sign the first or the last packet in terms of authentication probability and resistance to packet loss.

In th is paper we will introduce another scheme for authenticating multicast streams that is robust against packet loss, reduces the overhead, and prevents the DoS attacks experienced by the receiver in the same time. Our scheme-The Multiple Connected Chain signing the First packet (MCF) is to append the hash values of specific packets to other packets,then append some hashes to the signature packet which is sent as the first packet in the block. This scheme is aspecially efficient in terms of receiver's delay. We discuss and evaluate the performance of our proposed scheme against those that sign the last packet of the block.

Keywords—multicast stream authentication, hash chain construction, signature amortization, authentication probability.

I. Introduction

THE authentication of multicast streams using signature amortization requires a single signature for each group of packets, which is known as a block. The signature packet is usually the last or the first packet of each block. Signing the last packet increases the receiver's delay and buffer capacity. More over the receivers may mount denail of service (DoS) attacks; that is, if any receiver does not receive the signature as the first packet of the block, then he is forced to keep in buffer some unsecured packets. The reason is that the receiver is unable to verify the packets authenticity immediately, that is, whether these packets are valid or not [1], [2].

In previous works [3], [4], [5], we introduced a Multiple Connected Chains (MC) model for amortization schemes that signs the last packet of each block. So as to overcome DoS attacks, we introduce and discuss our scheme by signing the first packet of the block. We also discuss signing the first packet and its effect on the authentication probability, loss resistance and the buffer capacities and delays of the sender and receivers. We show how to measure the effeciency metrics of amortization schemes, such as overhead, loss resistance and the authentication probability. We compare the results of our introduced scheme with MC model, so as to show which is better to sign the first or the last packet of the block.

This paper is organized as follows: Section II introduces our authentication scheme. In Section III we analyze the efficiency of our scheme in terms of overhead and in Section IV in terms of loss resistance. The authentication probability of our scheme is derived and analyzed in Section V. In Section VI we show the required buffer and delay for both the sender and receiver. In Section VII we evaluate the performance of our scheme and in Section VIII we present previous works on stream authentication schemes. In Section IX we give the conclusion of our study.

M. Shatnawi is with the Departmentt of Computer Information Systems, Jordan University of Science and Technology, mshatnawi@just.edu.jo
Q. Abuein is with the Department. of Computer Information Systems, Jordan University of Science and Technology, qabuein@just.edu.jo
S. Shibusawa is with the Department. of Computer and Information Sciences, Ibaraki University, Japan, sibusawa@mx.ibaraki.ac.jp

II. Our Authentication Scheme

In this section we descripe our authentication scheme that signs the first packet of each block so as to be robust against DoS attacks. We call the model of our scheme a Multiple Connected Chains signing the First packet (MCF). Table I shows the notation used in this paper. A sender intends to send a stream of N messages to receivers. Each message M_i is sent along with additional authentication information. Our MCF model is also efficient in terms of loss resistance and overhead.

The stream is divided into blocks, each block consists of some packets. A sender appends the hash $H(P_i)$ of a packet P_i to specific other packets to achieve robustness against packet loss. For each block the sender then concatenate hashes of specific packets together and signs them. The signed packet is called a signature packet P_{sig}. The sender sends a signature packet at the begining of each block and sent as the first packet to receivers so as to enable them from verifying the received packets directly, which protects receivers from DoS attacks.

A packet P_i contains the hash values $H(P_i)$ of ν other packets as P_{i+1} and P_{i+jc}, where $j = 1, 2, \cdots, \nu - 1$. For example, when $\nu = 3$, P_i contains $H(P_{i+1})$, $H(P_{i+c})$ and $H(P_{i+2c})$. Let $A(c, \nu)$ denote a set of the packets that have their hashes appended to P_i, then

$$A(c, \nu) = \{P_{i+1}, P_{i+c}, P_{i+2c}, \cdots, P_{i+(\nu-1)c}\}. \quad (1)$$

Fig. 1 shows the appended hashes to each packet according to MCF, when $\nu = 3$. So as to construct MCF model and be robust against packet loss, we need the value of ν as $\nu \geq 2$. For each block μ hashes are concatenated together and signed using the sender's digital key. Let P_{j_1} be the first packet that

TABLE I
NOTATION

symbol	representation
N	the number of messages in the stream
c	the number of chains in MCF model
k	the number of slices in a block
ν	the number of packets that have hashes appended to P_i
μ	the number of hashes appended to the signature
j_i	The number of the packet that has its hash appended to a signature packet, where $1 \leq i \leq \mu$
s	the signature size (RSA is 128 bytes)
h	the hash size (SHA-512 is 64 bytes)
δ	the communication overhead per packet in bytes
γ	the number of signature packets
ℓ	the loss resistance

