Efficient Loss Resistance Multicast Stream Authentication

Qusai Abuein   Susumu Shibusawa
Department of Computer and Information Sciences
Ibaraki University
Hitachi, Ibaraki 316-8511, Japan
Tel: +81-294-38-5143   Fax: +81-294-38-5282
{abueinq, sibusawa}@cis.ibaraki.ac.jp

Abstract

Several solutions had been introduced to authenticate streamed data delivered in real-time over insecure networks, where there is no guarantee that every packet will be delivered. Some solutions resist any type of packet loss, others resist burst loss. Amortization schemes reduce the overhead caused by other schemes, but suffer from several weak points, such as where to place the signature packet, that is, after how many packets to send the signature. How many hashes to append to each packet, in addition to no clear chain structure analysis had been introduced, so as to show the effect on the efficiency in terms of the authentication probability, loss resistance and overhead. In this paper we introduce a new chain construction for multicast stream authentication delivered in real-time using signature amortization, giving solutions for the shortcomings. We also introduce a theoretical analysis of the chain construction to show its effect on the authentication efficiency. The proposed scheme consists of several odd-even chains, where the odd chains link some of the odd numbered packets, and the even chains link some of the even numbered ones. The scheme achieves better performance in terms of loss resistance and low overhead by changing the number of chains. That is when increasing the number of chains, low overhead and longer packet loss resistance are achieved. The sender's buffer capacity is taken into consideration when choosing the number of chains. We also introduce equations to quantify the requirements such as the buffer size and delay on the receiver.

1 Introduction

Multicast streaming environment is usually burst-lossy in which a consecutive packets are lost. Security in such environment is challenge, specially in terms of communication and computation overhead[1]. Streamed data is potentially very long or infinite sequence of bits that the sender sends to the receiver who must consume it at more or less the input rate, known as real-time[2].

Video conferences, TV broadcast, digitized video and audio, online gaming, stock quotes are examples of streams. Authenticating the real-time streams is more difficult due to the packet loss, high overhead and delay. We define delay here as the time in number of packets, the receiver has to wait until he is able to authenticate the received packets.

Several schemes have been introduced to solve such problems using digital signatures to provide non-repudiation[3]. Signing each packet suffer from high computation and communication overhead on both the sender and the receiver. Digital signatures are also slow in computation comparing to hashes. According to [4], a Pentium II 300 MHz machine devoting all its processor time can only generate 80 RSA signatures of size 512-bit and verify 2000 signatures per second. More studies about computation time are found in [1] and [3].

To reduce the high cost caused by sign-each schemes, amortization schemes have been introduced such as [2],[5] and [6], in which a single signature is amortized over multiple packets, using multiple hash links to achieve robustness to packet loss.

Studies such as [7] and [8] analyze the chains construction introduced by amortization schemes based on the graph theory, showing that the authentication probability and the overhead are dependent, that is, increasing one factor increases the other. Amortization schemes still suffer from shortcomings such as no clear way had been introduced to choose the block size, how many hashes to append to each packet, and how to lengthen the path between a packet and the signature one, so as to increase loss resistance.

In this paper we will introduce a new chain construction scheme to authenticate IP multicast streams using signature amortization. Our scheme solves most the shortcomings of the previously proposed amortization schemes. We also introduce a clear analysis and equations so as to determine the block size for a multicast stream authentication. Our analysis also shows the relation between the overhead, signature position, number of signatures in the stream and the authentication probability, enabling the sender to choose the factors that achieve the expected performance according to the available resources. Our scheme achieves more burst loss resistance without increasing the overhead.

In the next Section we discuss related work, and the chain construction is given in Section 3. In Section 4 we derive the authentication probability and loss resistance. In Section 5 we show the required buffer and delay on both the sender and receiver. Conclusions are given in Section 7 after evaluating our model performance in Section 6.

2 Related Work

Amortization schemes are initially studied by the authors of [2], where they introduced a very simple chain which does not tolerate loss. Even missing a single packet leaves the rest of whole block unauthenticable. Their major contribution is that they proved the security of the hash chaining technique.

EMSS was introduced by the authors of [5], to overcome the weak loss resistance [2], by storing the hash of each packet in multiple locations and append multiple hashes to the signature packet. This method according to [7] and [8], is in turn increases the overhead. EMSS determines the block size and the number of hashes to append in each packet by experiments. It also chooses the location of these hashes randomly.

