Signature Amortization Using Multiple Connected Chains

Qusai Abuein[1] and Susumu Shibusawa[2]

[1] Graduate School of Science and Engineering
[2] Department of Computer and Information Sciences
Ibaraki University, Hitachi, Ibaraki 316-8511, Japan
{abueinq, sibusawa}@cis.ibaraki.ac.jp

Abstract. Amortization schemes for authenticating streamed data have been introduced as a solution to reduce the high overhead that sign-each schemes suffer from. The hash chains structure of amortization schemes and the number of hash values appended to other packets affect the efficiency of the authentication scheme specially against packet loss. Which packets should have hashes appended to the signature packet and how many hashes to append to it have no solutions yet. This paper introduces a new hash chain construction to achieve longer resistance against packet loss and reduces the overhead. The proposed scheme consists of multiple connected chains, each chain links several packets together. Our scheme specifies clearly how to choose the packets that should have hashes appended to a signature packet, in addition to deriving their loss probability. We study the effect of the number of hashes that are appended to a signature packet on the overhead. We introduce a measure so as to know the number of packets receivers need to buffer until they can authenticate the received packets. The number of chains of our model plays a main role in the efficiency of our scheme in terms of loss resistance and overhead.

Keywords: Multicast stream authentication, hash chain, signature amortization, web security.

1 Introduction

Digitally sign each packet using any signing algorithm such as RSA requires high computation and communication overhead and causes delay on both the sender and receivers [1], even if faster signing algorithms are used such as in [2]. Alternatively, Message Authentication Codes (MAC) are introduced such as TESLA in [3], which requires time synchronization between the sender and receivers. Another alternative is amortization schemes, such as Authentication Tree (AT) in [4], Efficient Multi-chained Stream Signature (EMSS) in [3] and Augmented Chain (AC) in [5]. In amortization schemes the stream is divided into blocks, a single packet in each block is digitally signed and the rest of the packets in the block are linked to the signed one using multiple hashes links. The

J. Dittmann, S. Katzenbeisser, and A. Uhl (Eds.): CMS 2005, LNCS 3677, pp. 65–76, 2005.

linked packets form what is known as hash chains. The security of amortization schemes is proven by Gennaro and Rohatgi in [6].

AT requires high amount of overhead, since each packet contains a signature with the authentication information so as to be individually verifiable. EMSS strengthens the resistance against packet loss of the scheme introduced by the authors of [6] by storing the hash value of each packet in multiple locations. EMSS determines the best hash chain construction by experiments and randomly chooses packets that have hashes appended to other packets.

AC uses similar strategy to EMSS so as to achieve longer resistance against packet loss, by choosing packets that have hashes appended to other packets in a deterministic way. AC does not include the means of choosing the number of packets to be inserted between each pair of the original chain.

Both EMSS and AC increases the resistance against packet loss by increasing the number of hashes in each packet, that will in turn increases the overhead [7], [8]. More details about AC analysis is found in [9], where it is applied to two case studies and compared to EMSS.

The authors of [7] and [8] give analysis of hash chains based on graph theory. They show that to increase the authentication probability, which is defined as the conditional probability that a packet is verified, the number of paths from any packet to the signature one should be increased. The main aim of Piggybacking scheme in [8] is to achieve high resistance against multiple bursts.

Signature amortization using Information Dispersal Algorithm (SAIDA) in [10], reduces the overhead of amortization schemes by using erasure codes. According to [11], the computation resources of receivers for the Forward Error Correction (FEC) in SAIDA is high comparing to that of hashes in other amortization schemes. The low computation and communication cost of hashes [12], [13] and [14] makes amortization schemes widely adopted and researchers still have high concern about it [15].

In this paper we will introduce a general construction of our Multiple Connected Chains (MC) model [16] and [17] and analyse the generalization of our model. We will also introduce a measure to determine the maximum number of hashes to be appended to the signature packet. We also show how the loss probability of the packets that have hashes appended to a signature packet is affected by their position, which in turn has a great effect on the authentication scheme. We also study the relation between the number of hashes appended to the signature packet and the overhead. Our solution is efficient in achieving longer resistance against burst packet loss and reducing overhead at the same time.

