AMO - Advanced Modeling and Optimization, volume 7, number 2, 2005

A Graph-based New Amortization Scheme for Multicast Streams Authentication

Qusai Abuein
Graduate School of Science and Engineering

Susumu Shibusawa
Department of Computer and Information Sciences

Ibaraki University
Hitachi, Ibaraki 316-8511, Japan
{abueinq, sibusawa}@cis.ibaraki.ac.jp

Abstract

We present a graph-based new amortization scheme for multicast streams authentication that achieves stronger resistance against packet loss and reduces the overhead in the same time. The hash chains of the existing amortization schemes have no systematic way to construct them, the construction had been determined by simulation. These schemes lack the theoretical model that helps in determining the values of the parameters and measuring the efficiency metrics of the authentication schemes. The proposed scheme in this paper consists of multiple connected chains, each chain connects some packets together. The parameters values of the hash chain of our scheme is determined in advance and the hash chain is constructed systematically. We also introduce mathematical tools that help in measuring the efficiency metrics such the authentication probability, the loss resistance and the overhead. The number of chains in our model plays the main role in the efficiency of our scheme in terms of loss resistance and overhead. High authentication probability can be achieved by increasing the number of hashes that are appended to the signature packet and by increasing the number of packets that contain the hash of a previous packet.

Key Words: Internet security, multicast stream authentication, hash chain graph, signature amortization, authentication probability

1 Introduction

Multicast is an efficient way of transmitting information, since a single copy of packets is sent by a sender and delivered to every receiver within the multicast group via multicast-enabled routers. A stream is defined as a very long or infinite sequence of bits that the sender sends to the receivers who must consume the data it receive at the input rate. Some examples of multicast streams

include TV and radio broadcasts, stock quotes, videoconferencing, software update and online gaming. Security is a challenge issue due to the large number of sent packets and receivers, packet loss, delay and overhead. Signing each packet in the stream to authenticate it results in a great computation and communication overhead on both the sender and receivers, even if fast signing algorithms are sued [Rohatgi, 1999], [Wong et al., 1999].

The practical and efficient alternative that reduces the overhead is to amortize a single signature over a group of packets, the schemes that use such an alternative are known as amortization schemes or hash chained schemes. Several amortization schemes [Gennaro et al., 1997], [Perrig et al., 2000], [Golle et al., 2001] have been introduced as a solution for authenticating multicast streams to reduce the high cost of sign-each schemes. In amortization schemes, a stream is divided into blocks; each block consists of some packets. A single packet in the block, usually the first or the last one, is digitally signed using any signing algorithm. The hash value of each packet of the remaining packets of a block is computed using any hashing function. The packets of a block are linked together through multiple links using concatenation function to achieve robustness against packet loss forming what is known as hash chains. Hashes of some packets are appended to a signature packet before it is signed so that the receivers can verify the received packets.

The Efficient Multi-chained Stream Signature (EMSS) [Gennaro et al., 1997] and Augmented Chain (AC) [Golle et al., 2001] amortization schemes append the hash of a packet to several other packets in order to increase robustness against packet loss and achieve higher authentication probability which is defined as the conditional probability that a packet is verified. EMSS and AC also solved the weak robustness against packet loss of Gennaro and Rohatgi's scheme [Gennaro et al., 1997]. But this solution increases the overhead and is chosen randomly, there is no systematic way that tells what and how many packets should contain the has value of other packets. The best construction had determined by simulation. The solution of multicast stream authentication using signature amortization schemes is an efficient one in terms of overhead [Lysyanskaya et al., 2004], [Christophe, 2005]. However, how to construct the hash chains remains an open problem [Chan, 2003].

To our knowledge, there is no amortization scheme that achieves stronger resistance against loss and reduces the overhead in the same time, so such scheme is necessary. Also the parameters values of the hash chain should be easily determined to build the best construction that achieves the desired performance, that is, achieve a successful authentication with low overhead despite the presence of loss. Also the hash chain topology of the existing schemes and being unable to determine the values of the parameters of the hash chains in advance make it hard to derive an exact formula to characterize the efficiency metrics such as the authentication probability of such scheme.

In this paper we introduce a graph-based new hash chain construction for amortization schemes that achieves stronger resistance against packet loss and reduces the overhead in the same time. The Multiple Connected Chains (MC) model consists of multiple chains connected to each other and each chain connects some packets together [Abuein et al., 2004a, 2004*b*, 2005*a*, 2005*b*]. The number of packets that contain the hash of a previous packet is not changed as the packet loss increases, instead the number of chains is changed and it plays the main role in the efficiency of MC model. Increasing the number of chains of MC model achieves stronger resistance against loss and reduces the overhead in the same time.

The parameters of MC model can be easily determined in advance, which helps in measuring its efficiency metrics such as authentication probability. In this paper we derive the authentication probability of MC model using binomial model and 2-state Markov model, also known as Gilbert model. Burst packet loss is best characterized using Gilbert model as reported by [Sanneck et al., 2000] and [Jiang et al., 2000].