Augmented chain (AC) introduced by the authors of [6], to achieve longer burst loss resistance using similar strategy to EMSS, but the locations of the appended hashes are deterministic. AC does not detail how to choose the number of packets to be inserted between each pair of the original chain. More details about the analysis of AC is found in [9], where it is applied to two case studies and compared to EMSS.

The authors of [7] and [8] give an analysis of the hash chain based on graph theory. They show that to increase the authentication probability, the number of paths from any packet to the signature one should be increased, but that will increase the overhead.

We will introduce a new chain construction that achieves longer loss resistance without increasing the overhead. Also we will connect the packets preceding the signature to that after it, so as to increase the authentication probability, that is, the packets are not dependent on a single signature.

3 Chain Construction

The hash value of each packet is computed using any hash algorithm such as SHA-1. Each packet is linked to three other packets, that is, its hash value is appended to three other packets, so as to increase robustness to packet loss. After a specific number p of packets we place a signature packet, using RSA for example, that contains several hashes of previous packets. Non-repudiation is achieved by using the digital signature. The authentication of the received packets is possible upon authenticating those packets whose hashes are appended to the signature one. The security of such schemes are proved by Rohatgi [2].

The packets preceding each signature are linked with that succeeding it so as to increase loss resistance and authentication probability, since if a signature packet is lost, the received packets can be authenticated upon receiving the next signature.

When stream S consists of N contiguous packets P_i, where $1 \leq i \leq N$,

$$S = \{P_1, P_2, \ldots, P_N\}$$

3.1 Basic Construction

Since the stream is sent to the receiver as sequenced packets, that is, each packet has a unique sequence number, we introduce two types of chains, odd and even chains. Odd chain links some of the odd packets together and the even chain links some of the even packets together. When the stream consists of c chains, each packet P_i is connected to two other packets as follows: P_{i+c}, P_{i+2c}. Figure 1 depicts the case when the number of chains c equals 2, that is, we have a single odd chain and a single even chain.

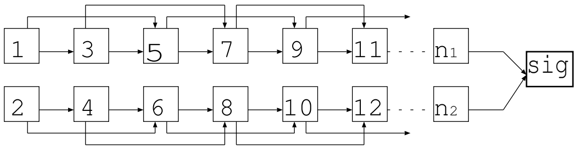

Figure 1: Single odd and single even chain, c=2.

The odd chain links the odd numbered packets $\{1, 3, 5, \ldots\}$, while the even chain links the even ones $\{2, 4, 6, \ldots\}$. Let ch_i represent the ith chain, then each chain will contain n_i packets where $n_i \leq \lceil \frac{N}{c} \rceil$ and $1 \leq i \leq c$, accordingly,

$$ch_i = \{P_i, P_{i+c}, P_{i+2c}, \ldots, P_{n_i}\}$$

For example, let $N = 16$ and $c = 4$, then we will have 4 chains, two odd and two even chains, $ch_1 = \{1, 5, 9, 13\}$, $ch_2 = \{2, 6, 10, 14\}$, $ch_3 = \{3, 7, 11, 15\}$ and $ch_4 = \{4, 8, 12, 16\}$.

Packet P_i is sent after its hash $H(P_i)$ is computed. The hash $H(P_i)$ is appended to P_{i+c} before computing the hash $H(P_{i+c})$. While both $H(P_i)$ and $H(P_{i+c})$ are appended to P_{i+2c} before computing its hash $H(P_{i+2c})$, as follows:

$$P_{i+c}||H(P_i) \rightarrow H(P_{i+c})$$
$$P_{i+2c}||H(P_i)||H(P_{i+c}) \rightarrow H(P_{i+2c})$$

where || denotes concatenation and → denotes to compute. The signature packet P_{sig_j}, where $j \geq 1$, is appended with some hashes of previous packets and signed as follow:

$$SA(H(P_{n_1})||H(P_{n_2})|| \ldots ||H(P_{n_i})) \rightarrow P_{sig_j}$$

where SA represents the signing algorithm, such as RSA.