In Section 2, we introduce MC model. In Section 3, we discuss the efficiency of MC model. In Section 4, we show the required buffer and delay for both the sender and receiver. Section 5 reports a performance evaluation of our model as compared to other models followed by conclusion and future work in Section 6.

2 Multiple Connected Chains Model

Table 1 shows the notation used in MC model. A packet P_i is defined as a message M_i a sender sends to receivers along with additional authentication information. When a stream S consists of N contiguous packets we represent S as:

$$S = \{P_1, P_2, \cdots, P_N\}.$$

We introduce a Multiple Connected Chains (MC) model for multicast stream authentication using signature amortization in which a stream is divided into a number of blocks and each block consists of some packets. A single packet in each block is digitally signed and the rest of the packets are concatenated to the signed packet through hash chains in a way that allows the receiver to authenticate the received packets.

Table 1. Notation

symbol	representation
ν	number of packets containing the hash of P_i
μ	number of hashes appended to the signature
β	number of hashes appended to the packets of the stream
h	hash size (SHA-1 is 20 bytes, MD5 is 16 bytes)
H	total size of all hashes in the stream
γ	number of signatures in the stream
N	number of packets in the stream
k	number of slices in a block
c	number of chains in the stream
δ	communication overhead per packet in bytes
s	signature size (RSA is 128 bytes)
ℓ	loss resistance

A block of MC model consists of c chains, where each chain consists of some packets and the hash value $H(P_i)$ of each packet P_i is appended to packet P_{i+1} in addition to ν other packets P_{i+jc} where $j = 1, 2, \cdots, \nu$. So as for MC model to be constructed and robust against packet loss, we need the value of ν as $\nu \geq 1$. For example, when $\nu = 3$, $H(P_i)$ is appended to P_{i+1}, P_{i+c}, P_{i+2c} and P_{i+3c}. Let $A(c, \nu)$ denote a set of the packets that contain $H(P_i)$, then

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

A signature packet P_{sig} in MC model contains μ hashes of non-contiguous packets chosen from the last c packets preceding P_{sig}. The reason to choose these packets as non-contiguous 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 [2], [18], [19]. We mean by non-contiguous packets that the next packet to P_i is P_{i+j} where $j > i + 1$. On the other hand contiguous packets mean that the next packet to P_i is P_{i+1}.

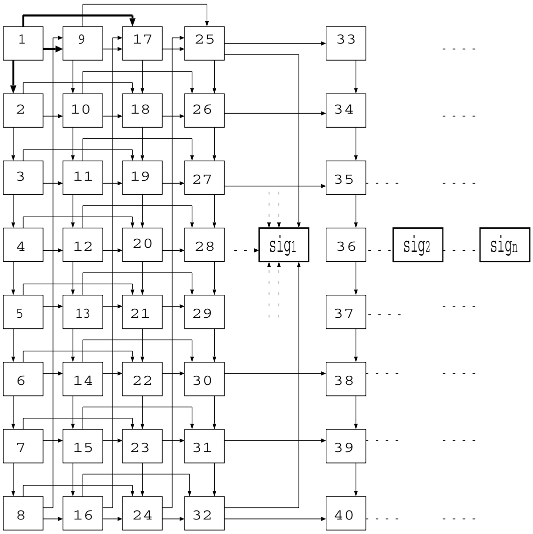

Fig. 1. A construction of MC model when $c = 8$, $k = 4$ and $\nu = 2$

Each signature packet is sent after every kc packets, which determines the block size, where k denotes the number of slices in MC model. The group of the first c packets $\{P_1, P_2, \cdots, P_c\}$ is the first slice in MC 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. 1 depicts a construction of MC model when $c = 8$, $k = 4$, and $\nu = 2$.