This paper is organized as follows: Section 2 introduces the MC model. In Section 3 we analyze the efficiency of our scheme in terms of overhead and in Section 4 in terms of loss resistance. The authentication probability of our scheme is derived and analyzed in Section 5. In Section 6 we show the required buffer and delay for both the sender and receiver. In Section 7 we evaluate the performance of our scheme and in Section 8 we present previous works on stream authentication schemes. In Section 9 we give the conclusion of our study.

2 Multiple Connected Chains Model

Table 1 shows the notation used in this paper. A packet P_i is defined as a message M_i a sender sends to receivers along with additional authentication information. We introduce a Multiple Connected Chains (MC) model for multicast stream authentication using signature amortization that achieves stronger resistance against packet loss and reduces the overhead in the same time. Our model using amortization schemes divides a stream of N messages into blocks, where each block consists of some messages. 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 using his digital key. The signed packet is called a signature packet P_{sig}. The sender sends a signature packet at the end of each block. Appending hashes to other packets and to the signature packets enable the receivers to authenticate the received packets.

The hash $H(P_i)$ of each packet P_i in MC model is appended to ν other packets as P_{i+1} and P_{i+jc}, where $j = 1, 2, \cdots, \nu - 1$. For example, when $\nu = 3$, $H(P_i)$ is appended to P_{i+1}, P_{i+c} and

Table 1: Notation

symbol	representation
N	the number of messages in the stream
c	the number of chains in MC model
k	the number of slices in a block
ν	the number of packets that contain the hash of 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

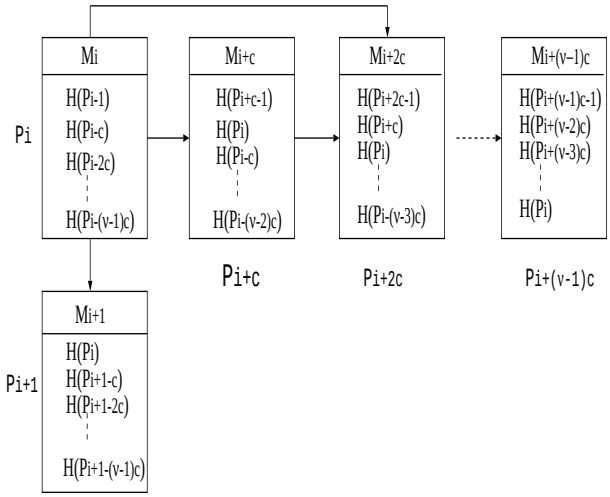

Figure 1: Appending hashes to other packets in MC model.

P_{i+2c}. Let $A(c, \nu)$ denote a set of the packets that contain $H(P_i)$, then:

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

Figure 1 shows the appended hashes to each packet according to MC, when $\nu = 3$. So as to construct MC 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 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 $j_1 < j_2 \cdots < j_\mu$.

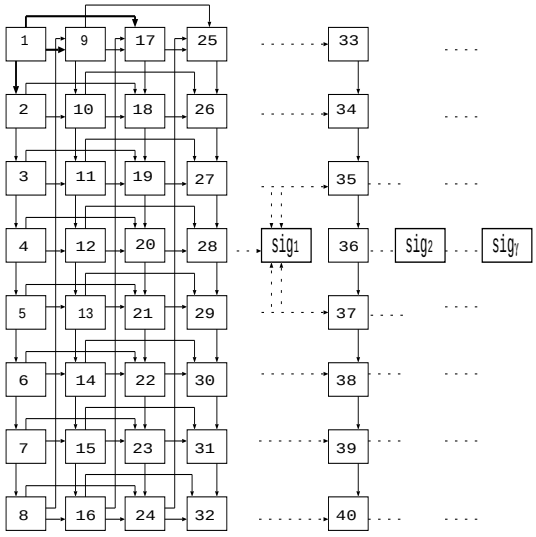

Figure 2: A construction of MC model for $c = 8$, $k = 4$ and $\nu = 3$.

MC model consists of c chains, where each chain consists of some packets. The block size of MC 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 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.

Figure 2 depicts a construction of MC model for $c = 8$, $k = 4$ and $\nu = 3$. the sender computes the hash of the first message M_1 without appending additional information to it since there are no messages preceding it, then sends P_1, so $P_1 = M_1$. The sender constructs P_i by concatenating the hash $H(P_{i-1})$ with every message M_i, then computes $H(P_i)$ and sends P_i, where $2 \leq i \leq c$. While he constructs P_i by concatenating the hashes $H(P_{i-1})$ and $H(P_{i-c})$ with every message M_i, then computes $H(P_i)$ and sends P_i, where $c + 1 \leq i \leq 2c$. 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 sends P_i, where $2c + 1 \leq i \leq N$.