3.2 Multiple Connected Odd-Even Chains Model

In our MCOEC model P_i is connected to P_{i+1} in addition to P_{i+c} and P_{i+2c} to increase robustness to packet loss. The signature packet is appended with three hashes of non-contiguous packets, and sent after kc packets, where $k \geq 3$, the sender will experience no delay since the hash of P_i depends on previously computed hashes. The reason to choose these packets as non-contiguous is that Internet packet loss is burst in nature, in which if a packet P_i is lost, packet P_{i+1} is likely to be lost [10],[11] and [12]. Figure 2 depicts a construction of our model when c is 8 and the signature position p is after every $3c$ packet. So as to increase the authentication probability, the packets preceding the signature are connected with those after it, that is, the authentication of the packets are not dependent on a single signature.

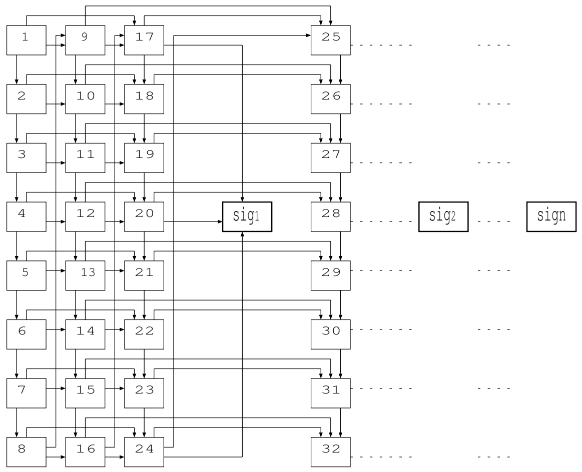

Figure 2: MCOEC model: $c = 8, p = 3c$.

When the number of chains c is 8 and signature position is $3c$, we compare two cases to connect the three packets to the signature, non-contiguous and contiguous.
Case 1: non-contiguous connection. Let the packets appended to P_{sig_1} be P_{17}, P_{20}, and P_{24}. If a burst loss of length 3 started at P_{22}, that will leave only P_{21} authenticable upon receiving P_{sig_2}. The reason is that P_{20} is connected to P_{sig_1}, so the whole packets preceding it will be authenticable upon receiving P_{sig_1}.
Case 2: contiguous connection. Let these packets be P_{22}, P_{23}, and P_{24}, having a burst of length 3 starting at P_{22} will leave the whole preceded packets authenticable upon receiving P_{sig_2}.

These two cases show that appending non-contiguous packets to the signature is better than contiguous ones for burst packet loss.

3.3 Hashes and Signatures

Table 1 shows the notation used in our model. Loss resistance and communication overhead are represented as ℓ and δ respectively.

Table 1: Notation

symbol	representation
β	number of hashes in the stream
h	hash size (SHA-1 is 16, MD5 is 10 bytes)
H	total size of all hashes in the stream
γ	number of signatures in the stream
N	number of packets in the stream
p	number of packets preceding the signature
δ	communication overhead per packet in byte
s	signature size (RSA is 128 bytes)
ℓ	loss resistance
τ	loss ratio

Since each packet is connected to three other packets and c chains exist, each packet of the first c packets P_1 to P_c in the chain contains only a single hash, that is, in total there are c hashes. While each packet of the second c packets P_{c+1} to P_{2c} in the chain contains 2 hashes of the previous packets, in total there are $2c$ hashes. Each packet of the rest of the packets P_{2c+1} to P_N contains 3 hashes. Accordingly the number of hashes β in the stream is computed as follows:

$$\beta = 3c + 3(N - 2c) = 3(N - c) \tag{1}$$

The total size of all hashes H in the stream is as follows:

$$H = h\beta \tag{2}$$

From equation (2), H depends on both N and c, since the hash size is fixed for the same hash algorithm, such as SHA-1. H increases linearly as N increases for a fixed value c. Since N is much bigger than c, the decrease of H is small when c increases.

For the signature position p, the number of signatures γ in the stream is expressed as follows:

$$\gamma = \lceil \frac{N}{p} \rceil \tag{3}$$

From the previous equation (3) we note that γ depends on p, that is, increasing p decreases the value of γ. For example, when N equals 320 packets, c is 8 and p is after $3c$, i.e., signature packet is placed after every 24 packets, then p is equal to 14. Increasing p to $5c$ reduces γ to 7.

3.4 Overhead per Packet

The communication overhead means the total size of the added information to the packet so as to be authenticated, such as hashes and digital signature. Dividing the overhead by the total number of packets in the stream, gives the overhead per packet.