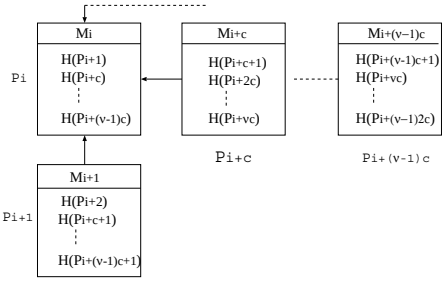

Fig. 1. Appending hashes to other packets in MCF model.

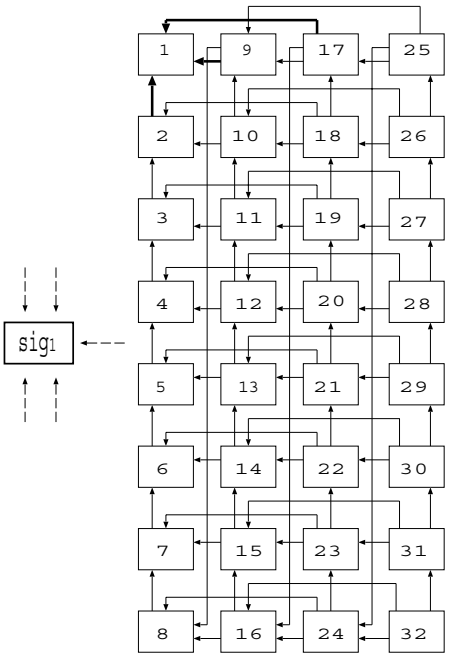

Fig. 2. A construction of a single block of MCF model for $c = 8$, $k = 4$ and $\nu = 3$.

has its hash appended to P_{sig} and P_{j_μ} be the last one. Then the set of the packets that have their μ hashes appended to P_{sig} is:

$$E(\mu) = \{P_{j_1}, P_{j_2}, \cdots, P_{j_\mu}\}, \tag{2}$$

where $j_1 < j_2 \cdots < j_\mu$.

MCF model consists of c chains, where each chain consists of some packets. The block size of MCF model is ck packets, where k represents the number of slices. The group of the first c packets $\{P_1, P_2, \cdots, P_c\}$ is the first slice in MCF model, the group of the second c packets $\{P_{c+1}, P_{c+2}, \cdots, P_{2c}\}$ is the second slice, and so on. Fig. 2 depicts a construction of a single block of MCF model for $c = 8$, $k = 4$ and $\nu = 3$. Note that according to MCF model, where the signature packet is the first packet of a block, it is impossible to connect the blocks of the stream to each other. To do so, the sender needs to buffer the whole stream. For MCF model we assume that P_{sig} is always received. The sender buffers the ck packets, computes $H(P_{ck})$ as $H(M_{ck})$ and buffers $H(P_{ck})$. The sender constructs P_i by concatenating the hash $H(P_{i+1})$ with every message M_i, then computes $H(P_i)$ and keep it in the buffer, where $(k-1)c < i < ck$. While he constructs P_i by concatenating the hashes $H(P_{i+1})$ and $H(P_{i+c})$ with every message M_i, the sender computes $H(P_i)$ and buffers it, where $(k-2)c < i < (k-1)c$. Every packet P_i is constructed by concatenating the hashes $H(P_{i+1})$, $H(P_{i+c})$ and $H(P_{i+2c})$ with every message M_i, then computes $H(P_i)$ and buffers it, where $1 \leq i \leq (k-2)c$.

The sender then concatenates the μ hashes $H(P_{j_1})$, $H(P_{j_2})$, $\cdots$, $H(P_{j_\mu})$ together and signs them to construct the signature packet P_{sig_1}, then sends P_{sig_1}. Then starts sending the packets of the block. The receivers upon receiving the packets, can verify and use each packet directly without delay.