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}. The sender experiences a single packet delay since the necessary data for computing the hash value of any packet and the signature packet depend on other hash values that are already computed.

The following steps describes the authentication procedure the sender performs on each block according to MC model:

1. $P_1 := M_1$,

2. Compute $H(P_1)$, send P_1,

3. $P_i := (H(P_{i-1})||H(P_{i-c})||H(P_{i-2c})||\cdots||H(P_{i-(\nu-1)c})||M_i)$, where $2 \leq i \leq ck$,

4. Compute $H(P_i)$, where $2 \leq i \leq ck$, send P_1,

5. $P_{sig} := SA(K, H(P_{j_1})||H(P_{j_2})||\cdots||H(P_{j_\mu}))$,

6. Send P_{sig}

where := denotes computation, || denotes concatenation, SA represents the signing algorithm and K represents the digital key.

The receiver performs the verification steps in an opposite manner, that is, verifies the signature packet first, retrieve $H(P_{j_1}), H(P_{j_2}), \cdots, H(P_{j_\mu})$. If the signature packet is secure then the hashes appended to it are considered secure too. When receiving the remaining packets of the block, the receiver computes the hash value of each packet starting from the last packet in the block, compares the computed value to the retrieved one, if both are equal then that packet is considered secure. After verifying the first packet the receiver starts using the received packets.

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

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

Definition 1 *The communication overhead per packet δ in bytes is the total size of the hashes that are appended to the whole packets and the total size of the signature packets divided by N, so:*

$$\delta = \frac{h\beta + s\gamma}{N} \tag{4}$$

□

Multiplying the hash value h by β gives the total size of all hashes that are appended to the whole packets, while multiplying the size of a signature packet s by the total number of signature packets $\gamma = \lceil \frac{N}{kc} \rceil$ gives the total size of signature packets. Solving Equation (4) accordingly, gives the following:

$$\delta = \frac{h}{N}((\frac{\nu - \nu^2}{2})c - 1) + h\nu + \frac{s}{N}\left\lceil \frac{N}{ck} \right\rceil \tag{5}$$

The stream size N is assumed to be known in advance for Equations (3), 4 and (5). In the case N is unknown or infinite, the following equation is given:

$$\lim_{N\to\infty} \delta = \lim_{N\to\infty} \left\{ \frac{h}{N}((\frac{\nu - \nu^2}{2})c - 1) + h\nu + \frac{s}{N}\left\lceil \frac{N}{ck} \right\rceil \right\}$$

$$= h\nu + \frac{s}{ck} \tag{6}$$

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. Figure 3 depicts δ in terms of k for different streams when $c = 16, \nu = 3, s = 128$ bytes and $h = 64$ bytes. While the decrease of δ in terms of c is depicted in Figure 4 when $k = 3$.

We showed how to measure the overhead per packet according to MC 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 φ, 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 4.

The appropriate value of μ can be chosen in the same manner ν have been chosen. While we choose the packets of $E(\mu)$ 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

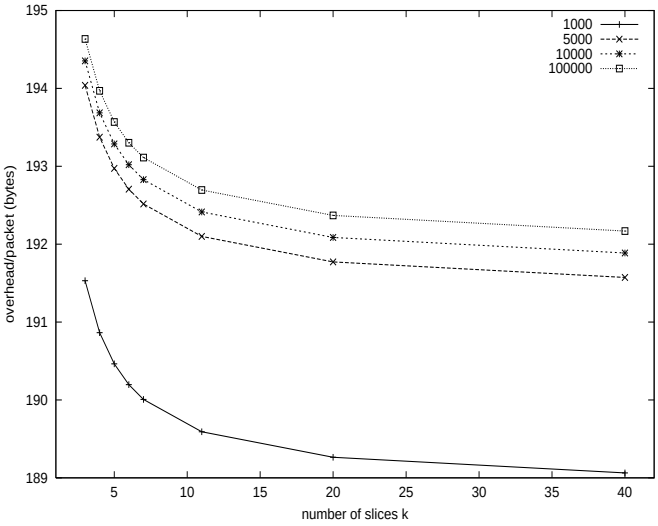

Figure 3: Overhead per packet in terms of number of slices k for different streams when $c = 16$, $\nu = 2$, $s = 128$ and $h = 64$.

in nature, and if a packet P_i is lost, packet P_{i+1} is likely to be lost [Sanneck et al., 2000], [Jiang et al., 200], [Yajnik et al., 1996].