Let E denote a set of the last c packets preceding a signature one, then $E = \{P_{(k-1)c+1}, P_{(k-1)c+2}, \cdots, P_{kc}\}$. Let the first packet of those that have their μ hashes appended to P_{sig} chosen from E denote $P_{(k-1)c+1} = P_{j_1}$, the last one denotes $P_{kc} = P_{j_\mu}$, where μ is the number of hashes appended to a signature packet. So the set of the packets that have their μ hashes appended to P_{sig} is:

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

where $j_1 < j_2 \cdots < j_\mu$. In Fig. 1, the sender computes the hash value of the first packet $H(P_1)$, then sends P_1. A hash $H(P_{i-1})$ is appended to every packet P_i, where $2 \leq i \leq c$ before computing each packet's hash value $H(P_i)$ and then sends P_i, where $2 \leq i \leq c$. While $H(P_{i-1})$ and $H(P_{i-c})$ are appended to every packet P_i, where $c+1 \leq i \leq 2c$ before computing each packet's hash value $H(P_i)$ and then sends P_i, where $c+1 \leq i \leq 2c$. Every packet P_i, where $2c+1 \leq i \leq N$ contains $H(P_{i-1})$, $H(P_{i-c})$ and $H(P_{i-2c})$ before computing each packet's hash value $H(P_i)$ and then sends P_i, where $2c+1 \leq i \leq N$.

The sender then appends μ hashes to the signature packet P_{sig_1}, signs it and sends it. The sender experiences a single packet delay since the computations of each packet's hash value and the signature packet depend on previously computed hashes.

So in Fig. 1, the sender performs the following computation processes:

- $H(P_1)$ is computed,
- $(H(P_{i-1})||M_i) \rightarrow H(P_i)$, where $2 \leq i \leq c$,
- $(H(P_{i-1})||H(P_{i-c})||M_i) \rightarrow H(P_i)$, where $c + 1 \leq i \leq 2c$,
- $(H(P_{i-1})||H(P_{i-c})||H(P_{i-2c})||M_i) \rightarrow H(P_i)$, where $2c + 1 \leq i \leq N$,
- $SA(K, H(P_{j_1})||H(P_{j_2})|| \cdots ||H(P_{j_\mu})) \rightarrow P_{sig_1}$,

where $||$ denotes concatenation, $\rightarrow$ denotes computation, SA represents the signing algorithm, and K represents the private key. The security of these amortization schemes is proven by Gennaro and Rohatgi in [6].

The following steps describes the authentication procedure according to MC model. Since the same operations are performed on every block we only describe it for the first block:

1. Choose value of ν
2. Determine the number of chains c
3. Choose values of μ and k
4. Append necessary hash values to P_i, compute $H(P_i)$ and send P_i, $1 \leq i \leq N$
5. Choose $E(\mu)$
6. Append μ hashes to P_{sig}, sign and send P_{sig}.

3 The Efficiency of MC Model

In this section we introduce factors, such as communication overhead and number of chains, that affect the performance of the authentication scheme, equations to measure these factors and the loss probability of $E(\mu)$.

3.1 Communication Overhead

The communication overhead means the total size of the information added to a packet to authenticate it, such as hashes and digital signature. The number of packets ν, number of hashes μ and number of chains c influence the performance of the authentication scheme.

Since in MC model the packets of $E(\mu)$ are chosen as non-contiguous to each other from the last c packets preceding the signature one, the value of μ is computed as

$$\mu \leq \left\lceil \frac{c}{2} \right\rceil, \tag{1}$$

Since each packet P_i in MC model contains hashes of previous packets, P_1 contains no additional hashes. While each of the rest packets of the first slice $\{P_2, P_3, \cdots, P_c\}$ contains only a single hash, that is, in total there are $c-1$ hashes. Each packet of the second slice $\{P_{c+1}, P_{c+2}, \cdots, P_{2c}\}$ contains 2 hashes of the previous packets, so that in total there are $2c$ hashes. Each packet of the ith slice contains i hashes of previous packets where $i \leq \nu$ except for P_1. In total we have $c - 1 + 2c + 3c + \cdots + \nu c$ hashes in the first ν slices; that is, $(\frac{\nu^2+\nu}{2})c - 1$. Each packet of the remaining packets $\{P_{\nu c+1}, P_{\nu c+2}, \cdots, P_N\}$ contains $\nu+1$ hashes of previous packets. In total we have $(\nu+1)(N-\nu c)$ hashes. Accordingly, the total number of hashes β appended to the packets of stream S is computed as