Lemma 1 *The communication overhead δ in bytes per packet is as follows:*

$$\delta = \frac{H + \gamma(s + 3h)}{N} \tag{4}$$

Proof: Since the packets of the stream contain hashes and signatures in addition to data, the total of all hashes in the stream is given as H, while every signature packet contains a signature and 3 hashes of other packets, so we have $s+3h$ overhead in a signature packet. Since we have γ signatures in the stream, the overhead of all signature packets is $\gamma(s+3h)$. In total we have, $H+\gamma(s+3h)$. The overhead per packet is given by dividing $H+\gamma(s+3h)$ over N, which is δ. □

The stream size N is assumed to be known in advance for equations (1), (2), (3) and (4). When $N = p$, the number of signatures $\gamma = 1$, β and δ become as follows:

$$\beta = 3c + 3(p - 2c) = 3(p - c) \tag{5}$$

$$\delta = \frac{H + s + 3h}{p} \tag{6}$$

In the case N is unknown or infinite, the following equation is given:

$$\lim_{N\to\infty} \delta = \lim_{N\to\infty} \frac{H + \gamma(s + 3h)}{N} = 3h + \frac{3h + s}{p} \tag{7}$$

For any stream of length N, the hash size h is fixed, and γ depends on p. For that reason, p is the main factor that affects δ. Figure 3 shows how δ for different streams decreases with respect to p when c is 8. Signature position p is always after kc packets, where $k \geq 3$, increasing k decreases δ. For the stream of size 320, 1000, 2000 and 5000, the overhead per packet δ decreases 12.1%, 11.2%, 11.3% and 11.4% respectively, when increasing p from $3c$ to $20c$.

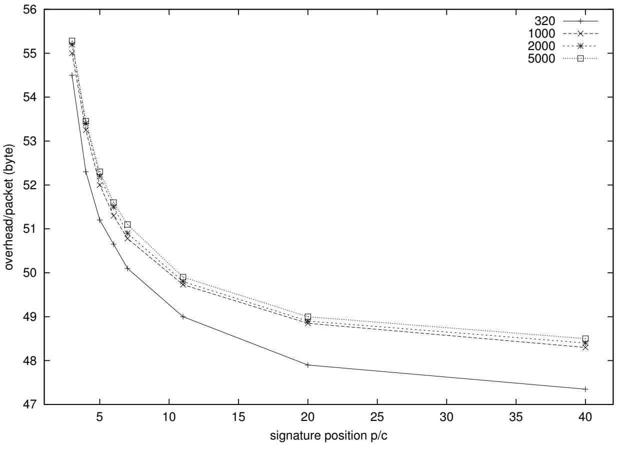

Figure 3: Overhead per packet for different streams with different signature positions when c is 8.

Figure 4 depicts δ when c is 16. The decrement in δ when increasing p from $3c$ to $20c$ is 6.7%, 5.9%, 6.0% and 6.0% for the streams of size 320, 1000, 2000 and 5000 respectively.

The chain construction mainly depends on the number of the chains c. The increase of c affects δ more than the signature position p. This effect is shown in Figure 5 for different chains c and streams, where the signature position p is after $3c$ packets. The overhead per packet δ decreases 27.5%, 16.4% and 13.8% for the streams 320, 1000 and 2000 respectively, when increasing c from 8 to 64.

4 Loss Resistance and Authentication Probability

4.1 Loss Resistance

Loss resistance ℓ is the maximum number of lost packets the scheme can resist so as to be able to authenticate the received packets. There are two kinds of losses, consecutive loss of packets known as burst loss and the other is random loss. So as to resist burst loss, the distance from packet P_i to the signature packet P_{sig_j}, where $1 \leq j \leq \gamma$,

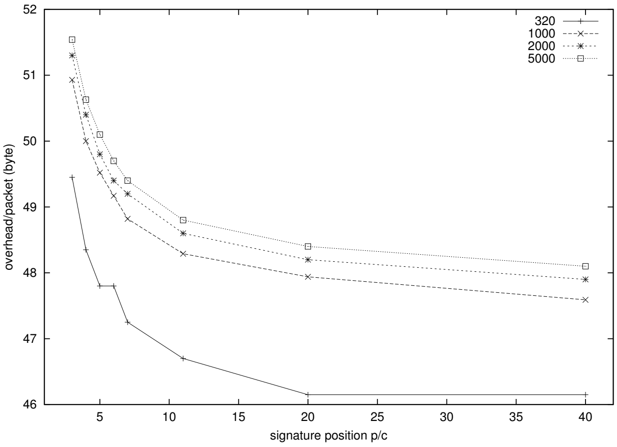

Figure 4: Overhead per packet for different streams with different signature positions when c is 16.