4 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 contains the hash $H(P_i)$ of a packet P_i according to MC model. So loss resistance is equal to the number of packets between P_i and $P_{i+(\nu-1)c}$, accordingly:

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

Equation (7) 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

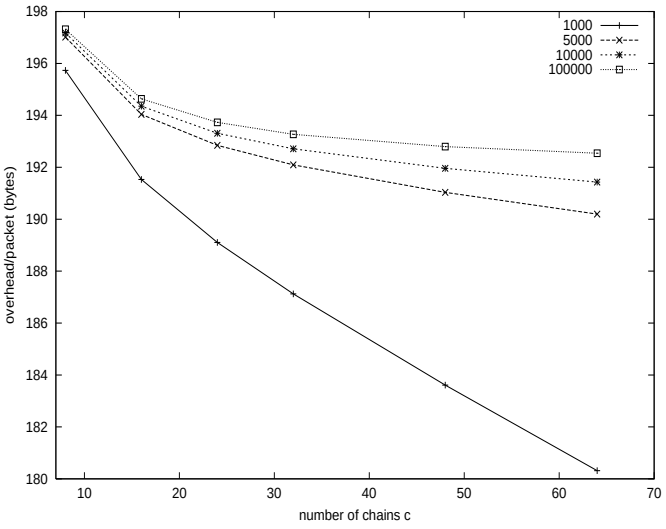

Figure 4: Overhead per packet in terms of number of chains c for different streams when $k = 3$, $\nu = 2$, $s = 128$ and $h = 64$.

packets that lies before the start of the burst becomes impossible. Accordingly, $(\nu - 1)c - 1 \geq b$, that is:

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

5 Authentication Probability

The authentication probability is an important metric to measure the efficiency of the authentication scheme. In this section we derive the authentication probability of our scheme using binomial and 2-state Markov model and analyze the authentication probability in terms of several parameters.

According to MC 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}.

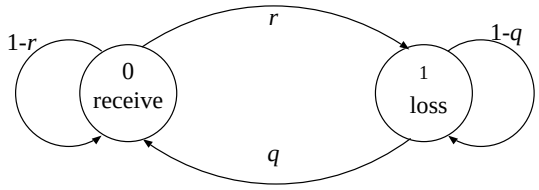

Figure 5: 2-state Markov model for burst packet loss.

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:

$$P_r\{P_i\} = P_r\{P_i \; is \; verifiable \mid P_i \; is \; received \; \} \tag{9}$$

5.1 Authentication Probability Using Binomial Model

In binomial model, let τ denote the packet loss ratio. According to MC model, a packet P_i can not be authenticated when all the packets of $E(\mu)$ or all the packets of $A(c,\nu)$ or both are lost. This includes the following cases: first, when all $A(c,\nu)$ are lost and $E(\mu)$ is the combination of lost and receive. The second case is when all $E(\mu)$ are lost and $A(c,\nu)$ is the combination of lost and receive, while the last case is when all of $A(c,\nu)$ and all of $E(\mu)$ are lost.

The probability that all the packets of $A(c,\nu)$ are lost is τ^{ν}, the probability that all the packets of $E(\mu)$ are lost is τ^{μ} and the probability that all the packets of $A(c,\nu)$ and all the packets of $E(\mu)$ are lost is $\tau^{\nu+\mu}$. Accordingly, the following Lemma gives the authentication probability $P_r\{P_i\}$ based on binomial model.

Lemma 1 *Based on binomial model, the authentication probability of the packet P_i in a block of MC model is given as follows, when $i + (\nu - 1)c \leq j_1$:*

$$P_r\{P_i\} = 1 - \tau^{\nu} - \tau^{\mu} + \tau^{\nu+\mu}. \tag{10}$$

Proof: The probability of the first case of those that can not authenticate P_i is $\tau^{\nu}(1-\tau^{\mu})$, the probability of the second case is $\tau^{\mu}(1-\tau^{\nu})$, while the probability of the last case is $\tau^{\nu}\tau^{\mu}$. Excluding the probabilities of these three cases gives the authentication probability: $P_r\{P_i\} = 1 - (\tau^{\nu}(1-\tau^{\mu}) - \tau^{\mu}(1-\tau^{\nu}) - \tau^{\nu}\tau^{\mu})$. Solving the last formula gives the desired result, $P_r\{P_i\}$. □

To achieve high authentication probability $P_r\{P_i\}$ of our scheme using binomial model the values of ν and μ should be increased as the expected loss ration τ increases. Figure 6 shows the required value of ν in terms of τ so as to always guarantee that at least one packet of ν is received with different probabilities φ, 90%, 95%, 97% and 99%. The required value of ν increases as the desired φ or the expected τ increase.

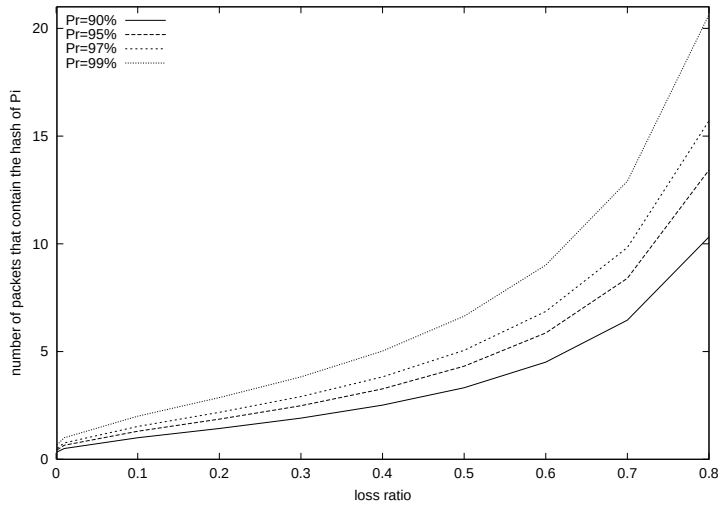

Figure 6: Required value of ν in terms of τ to achieve the desired φ.