$$\beta = (\frac{\nu^2+\nu}{2})c + (\nu+1)(N-\nu c) - 1. \tag{2}$$

The total size of all hashes H in the stream depends on the hash value the algorithm uses. In general H is computed as

$$H = h\beta. \tag{3}$$

Since there are kc packets in each block, the number of signatures γ in the stream is expressed as

$$\gamma = \left\lceil \frac{N}{kc} \right\rceil . \tag{4}$$

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*

$$\delta = \frac{H + \gamma(s+\mu h)}{N}. \tag{5}$$

Proof. Packets of a 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 μ hashes of other packets. Therefore, we have $s+\mu h$ overhead per signature packet. Since we have γ signatures in the stream, the overhead of all signature packets is $\gamma(s+\mu h)$. In total we have, $H+\gamma(s+\mu h)$, dividing this total by N gives the overhead per packet δ. □

The stream size N is assumed to be known in advance for the above equations. In case N is unknown or infinite, the following equation is obtained:

$$\lim_{N\to\infty} \delta = \lim_{N\to\infty} \frac{H+\gamma(s+\mu h)}{N} = (\nu+1)h + \frac{s+\mu h}{kc}. \tag{6}$$

Loss resistance ℓ is the maximum number of lost packets the scheme can sustain and still able to authenticate received packets. To resist burst loss of packets, the distance from P_i to the last packet that contains $H(P_i)$ must be longer than the expected burst packet loss length. Accordingly, in our scheme resistance ℓ against burst loss is achieved by

$$\ell = \nu c - 1 \ . \tag{7}$$

Longer resistance against burst packet loss is achieved by increasing the number of chains c in our model. The number of chains plays an important role in the efficiency of our model, so as to determine the appropriate number c, we introduce a measure regarding burst packet loss length b and the loss resistance

ℓ. The model must resist the expected burst loss b, otherwise, the authentication of the received packets preceding the start of the loss can not take place. Accordingly, $\nu c - 1 \geq b$, that is,

$$c \geq \left\lceil \frac{b+1}{\nu} \right\rceil . \tag{8}$$

Fig. 2 shows how δ for different streams decreases in terms of k when $c = 16$, $\mu = 3$, $\nu = 2$, $s = 128$ bytes, and $h = 16$ bytes. Increasing number of slices k decreases δ. For streams of sizes $N = 320$, 1000, 2000 and 5000, the overhead per packet δ decreases 6.7%, 5.9%, 6.0% and 6.0%, respectively, when increasing k from 3 to 20. Our model construction depends mainly on the number of the chains c. The increase of c affects δ positively. This effect is shown in Fig. 3 for different chains c and streams, where the number of slices $k = 3$. The overhead per packet δ decreases 27.5%, 16.6%, 14.1% and 12.5% for the streams 320, 1000, 2000 and 5000, respectively, when increasing c from 8 to 64. The effect of μ on the overhead is depicted in Fig.4, when $c = 16$, $k = 3$ and $h = 16$ bytes. The increase of δ is linear with respect to μ. The overhead increment ratio of small streams N is more than that of large ones regarding μ, that is, when N increases the increment ratio of δ becomes less regarding μ. When increasing μ from 2 to 16, the ratio is equal to 3.43%, 2.84%, 2.55% and 2.45% for the streams of sizes 320, 1000, 2000 and 5000 respectively.