The authentication steps that the sender performs are the same for each block, so we descripe these steps for a single block as follows:

1) Choose value of ν
2) Determine the number of chains c
3) Choose values of μ and k
4) Buffer the packets of the block
5) Append necessary hash values to P_i, compute $H(P_i)$ and bufer it, where $i = ck, ck-1, \cdots, 1$
6) Choose $E(\mu)$
7) Append μ hashes to P_{sig}, sign and send P_{sig}.
8) Send P_i, $1 \leq i \leq ck$

While the verification steps the receivers perform for a single block are as follows:

1) Receive P_{sig}.
2) Resort the order of the received packets P_i,$1 \leq i \leq ck$.
3) Retrieve $H(P_{j_1})$,$H(P_{j2})$, $\cdots$, $H(P_{j\mu})$.
4) Compute $H(P_i)$ and retrieve $H(P_{i+1})$,$H(P_{i+c})$, $\cdots$,$H(P_{i+(\nu-1)c})$.
5) Compare the computed hash values to the retrieved ones.
6) After verifying P_{ck} , the receiver starts using the received packets.

III. OVERHEAD

The computation overhead is the number of additional information such as hashes and digital signatures that the sender computes so as to authenticate the packets. According

to our scheme the sender computes N hash values for a stream of N messages and a single signature packet for each block.

While the communication overhead means the total size of added information to the packets to authenticate it. The overhead is an important metric to measure the efficiency of the authentication schemes. In this section we show how to measure the communication overhead per packet according to our scheme, the parameters that affect the overhead and how to choose the values of these parameters.

Since a packet P_i in MCF model contains hashes of succeeding packets, P_{ck} contains no additional hashes. While each of the remaining packets of the kth slice $\{P_{ck-1}, P_{ck-2}, \cdots, P_{(k-1)c+1}\}$ contains only a single hash, that is, there are $c-1$ hashes in the kth slice. Each packet of the$(k-1)th$ slice $\{P_{(k-2)c+1}, P_{(k-2)}c+2, \cdots, P_{(k-1)c}\}$ contains 2 hashes, so there are $2c$ hashes in the $(k-1)th$ slice. The number of hashes in $\{P_{(k-\nu)c+1}, P_{(k-\nu)c+2}, \cdots, P_{ck}\}$ is $c-1+2c+3c+\cdots+\nu c$; that is , $(\frac{\nu^2+\nu}{2})c-1$ each packet of the remaining packets $\{P_{(k-\nu)c}, P_{(k-\nu)c-1}, \cdots, P_1\}$ contains ν hashes, so there are $\nu(ck-\nu c)$ hashes. Accordingly, the total number of hashes β that are appended to the packets of a block of size ck packets is computed as:

$$\beta = (\frac{\nu - \nu^2}{2} + \nu k)c - 1. \quad (3)$$

Definition 1: The communication overhead per packet δ in bytes is the total size of the hashes that are appended to the whole packets of a block and the size of the signature packets divided by ck.

$$\delta = \frac{h\beta + s}{ck}. \quad (4)$$

Multiplying the hash value h by β gives the total size of all hashes that are appended to the whole packets of a block, while there is a signature packet of size s in each block. Solving Equation (4) accordingly, gives the following:

$$\delta = \frac{h}{k}(\frac{\nu - \nu^2}{2}) + h\nu - (\frac{h-s}{ck}). \quad (5)$$

The overhead per packet δ decreases as the block size ck increases as Equation (5) shows. This can be achieved by increasing the number of chains c, the number of slices k or both. Fig. 3 depicts δ in terms of c for a block size of 80 packets when $c = 16, \nu = 2, s = 128$ bytes and $h = 64$ bytes.

We showed how to measure the overhead per packet according to MCF model, now we show how to determine the values of the parameters ν, μ and the set $E(\mu)$. There are two kinds of packet loss the scheme need to resist, random and burst packet losses. The values of ν and μ must be chosen so as to resist both losses. According to the expected loss ratio τ, the sender can choose the value of ν so as to guarantee the receive of at least one packet of $A(c, \nu)$ with the desired probability, which is equal to $1 - \tau^{\nu}$. So as to resist longer burst loss we increase the value of c instead of increasing ν and μ so as to reduce the overhead as will be shown in Section IV.

The appropriate value of μ can be chosen in the same manner ν have been chosen. While we choose the packets of

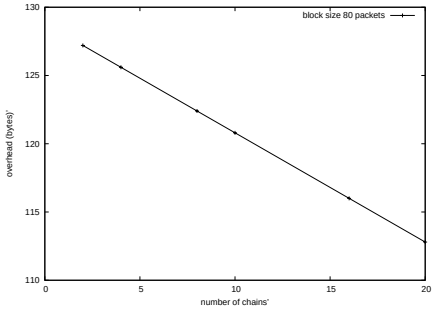

Fig. 3. Overhead per packet in terms of number of chains c for a block when $\nu = 2$, $s = 128$ and $h = 64$.