Figure 7 depicts the authentication probability $P_r\{P_i\}$ in terms of ν for different values of μ when the loss ratio $\tau = 5\%$. The required values of ν and μ increases as τ increases so as to achieve higher authentication probability.

5.2 Authentication Probability Using 2-State Markov Model

The burst packet loss is well characterized using 2-state Markov model [Sanneck et al., 2000], [Jiang et al., 200]. Figure 5 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. The transition matrix P of the 2-state Markov model is expressed as:

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

where p_{ij} is the transition probability from state i to state j. The n step transition matrix $P^{(n)}$ of the 2-state Markov model is as follows:

$$P^{(n)} = \begin{bmatrix} p_{00}^{(n)} & p_{01}^{(n)} \\ p_{10}^{(n)} & p_{11}^{(n)} \end{bmatrix} \tag{12}$$

where $p_{ij}^{(n)}$ is the n step transition probability from state i to state j.

According to 2-state Markov model depicted in Figure 5, receive and loss states are denoted 0 and 1, respectively. Table 2 shows the combination of the transition states when $\nu = 2$, $\mu = 2$ and the transition probabilities that authenticate P_i, where P_{rec} means the receive probability.

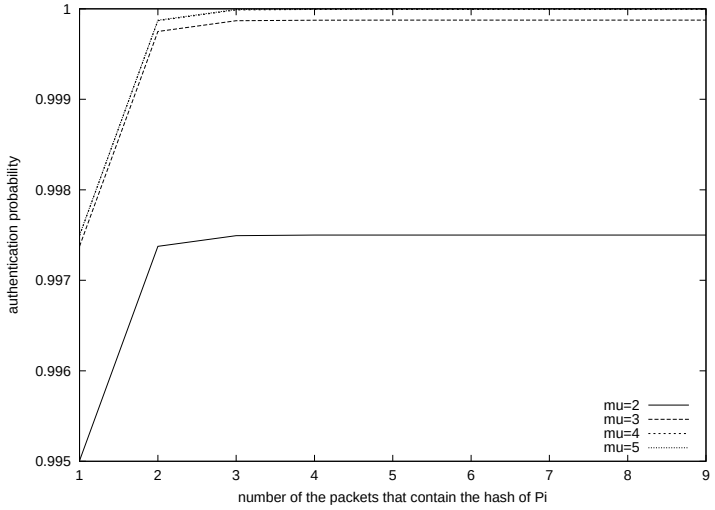

Figure 7: Authentication probability using binomial model in terms of ν for different values of μ when τ is 5%.

Note that packet P_i is always assumed to be received. While all the whole packets of $A(c, \nu)$ or $E(\mu)$ can not be lost. As an example, the first case 00101 of Table 2 means that packets P_i, P_{i+1} and P_{j_1} are received while packets P_{i+c} and P_{j_2} are lost. There is only one step from packet P_i to P_{i+1}, there are $(c-1)$ steps from P_{i+1} to P_{i+c}, there are $(j_1 - i - c)$ steps from P_{i+c} to P_{j_1} and $(j_2 - j_1)$ steps from P_{j_1} to P_{j_2}. So the transition probability of the first case of Table 2 is $p_{00}p_{01}^{(c-1)}p_{10}^{(j_1-i-c)}p_{01}^{(j_2-j_1)}$.

Based on 2-state Markov model, the number of cases m that authenticate P_i is:

$$m = (2^{\nu} - 1) \times (2^{\mu} - 1) \tag{13}$$

Since packet P_i is assumed to be received, its authentication probability $P_r\{P_i\}$ is the total of the transition probabilities. The first transition state starts from 0 since P_i is received, to either 0 or 1; that is, P_{i+1} is either received or lost, where the value of g_1 is either 0 or 1. The second transition state goes from state g_1 to either 0 or 1; that is, P_{i+c} is either received or lost. The remaining transition states go from g_2 to g_3, then from g_3 to $g_4, \cdots$, until g_ν, where $g_l,\ l = 1, 2, \cdots, \nu$, is either 0 or 1.

The next transition state goes from g_ν to either 0 or 1, which means P_{j_1} is either received or lost. The next transition state goes from h_1 to h_2; that is, P_{j_2} is either received or lost. The remaining transition states go from h_2 to h_3, then from h_3 to $h_4, \cdots$, until h_μ, where $h_l,\ l = 1, 2, \cdots, \mu$, is either 0 or 1.