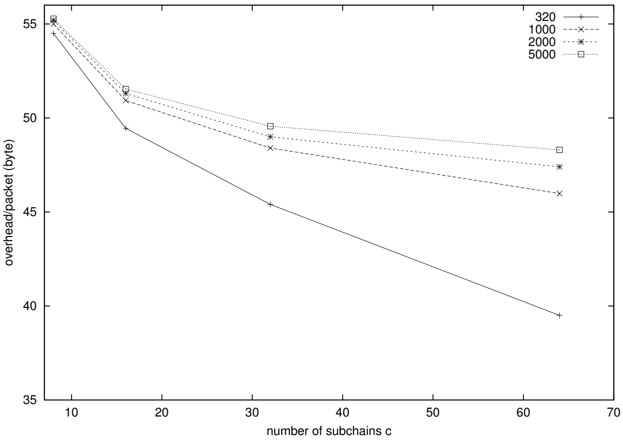

Figure 5: Overhead per packet for different streams where p is after $3c$.

must be longer than the length of the expected burst. On the other hand, random loss resistance requires more paths from P_i to P_{sig_j}, that is, appending the hash of P_i to more packets.

In our scheme we increase the path length between P_i and P_{sig_j} by increasing c, accordingly resistance ℓ to burst loss is achieved as follows:

$$\ell = 2c - 1 \tag{8}$$

In our model the hash of P_i is appended to three other packets, P_{i+1}, P_{i+c} and P_{i+2c}, so resistance to random loss depends on the probability to receive at least one of the three packets that contains the hash of P_i, this probability is shown in the next Section.

4.2 Authentication Probability

The authentication of any received packet P_i is possible if:

- the signature packet P_{sig_j} arrives,
- at least one of the three packets that contains the hash of the received packet P_i arrives, and
- at least one of the three packets that are connected to P_{sig_j} is arrived.

Since the packets preceding the first signature packet P_{sig_1} are connected to some of the packets after P_{sig_1}, the authentication of the received packets is possible even if P_{sig_1} is lost, that is, the next signature is received. For loss ratio τ, the probability P_r that at least one of γ signatures arrives is as follows:

$$P_r = 1 - \tau^{\gamma} \tag{9}$$

The probability that at least one packet out of three packets, P_{i+1}, P_{i+c} and P_{i+2c} arrives, so as to be able to authenticate the received packet P_i can be computed from similar equation as (9).

4.3 Considerations

Our model contains chains that are connected to each other through multiple points, and the number of chains c plays a main role in its efficiency. When the number of chains c increases, the packets that precede the signature one P_{sig_j} increases which leads to decrease in the total size of all hashes H, while increase the loss resistance ℓ. Moreover, when c increases, the number of signatures γ in the chain decreases, which in turn decreases the overhead per packet δ.

The more signature packets P_{sig_j} are arrived to the receiver, the higher the authentication probability is. From equation (3) one can chose γ such that, high probability to receive signatures and low overhead δ are achieved.

It is better to use a small value of c in case of small streams N, so as to increase the value of γ and achieve high authentication probability taking into consideration ℓ to be longer than the expected burst length. On the other hand, small value of c in case of large streams N makes γ large, which in turn increases δ. For that reason, increasing the value of c when N increases achieves low δ, which in turn increases ℓ.

When the signature position p increases for any stream of length N and number of chains c, γ decreases with a slight decrease in δ. So one can choose the value of p such that achieves high P_r and low δ according to N and c.

5 Buffer Capacity and Delay

The authentication of any received packet is possible if its hash is stored in two distinct locations of the received packets. The scope of any packet P_i is the maximum length from that packet to the other packet that contains its hash P_j, where $j > i$. In our model the hash of P_i is appended to P_{i+2c} at most, so the scope is $2c + 1$.

According to resources available to the sender, the chain that resists the expected burst loss can be constructed.