$E(\mu)$ such that the distance in number of packets between P_{j_1} and P_{j_μ} is greater than the length of the expected burst b so as to guarantee that at least one packet is received. Accordingly, choosing $j_\mu - j_1 \geq b$ guarantees achieving that goal wherever the burst occurs. The reason to choose the packets of $E(\mu)$ in terms of b is that Internet packet loss is burst in nature, and if a packet P_i is lost, packet P_{i+1} is likely to be lost [6], [7], [8].

IV. Loss Resistance and Number of Chains

Loss resistance ℓ is the maximum number of lost packets the scheme can sustain and still able to authenticate the received packets. Loss resistance is another important metric to measure the efficiency of the authentication scheme. The stronger resistance against packet loss is achieved, the more efficient the scheme is. In this section we show how to measure the loss resistance ℓ that our scheme can achieve and how to choose the appropriate value of the parameter c that has the great influence of our scheme.

Packet $P_{i+(\nu-1)c}$ is the farthest packet that has its hash $H(P_i)$ appended to a packet P_i according to MCF model. So loss resistance is equal to the number of packets between P_i and $P_{i+(\nu-1)c}$, accordingly:

$$\ell = (\nu - 1)c - 1 \ . \quad (6)$$

Where $i \geq (\nu - 1)c$. Equation (6) shows that stronger loss resistance ℓ is achieved by increasing c, which reduces the overhead δ in the same time.

The number of chains c, plays the main role in the efficiency of our model in terms of overhead and loss resistance. We choose the appropriate value of c in terms of the length of the expected burst loss b. The scheme must resist the expected b; otherwise, the authentication of the received packets that lies before the start of the burst becomes impossible. Accordingly, $(\nu - 1)c - 1 \geq b$, that is,

$$c \geq \left\lceil \frac{b+1}{\nu - 1} \right\rceil . \quad (7)$$

V. Authentication Probability

The authentication probability is an important metric to measure the efficiency of the authentication scheme. In this

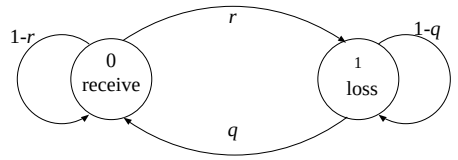

Fig. 4. 2-state Markov model for burst packet loss.

section we derive the authentication probability of our scheme using 2-state Markov model and analyze the authentication probability in terms of several parameters.

According to MCF model, packet P_i is authenticated if at least one packet of $E(\mu)$ and at least one packet of $A(c,\nu)$ are received, in addition to signature packet P_{sig}. Note that for P_i to be authenticated, all the whole packets of $E(\mu)$, $A(c,\nu)$ or both cannot be lost.

For the purpose of deriving the authentication probability of P_i, we assume the followings:

- the derivation applies to a single block.
- packets P_i and P_{sig_1} are received.
- $i+(\nu-1)c \leq j_1$. This means that the farthest packet that contains the hash of P_i lies before the first packet of those that have hashes appended to P_{sig}.

Let $P_r\{P_i\}$ denote the authentication probability of packet P_i when P_i is received, then $P_r\{P_i\}$ is expressed as:

$$P_r\{P_i\} = P_r\{P_i \; is \; verifiable \mid P_i \; is \; received \; \}. \quad (8)$$

The burst packet loss is well characterized using 2-state Markov model [6], [7]. Fig. 4 shows the 2-state Markov model where r represents the probability that the next packet is lost, provided the previous one has arrived. q is the transition probability from loss state to received state, and it is opposite to r.

A. Authentication Probability Using 2-State Markov Model

According to 2-state Markov model depicted in Fig. 4, receive and loss states are denoted 0 and 1, respectively.

Theorem 1: Based on 2-state Markov model the authentication probability of the ith packet P_i in a block of MCF is given as follows, when $i+(\nu-1)c \leq j_1$:

$$P_r\{P_i\} = \sum_{g,h} \left\{ \left[p_{0g_1} p_{g_1 g_2}^{(c-1)} \prod_{l=2}^{\nu-1} (p_{g_l g_{l+1}}^{(c)}) \right] \left[p_{g_\nu h_1}^{(j_1-i-(\nu-1)c)} \prod_{l=1}^{\mu-1} (p_{h_l h_{l+1}}^{(j_{l+1}-j_l)}) \right] \right\} \quad (9)$$

where $g_l \in \{0,1\}$, $l = 1,2,\cdots,\nu$, $g = (g_1, g_2, \cdots, g_\nu) \neq (1,1,\cdots,1)$. Also $h_l \in \{0,1\}$, $l = 1,2,\cdots,\mu$, $h = (h_1, h_2, \cdots, h_\mu) \neq (1,1,\cdots,1)$.