Theorem 1 *Based on 2-state Markov model the authentication probability of the ith packet P_i in*

Table 2: Transition states and probabilities of P_i when $\nu = 2$ and $\mu = 2$.

	A(c, ν)		E(μ)		
P_i	P_{i+1}	P_{i+c}	P_{j_1}	P_{j_2}	P_{rec}
0	0	1	0	1	$p_{00}p_{01}^{(c-1)}p_{10}^{(j_1-i-c)}p_{01}^{(j_2-j_1)}$
0	1	0	0	1	$p_{01}p_{10}^{(c-1)}p_{00}^{(j_1-i-c)}p_{01}^{(j_2-j_1)}$
0	0	1	1	0	$p_{00}p_{01}^{(c-1)}p_{11}^{(j_1-i-c)}p_{10}^{(j_2-j_1)}$
0	1	0	1	0	$p_{01}p_{10}^{(c-1)}p_{01}^{(j_1-i-c)}p_{10}^{(j_2-j_1)}$
0	0	1	0	0	$p_{00}p_{01}^{(c-1)}p_{10}^{(j_1-i-c)}p_{00}^{(j_2-j_1)}$
0	1	0	0	0	$p_{01}p_{10}^{(c-1)}p_{00}^{(j_1-i-c)}p_{00}^{(j_2-j_1)}$
0	0	0	0	1	$p_{00}p_{00}^{(c-1)}p_{00}^{(j_1-i-c)}p_{01}^{(j_2-j_1)}$
0	0	0	1	0	$p_{00}p_{00}^{(c-1)}p_{01}^{(j_1-i-c)}p_{10}^{(j_2-j_1)}$
0	0	0	0	0	$p_{00}p_{00}^{(c-1)}p_{00}^{(j_1-i-c)}p_{00}^{(j_2-j_1)}$

a block of MC is given as follows, when $i + (\nu - 1)c \leq j_1$*:*

$$P_r\{P_i\} = \sum_{g,h} \left\{ \left[p_{0g_1} p_{g_1g_2}^{(c-1)} \prod_{l=2}^{\nu-1} (p_{g_lg_{l+1}}^{(c)}) \right] \left[p_{g_\nu h_1}^{(j_1-i-(\nu-1)c)} \prod_{l=1}^{\mu-1} (p_{h_lh_{l+1}}^{(j_{l+1}-j_l)}) \right] \right\} \tag{14}$$

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 $p_{g_1g_2}^{(c-1)}$. 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 $\prod_{l=2}^{\nu-1}(p_{g_lg_{l+1}}^{(c)})$, and in total we have transition probability $p_{0g_1}p_{g_1g_2}^{(c-1)}\prod_{l=2}^{\nu-1}(p_{g_lg_{l+1}}^{(c)})$. 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 $\prod_{l=1}^{\mu-1}(p_{h_lh_{l+1}}^{(j_{l+1}-j_l)})$. The total of the whole transition probabilities gives the desired result. □

The authentication probability $P_r\{P_i\}$ of Theorem 1 is applied for a single block of MC model, while $P_r\{P_i\}$ can be applied for any block of MC model, which is given as follows:

Corollary 1 *When any signature packet* P_{sig_u} *is received, where* $1 \leq u \leq \gamma$ *and* $i + (\nu-1)c \leq j_1 < j_2 \cdots < j_\mu < uck$*, based on 2-state Markov model the authentication probability of the* i*th packet* P_i *is given as follows:*

$$P_r\{P_i\} = \sum_{g,h} \left\{ \left[p_{0g_1} p_{g_1g_2}^{(c-1)} \prod_{l=2}^{\nu-1} (p_{g_lg_{l+1}}^{(c)}) \right] \left[p_{g_\nu h_1}^{(j_1-i-(\nu-1)c)} \prod_{l=1}^{\mu-1} (p_{h_lh_{l+1}}^{(j_{l+1}-j_l)}) \right] \right\} \tag{15}$$

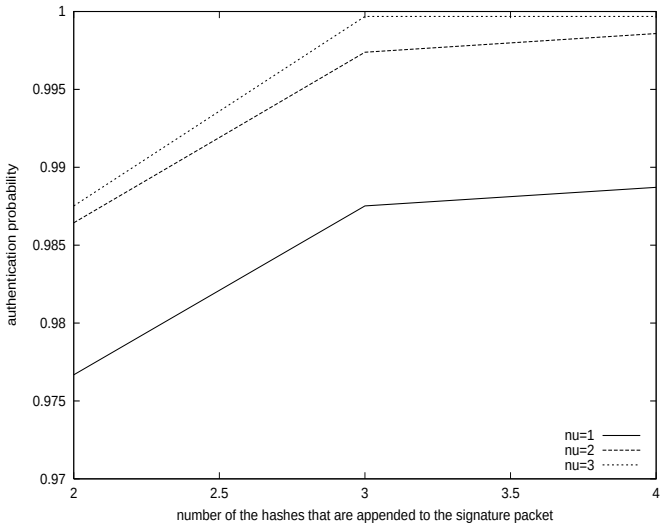

Figure 8: Authentication probability using 2-state Markov model in terms of μ, when $c = 8$, $k = 5$, $r = 0.1$ and $q = 0.8$.