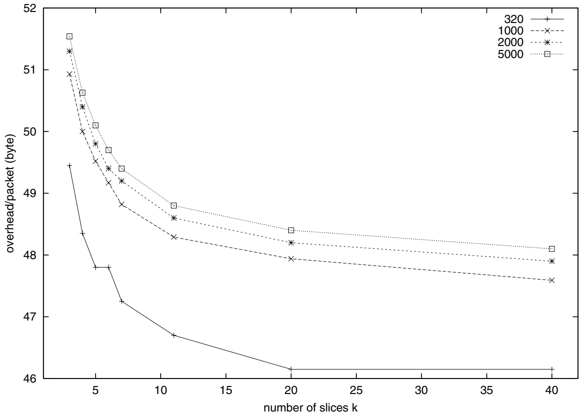

Fig. 2. Overhead per packet in terms of number of slices k for different streams when $c = 16$, $\nu = 2$ and $\mu = 3$

3.2 Characterizing Loss Probability with Gilbert Model

In this section we derive the loss probability of $E(\mu)$ in case the packets are contiguous to each other and non-contiguous using Gilbert model. Since in MC

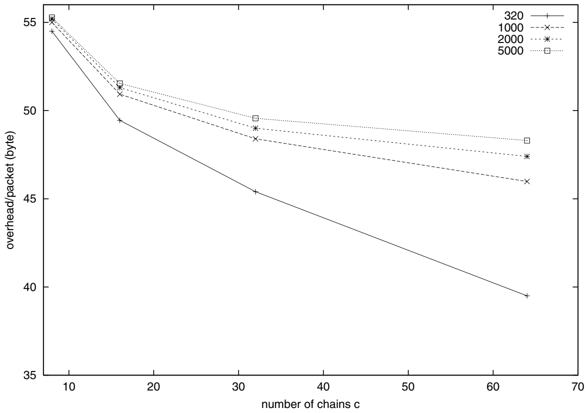

Fig. 3. Overhead per packet in terms of number of chains c for different streams where $k = 3$, $\nu = 2$ and $\mu = 3$

model the packets of $E(\mu)$ are chosen as non-contiguous to each other, the distance between each two packets of $E(\mu)$ as well as the loss probability of these packets depend on c.

Fig. 5 shows the Gilbert model that is used for characterizing burst packet loss. In the figure, r represents the probability that the next packet is lost, provided the previous one has arrived. q is the probability to transit from loss state to received state, and it is opposite to r. The transition matrix P of the Gilbert model is expressed as

$$P = \begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix} = \begin{bmatrix} 1-r & r \\ q & 1-q \end{bmatrix}, \tag{9}$$

where p_{ij} is the transition probability from state i to state j.

Lemma 2. *Let G be the set of the first signature packet P_{sig_1} and the last c packets preceding it, then $G = \{P_{(k-1)c+1}, \ldots, P_{kc}, P_{sig_1}\}$. Let P_{j_1} and P_{j_μ} be chosen as $P_{(k-1)c+1}$ and P_{kc}, respectively. Let ρ_1 be the loss probability of non-contiguous packets chosen from G whose hashes are appended to the signature one, and ρ_2 be the loss probability of contiguous ones. According to the Gilbert model ρ_1 and ρ_2 are given by*

$$\rho_1 = (1-r)^{c-2\mu+1} \cdot r^{\mu} \cdot q^{\mu}, \quad c > 2\mu - 1, \tag{10}$$

and

$$\rho_2 = (1-r)^{c-\mu} \cdot r \cdot (1-q)^{\mu-1} \cdot q, \quad c > \mu. \tag{11}$$

Proof. In case of non-contiguity in our model, the first packet is $P_{(k-1)c+1}$, the last one is P_{kc}, and the others are any non-contiguous packets in between the first and the last. When the packets that have hashes appended to P_{sig_1} are lost and the rest packets of G are received, we have μ transitions from non-loss state

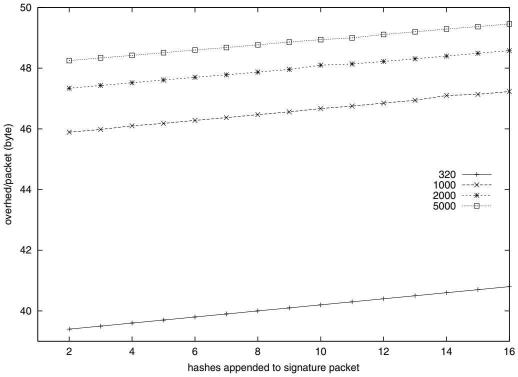

Fig. 4. Overhead per packet for different streams in terms of μ when $c = 16$ and $k = 3$