Proof: Since P_i is received, there is a single transition state from P_i to P_{i+1}, so the transition probability is denoted p_{0g_1}. There are $(c-1)$ transition states from P_{i+1} to P_{i+c}, so the transition probability is denoted $p_{g_1 g_2}^{(c-1)}$. On the other hand, there are c transition states between every two adjacent packets of $A(c,\nu) - \{P_{i+1}\}$, so we have transition probability $\prod_{l=2}^{\nu-1}(p_{g_l g_{l+1}}^{(c)})$, and in total we have transition probability $p_{0g_1} p_{g_1 g_2}^{(c-1)} \prod_{l=2}^{\nu-1}(p_{g_l g_{l+1}}^{(c)})$. Also a signature packet is assumed to be received and μ hashes of previous packets are appended to it. Since $i+(\nu-1)c \leq j_1$, we have $(j_1-i-(\nu-1)c)$ transition states from $P_{i+(\nu-1)c}$ to P_{j_1}, and the transition probability is denoted $p_{g_\nu h_1}^{(j_1-i-(\nu-1)c)}$. There are (j_2-j_1) transition states from P_{j_1} to $P_{j_2}, \cdots, (j_\mu - j_{\mu-1})$ transition states from $P_{j_{\mu-1}}$ to P_{j_μ}, so we have transition probability $\prod_{l=1}^{\mu-1}(p_{h_l h_{l+1}}^{(j_{l+1}-j_l)})$. The total of the whole transition probabilities gives the desired result. □

B. Analysis of the Authentication Probability

The analysis of the MCF model and the effect of the parameters ν, μ, c and the two probabilities r and q of Markov model are the same as those of MC model introduced in [5] when applied to a single block. When the analysis is applied to the whole stream MCF model achieves lower authentication probability, since the blocks of MCF model are not connceted to each other as mentiond in Section II.

VI. Buffer Capacity and Delay

The sender and receivers delays in number of packets as well as the buffers capacities are important metrics to measure the efficiency of the authentication scheme specially in real time streaming, where the receivers usually do not buffer large amounts of unconsumed data. In this section we show the effect of MCF model on the delays and buffers capacities for both the sender and the receivers.

A. Sender's Buffer and Delay

Since the first packet of each block is signed, the sender experiences a ck packet delay. The sender needs to buffer $(\nu-1)c+1$ hashes so as to compute the hash value of any packet P_i in addition to μ hashes neccessary to compute the signature packet P_{sig}. While he needs to buffer the whole ck packets of each block before sending P_{sig}. Accordingly, the sender's buffer size α is given as:

$$\alpha = (\nu-1)c + \mu + ck + 1 \quad (10)$$

This equation shows that the sender's buffer capacity increases as c increases for a single block.

B. Receiver's Buffer and Delay

Since the packets reach the receiver unordered, he needs to resort them before starting the verification. The receiver can verify the received packets with less delay, that is, a single packet delay, since the signature packet is signed at the begining of each block and packet p_i contains hashes of succedding packets. The receiver needs to buffer $(\nu-1)c+1$ hash values that are retrieved from the packets in addition to the μ hash values that are retrieved from the signature packet for verification purposes. Accordingly, the receiver's buffer size α is given as:

$$\alpha = (\nu-1)c + \mu + 1. \quad (11)$$

VII. Evaluation

In this section we discuss and evaluate the performance of our scheme in terms of hash chain construction, loss resistance, authentication probability and sender's and receiver's delay against perviously introduced scheme MC model [5], which signs the last packet of the block.

A. Hash Chain Construction

The MCF model signs the first packet of the block, while the MC model signs the last one. Signing the first packet prevents us from connecting the blocks with each other, which affects the authentication probabilty and the resistance agains burst packet loss for some packets of the block as will be shown next subsection.

B. Loss Resistance

Our MCF model achieves loss resistance equal to $\ell = (\nu - 1)c - 1$ as given by equation (6) for a single bloack only, since the blocks are not connected to each other, which means that the c packets that preceed P_{sig} do not achieve same resistance to burst loss as those of the other packets of the block. The MC model achieves resistance to burst loss equals to $(\nu - 1)c - 1$ for the whole stream.