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: Similar to that of Theorem 1 □.

We show the effect of the parameters ν, μ, c and the two probabilities r and q of Markov model over the authentication probability $P_r\{P_i\}$ of our scheme using 2-state Markov model. The values of these parameters are chosen as follow:

- $1 \leq \nu \leq 3$, the number of packets containing the hash of a packet P_i,
- $2 \leq \mu \leq 4$, the number of hashes appended to a signature packet,
- $2 \leq c \leq 32$, the number of chains in MC model,
- $0.1 \leq q \leq 0.9$, the transition probability from loss to receive,
- $r = 0.001, 0.01, 0.1$, the transition probability from receive to loss.

Figure 8 depicts the authentication probability in terms of μ for different values of ν when $c = 8$, $k = 5$, $r = 0.1$ and $q = 0.8$, where the authentication probability increases when μ increases for all values of ν. The more hashes are appended to the signature packet, the higher the authentication probability is achieved. Figure 8 also shows that higher authentication probability is achieved when more packets contain the hash of P_i.

The authentication probability increases as the transition probability q increases for all values of μ as Figure 9 depicts. Also the authentication probability decreases as the transition probability

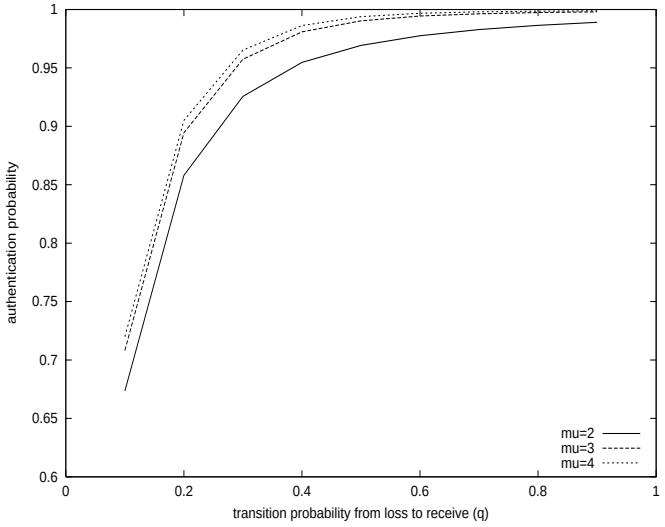

Figure 9: Authentication probability using 2-state Markov model in terms of q, when $\nu = 2$, $c = 8$, $k = 5$ and $r = 0.1$.

r increases for all values of μ as Figure 10 shows. Increasing the transition probability q achieves higher authentication probability since more packets are received. The transition probability r is opposite to q; that is, increasing r achieves lower authentication probability since more packets are lost.

Figure 11 depicts the authentication probability in terms of number of chains c. The greater the value of c, the greater μ can be chosen, this in turn will increase the authentication probability. Figure 11 is depicted using larger values for μ as c increases; that is, when $c = 4$, the value of $\mu = 2$ while when $c = 8$, the value of $\mu = 4$ and so on.

5.3 Loss Probability using 2-state Markov Model

In this section we derive the loss probability of $E(\mu)$ using 2-state Markov model. The loss probability here means the probability that all the whole packets of $E(\mu)$ are lost. Let ρ_1 be the loss probability of the packets of $E(\mu)$ that belong to any block of MC. According to 2-state Markov model there are $(j_2 - j_1)$ transition states from P_{j_1} to P_{j_2} and when these two packets are lost, its transition probability is denoted $p_{11}^{(j_2-j_1)}$, while there are $(j_3 - j_2)$ transition states from P_{j_2} to $P_{j_2}, \cdots$, $(j_\mu - j_{\mu-1})$ transition states from $P_{j_{\mu-1}}$ to P_{j_μ}, so ρ_1 is given as:

$$\rho_1 = \prod_{i=1}^{\mu-1} (p_{11}^{(j_{i+1}-j_i)}). \tag{16}$$

If the number of transition states between P_{j_i} and $P_{j_{i+1}}$ is increased, the loss probability ρ_1 of the packets of $E(\mu)$ becomes lower. That can be achieved by choosing P_{j_i} far from $P_{j_{i+1}}$, where $1 \leq i \leq \mu - 1$.