5.1 Sender's Buffer and Delay

The requested buffer size is equal to the scope of P_i, which is $2c + 1$. Loss resistance ℓ is dependent on c as well as requested buffer, that is, to increase ℓ, the number of chains c should be increased, in turn this will increase the requested buffer. For a sender who has unlimited buffer capacity, the number of chains c can be chosen to achieve the highest resistance ℓ with an acceptable δ and authentication probability.

On the other hand, if the buffer capacity of the sender is limited, c must be chosen such that the number of packets needed to be buffered is less or equal to the buffer capacity, while achieving loss resistance ℓ more than the expected burst length.

Let α represents the necessary buffer size, and the burst length denoted as b starts at P_{i+1}. Let the set of b lost packets denotes $B = \{P_{i+1}, \ldots, P_{i+b}\}$, the burst of length b keeps the rest of the packet sequence $S - B$ fully authenticated, that is even though b packets are lost, the received packet still can be authenticated.

The sender can always choose the number of chains c according to the available buffer capacity, so as to achieve enough resistance to the expected burst loss. The buffer capacity is then larger enough to store the scope of P_i, accordingly the following relations hold:

$$b \leq \ell \leq \alpha$$

where $\ell = 2c - 1$ and $\alpha = 2c + 1$.

So as to be able to authenticate the received packets, the scheme must resist the longest expected burst, this is achieved by choosing the value of c such that ℓ is longer than b and the buffer capacity is large enough.

For example, if the expected burst loss length $b = 10$ packets, $N = 320$, $p = 3c$ and $c = 8$, the sender needs to buffer $2c + 1$ packets and $\ell = 2c - 1$, accordingly,

$$10 \leq 15 \leq 17$$

While, if the expected burst loss length $b = 20$, the sender has to choose $c \geq 11$, so he should safe at least buffer size enough for 23 packets to achieve $\ell = 21$, accordingly,

$$20 \leq 21 \leq 23$$

5.2 Receiver's Buffer and Delay

The necessary buffer size for the receiver to authenticate the received packets depends on where the burst loss occurs and its length. If the burst loss does not include the signature packet, the necessary buffer size and experienced delay for authentication decrease. While if the burst loss includes a signature packet, the necessary buffer size and delay increase.

Let b_i, $i = 1, 2, \dots, n$, denotes the length of the burst loss i and n is the nth burst loss. Let also θ denotes the number of consecutive signatures loss, and α_1 be the number of packets the receiver needs to keep in the buffer. α_1 is equal to receiver delay.

In case a signature packet P_{sig_j} is received provided that $P_{sig_{j-1}}$ is also received, where $1 \leq j \leq \gamma$, the number of packets the receiver waits until he is able to authenticate the received packets is $\alpha_1 = p - \sum_{i=1}^{n} b_i$, since there are p packets preceding a signature and the total number of lost packets is subtracted.

While in case P_{sig_j} is received provided that all the signatures $(P_{sig_{j-\theta}}, \cdots, P_{sig_{j-1}})$ are lost, the delay in number of packets that the receiver waits and needs to buffer is given as follows:

$$\alpha_1 = (\theta + 1)p - \sum_{i=1}^{n} b_i \tag{10}$$

The connection of the packets preceding any signature packet P_{sig_j} with those after it makes the authentication of any received packets possible upon receiving any signature P_{sig_j}, where $1 \leq j \leq \gamma$. The delay and the requested buffer α_1 are increased when any signature packet is lost. Whether the receivers have different packet losses and different buffer capacities or the same, equation (10) holds when computing the delay and number of packets necessary for buffering, for each user.

For example, if $c = 8$, $p = 3c$ and a receiver experiences a burst loss length $b = 10$ which does not include the first signature packet P_{sig_1}, that is $\theta = 0$, the number of packets the receiver needs to buffer $\alpha_1 = 1(3c) - 10 = 14$ packets. While, if P_{sig_1} is lost, that is $\theta = 1$, accordingly $\alpha_1 = 2(3c) - 10 = 38$ packets.

6 Performance Evaluation

We compare our solution with two previously proposed schemes, EMSS [5] and Augmented Chain (AC)[6]. The comparison of our scheme with the EMSS and AC schemes is summarized in Table 2. The block size of all schemes is assumed p.