C. Authentication Probability

MCF model achieves less authentication probabilty for the whole stream comparing to MC model because hashes of some packets of a block, according to MCF model, are not appended to packets of a successor block.

D. Sender's Delay

Our previous scheme using MC model signs the last packet of a block. Therefore, the sender experiences a delay of a single packet. While in this scheme, using MCF model, where the first packet of each block is signed, the sender experiences a delay of ck packets, where c is the number of chains in MCF model and k is the number of slices in a block.

E. Receiver's Delay

Using MC model, the receiver's delay depends solely on the block size, as the receiver has to buffer all the packets preceeding the signatre packet, and wait until P_{sig} is received. In MCF model, however, the receiver experiences only a single packet delay, since the signature packet is received at the begining of each block and packet P_i contains hashes of succeeding packets. So the receiver can authenticate and use every packet at once upon receiving.

VIII. Related Works

Signing each packet of the stream separately is impractical solution in terms of computation and communication overhead and delay on both sender and receivers [9], even if we try to fasten the signing process using schems such as in [1]. To reduce the number of signatures, TESLA [8] was introduced depending on time synchronization between sender and receivers.

Another approcah used a single signature for each block of packets called signature amortization schemes were introduced, such as Authentication Tree [1], EMSS [8] and Augmented Chain [10]. The security of signature amortization schemes was introduced by Gennaro and Rohatgi in [11].

EMSS increases the weak loss resistance of the scheme introduced in [8] by increasing the number of hashes that are appended to a packet randomly chosen and apply multiple hashes to the signature one. According to [2] and [12] appneding more hashes to other packets will increase the overhead. The AC algorithm uses a deterministic way to strengthen resistance to loss as introduced in [13]

IX. Conclusion

Signing the first packet of a block reduces the receiver's delay and prevents the Denial of Service (DoS) attacks that is experienced by the reciever, but it affects negatively on the authentication probability and resistance to burst loss comparing to MC model where the signature packet is the last one of the block.

As future works, we will study the use of the Forward Error Correction (FEC) with our MC and MCF models to see the effect of it on the performance of our schemes.

References

[1] S. Miner and J. Staddon, "Graph-based authentication of digital streams," Proc. of the IEEE Symposium on Research in Security and Privacy, pp.232-246, May 2001.
[2] J. Park, E. Chong and H. Siegel, "Efficient multicast stream authentication using erasure codes," ACM Trans. on Information and System Security, vol.6, no.2, pp.258-258, May 2003.
[3] Q. Abuein and S. Shibusawa, "The performance of amortization scheme for secure multicast streaming," Proc. of the 6th Int. Workshop on Information Security Application, Jeju Island, Korea, Aug. 2005
[4] Q. Abuein and S. Shibusawa, "Signature amortization using multiple connected chains," Proc. of Springer LNCS 9th IFIP TC-6 TC-11 Int. Conf. on CMS, Sep. 2005.
[5] Q. Abuein and S. Shibusawa, A Graph-based new amortization scheme for multicast streams authentication, Journal of Advanced Modeling and Optimization, Vol. 7, No. 2, pp.238-261, 2005.
[6] H. Sanneck, G. Carle, and R. Koodli, "A framework model for packet loss metrics based on loss runlengths," SPIE/ACM SIGMM Multimedia Computing and Networking Conf., Jan. 2000.
[7] W. Jiang and H. Schulzrinne, "Modeling of packet loss and delay and their effect on real-time multimedia service quality," Proc. of 10th Int. Workshop on Network and Operations System Support for Digital Audio and Video, June 2000.
[8] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," IEEE Symposium on Security and Privacy, pp.56-73, May 2000.
[9] P. Rohatgi, "A compact and fast hybrid signature scheme for multicast packet authentication," Proc. of the 6th ACM Conf. on Computer and Communications Security, 1999.
[10] P. Golle and N. Modadugu. "Authenticating streamed data in the presence of random packet loss," Proc. of ISOC Network and Distributed System Security Symposium, pp.13-22, 2001.
[11] R. Gennaro, and P. Rohatgi, "How to sign digital streams," Advances in Cryptology - CRYPTO'97, pp.180-197, 1997.
[12] A. Chan, "A graph-theoretical analysis of multicast authentication," Proc. of the 23rd Int. Conf. on Distributed Computing Systems, 2003.
[13] P. Alain and M. Refik, "Authenticating real time packet stream and multicast," Proc. of 7th IEEE Symposium on Computers and Communications, July 2002.