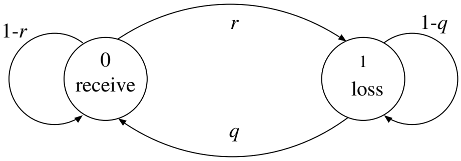

Fig. 5. The Gilbert model for burst packet loss

to loss state and μ inverse transitions. The other transitions are from non-loss state to non-loss state, and the total number is $c - 2\mu + 1$. Accordingly,

$$\rho_1 = (1 - r)^{c-2\mu+1} \cdot r^{\mu} \cdot q^{\mu}$$

In case of contiguity, packets that have their μ hashes appended to P_{sig_1} are $P_{kc-\mu}$, ..., P_{kc-1} and P_{kc}. When all of these packets are lost and the other packets of G are received, the loss probability ρ_2 can be derived similarly. □

Lemma 3. *In the Gilbert model, let r be the probability that the next packet is lost, provided that the previous one has arrived, and q be the opposite probability. Then, the relation between loss probabilities ρ_1 and ρ_2 given in Lemma 2 is*

$$\rho_1 \le \rho_2 \qquad \textit{when} \quad r + q \le 1,$$

and

$$\rho_1 > \rho_2 \qquad \textit{when} \quad r + q > 1.$$

Proof. Dividing ρ_2 by ρ_1, we get

$$\frac{\rho_2}{\rho_1} = (\frac{1-r}{r} \times \frac{1-q}{q})^{\mu-1}.$$

When $\frac{1-r}{r} \times \frac{1-q}{q} > 1$, we get $r + q < 1$ and $\rho_1 < \rho_2$. On the other hand, when $\frac{1-r}{r} \times \frac{1-q}{q} < 1$, we get $r + q > 1$ and $\rho_1 > \rho_2$. □

The loss probability of the packets that have their hashes appended to the signature one in case of contiguity ρ_2 is equal to or greater than that of non-contiguity ρ_1 when $r + q \leq 1$ and smaller when $r + q > 1$.

4 Buffer Capacity and Delay

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 $P_{i+\nu c}$ at most, and so the scope is $\nu c + 1$. The details of the buffer capacity and delay can be found in [16], [17].

4.1 Sender's Buffer and Delay

The sender experiences a delay of a single packet, since the last packet of a block is signed and it depends on previously computed hashes. The requested buffer size denoted as α, is equal to the scope of P_i, so according to available buffer resources the sender can always choose the number of chains c so as to achieve sufficient resistance to the expected burst packet loss denoted as b. The buffer capacity is then large enough to store the scope of P_i. Accordingly the following relation holds $b \leq \ell \leq \alpha$.

4.2 Receiver's Buffer and Delay

The necessary buffer size for the receiver denoted as α_1 to authenticate the received packets depends on the following factors: the start of the burst loss, its length and the loss of the signature packet. When θ denote the number of consecutive signatures loss and n denotes the number of bursts, the following measures the necessary buffer and delay in number of packets the receiver waits:

$$\alpha_1 = (\theta + 1)kc - \sum_{i=1}^{n} b_i \tag{12}$$

5 Performance Evaluation

In this section we compare our solution with previously proposed schemes, EMSS [2] and AC [3]. The comparison is in terms of hash chain construction and loss resistance.

In terms of hash chain construction, The EMSS does not specify what and how many hashes to be appended to each packet, or to the signature packet. EMSS only determines the best case of the chain construction to achieve high robustness against packet loss by simulation.

AC also does not give a clear method to determine the number of packets to be inserted between every two packets of the original chain. Also, it does not explain clearly the signature packet, which packets to append their 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.