Table 2: Comparison of the Authentication Schemes

	EMSS	Augmented Chain	MCOEC
sender delay	1	y	1
receiver delay	p	p	p
computation overhead	$p + 1$,1	$p + 1$,1	p,1
communication overhead	variable	variable	variable
verification rate	variable	variable	variable

The criteria in Table 2 has the following meaning:

- sender delay is the delay on the sender side (in number of data packets) before the first packet in the block can be transmitted
- receiver delay is the delay on the receiver side (in number of data packets) before the verification of the first packet in the block is possible
- computation overhead is the number of hashes and signatures computed by the sender per block
- communication overhead means the size of the authentication information required for each packet
- verification rate means the number of verifiable packets of the entire stream divided by the total number of received packets of the stream

6.1 Hash Chain Construction

EMSS does not specify clearly what and how many hashes to append to each packet, neither to the signature packet. EMSS determines the best case by simulation only.

AC also does not give a clear method to determine the number of packets to insert between every two packets of the original chain. AC also does not explain clearly about the signature packet, the packets to append its hashes to the signature and the number of hashes to be appended to the signature.

Our solution specifies clearly the hashes to be appended to each packet and to the signature one, in addition to introducing a mathematical model and the loss probability.

6.2 Loss Resistance

Loss resistance achieved by EMSS depends on the way the packets are linked with each other. In case "$5 - 11 - 17 - 24 - 36 - 39$" scheme, that is, P_i is connected to

P_{i+5}, P_{i+11}, P_{i+17}, P_{i+24}, P_{i+36} and P_{i+39}, it achieves loss resistance equal to $i + 39 - i - 1 = 38$ packets. In this case the overhead is increased since every packet is appended to 6 other packets.

AC achieves loss resistance equal to $y(x - 1)$, where $x = a$ represents the strength of the chain and $y = p$ represents the sender buffer size in AC scheme. Here we use $C_{x,y}$ so as not to confuse with p in our model. When $C_{x,y} = C_{3,6}$, loss resistance equals to 12 packets.

Our solution on the other hand, achieves loss resistance given in equation (8). Note that ℓ does not depend on the number of hashes appended to each packet, it depends on c. Longer loss resistance is achieved by increasing c, and this will also reduce the overhead which is a main advantage for our scheme over those previously proposed.

We emphasize that MCOEC's advantages over the other schemes are the chain construction, mathematical equations to quantify the criteria and the loss resistance that can be achieved without increase in the overhead.

7 Conclusion

We introduced a new model for stream authentication, which is more efficient in term of loss resistance. The model basically depends on the number of chains, to increase loss resistance and reduce overhead per packet. We also introduce a mathematical equations to quantify the requirements. The sender's buffer capacity is taken into consideration when choosing the number of chains, so as to achieve the desired loss resistance.

Increasing the number of chains as the stream size increases reduces the overhead and increases loss resistance. On the other hand, when the stream size is small, reducing the number of chains increases the number of signatures which in turn increases the probability to receive signature packets. Reducing the number of chains should not exceed a limit such that desired loss resistance is not achieved.

More analysis and derivation of the authentication probability for our model is left as future work. Empirical study is going to be conducted to compare the experimental results with the theoretical ones.

References

[1] A. Perrig and J. D. Tygar, *Secure Broadcast Communication in Wired and Wireless Networks*, Kluwer Academic Publishers, 2003.

[2] R. Gennaro, and P. Rohatgi, "How to sign digital streams," Advances in Cryptology - CRYPTO'97, pp.180 – 197, 1997.

[3] W. Stallings, *Cryptography and Network Security Principles and Practices*, Prentice Hall, 2003.

[4] C. Wong and S. Lam, "Digital signatures for flows and multicasts," Technical Report TR-98-15, Dept. of Computer Sciences, University of Texas at Austin, May 1998.

[5] 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.

[6] 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.

[7] A. Chan, "A graph-theoretical analysis of multicast authentication," Proc. of the 23rd Int. Conf. on Distributed Computing Systems, 2003.

[8] 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.

[9] P. Alain and M. Refik, "Authenticating real time packet stream and multicast," Proc. of 7th IEEE Symposium on Computers and Communications, July 2002.

[10] M. Yajnik, J. Kurose, and D. Towsley, "Packet loss correlation in the mbone multicast network," Proc. of IEEE Global Internet, 1996.

[11] 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.

[12] 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.