Loss resistance achieved by EMSS depends on the way that the hash of a packet is appended to other packets. In the case of the scheme "$5-11-17-24-36-39$", that is, the hash of P_i is appended to P_{i+5}, P_{i+11}, P_{i+17}, P_{i+24}, P_{i+36}, and P_{i+39}, an EMSS achieves loss resistance equal to $i+39-i-1=38$ packets. For EMSS to increase loss resistance the hash of P_i should be appended to more packets, which in turn increases the overhead.

The AC achieves loss resistance equal to $p(a-1)$, where a represents the strength of the chain, and p represents the sender buffer size in the AC scheme. When $C_{a,p} = C_{3,6}$, loss resistance is equal to 12 packets. The way AC can increase the resistance against packet loss is by increasing p or a, which means to append more hashes to other packets that in turn increases the overhead.

Our solution on the other hand, achieves the loss resistance equal to $\ell = \nu c - 1$ as given by equation (7). Note that ℓ does not depend on the number of hashes appended to each packet and requires no extra computation resources, rather it depends on the number of chains c. Longer loss resistance is achieved by increasing c, and this will also reduce the overhead, which is the major advantage of our scheme over those previously proposed.

6 Conclusion

We introduced a generalization of our MC model for signature amortization to authenticate multicast streams. We analyzed the generalization and introduced a measure to determine the maximum number of hashes to be appended to the signature packet. The loss probability of the packets with hashes appended to the signature packet in case they are non-contiguous and contiguous are also discussed. The effect of the number of hashes that are appended to the signature packet on the overhead for different streams is also provided.

Our scheme achieves greater loss resistance against packet loss and lower overhead by increasing the number of chains of our model. The buffer capacity needed by the sender when constructing our model so as to achieve the desired resistance against packet loss is studied. The receivers buffer capacity and delay increases as the loss of signature packets increases.

As future works, derivation of the authentication probability for our model and discuss the optimal values of the number of chains, number of slices, μ and ν of our model. We will also conduct an empirical study to see the performance of our method and compare it to the performance of the existing methods.

References

1. 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)

2. 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 (2000)
3. 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 (2000) 56-73
4. C. K. Wong and S. S. Lam: Digital signatures for flows and multicasts. IEEE/ACM Trans. on Networking, **7** (1999) 502-513
5. P. Golle and N. Modadugu: Authenticating streamed data in the presence of random packet loss. Proc. of ISOC Network and Distributed System Security Symposium (2001) 13-22
6. R. Gennaro, and P. Rohatgi: How to sign digital streams. Advances in Cryptology - CRYPTO'97 (1997) 180-197
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 (2001) 232-246
9. P. Alain and M. Refik: Authenticating real time packet stream and multicast. Proc. of 7th IEEE Symposium on Computers and Communications (2002)
10. J. Park, E. Chong and H. Siegel: Efficient multicast stream authentication using erasure codes. ACM Trans. on Information and System Security, **6** (2003) 258-258
11. T. Cucinotta, G. Cecchetti and G. Ferraro: Adopting redundancy techniques for multicast stream authentication. Proc. of the $9th$ IEEE Workshop on FTDCS (2003)
12. A. Perrig and J. D. Tygar: Secure Broadcast Communication in Wired and Wireless Networks. Kluwer Academic Publishers (2003)
13. W. Stallings: Cryptography and Network Security Principles and Practices. Prentice Hall (2003)
14. 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 (1998)
15. A. Lysyanskaya, R. Tamassia, and N. Triandopoulos: Multicast authentication in fully adversarial networks. Proc. of IEEE Symposium on Security and Privacy (2004) 241-255
16. Q. Abuein and S. Shibusawa: Efficient multicast authentication scheme using signature amortization. Proc. of the IASTED Int. Conf. on CIIT (2004)
17. Q. Abuein and S. Shibusawa: New chain construction for multicast stream authentication. Proc. of the ICENCO Int. Conf. on NTIS (2004)
18. 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. (2000)
19. M. Yajnik, J. Kurose, and D. Towsley: Packet loss correlation in the mbone multicast network. Proc. of IEEE Global Internet (1996)