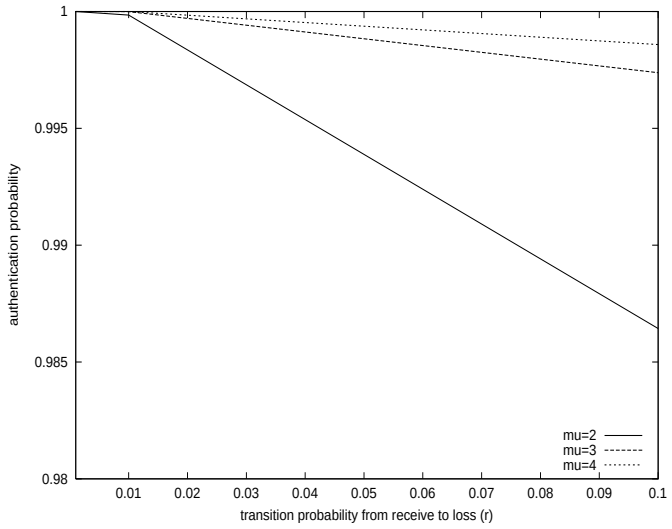

Figure 10: Authentication probability using 2-state Markov model in terms of r, when $\nu = 2$, $c = 8$, $k = 5$ and $q = 0.8$.

On the other hand, the loss probability of $E(\mu)$ against burst loss can also be discussed using 2-state Markov model, where the loss probability here means the probability that all the whole packets of $E(\mu)$ are lost as a consequence of burst loss. Let ρ_2 be the loss probability of $E(\mu)$ as a consequence of burst loss, where $E(\mu)$ belongs to any block of MC. The condition for the burst length b that results in loosing all the whole packets of $E(\mu)$ is given as $j_\mu - j_1 < b$. There is one transition state to the start of the burst of length b from the packet that preceeds it and its transition probability is denoted as p_{01}. There are $(b-1)$ transition states from the start of the burst of length b to its end and the transition probability is denoted as $p_{11}^{(b-1)}$. There is one transition state from the end of the burst of length b to the packet that follows and its transition probability is denoted as p_{10}. According to 2-state Markov model, ρ_2 is given as follows:

$$\rho_2 = r \cdot (1-q)^{b-1} \cdot q. \tag{17}$$

So as to reduce the loss probability ρ_2 of $E(\mu)$ as a consequence of burst loss of length b, the number of transition states between P_{j_1} and P_{j_μ} should be more than $(b-1)$. That can be achieved by choosing $j_\mu - j_1 \geq b$, which guarantees that at least one packet of $E(\mu)$ is received wherever the burst of length b occurs.

6 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,

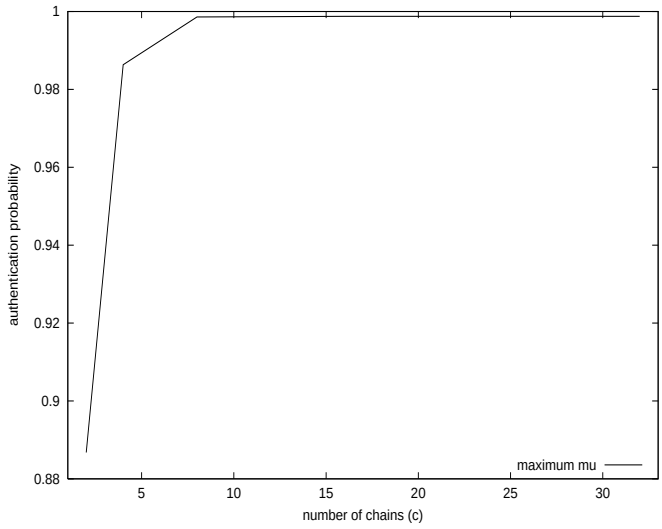

Figure 11: Authentication probability using 2-state Markov model in terms of c, when $\nu = 2$, $k = 5$ $r = 0.1$ and $q = 0.8$.

where the receivers usually do not buffer large amounts of unconsumed data. In this section we show the effect of MC model on the delays and buffers capacities for both the sender and the receivers.

6.1 Sender's Buffer and Delay

Since the signature packet is sent at the end of the block of MC model, the sender experiences a single packet delay. The sender does not need to keep the packets in the buffer, instead he keeps the hash values of some packets that are necessary to compute the hash values of succeeding packets and the signature packets. Accordingly, the requested buffer size is equal to the scope of a packet P_i in addition to μ hashes that are necessary to compute the signature packet. The scope of a packet P_i is defined as the maximum length from a packet P_i to the other packet P_j that contains $H(P_i)$, where $j > i$. In our model the hash of P_i is appended to $P_{i+(\nu-1)c}$ at most, so the scope of P_i is $(\nu - 1)c + 1$. Accordingly, the sender's buffer size α is given as:

$$\alpha = (\nu - 1)c + \mu + 1 \tag{18}$$

This equation shows that the sender's buffer capacity increases as c increases.

6.2 Receiver's Buffer and Delay

The necessary buffer size and delay for a receiver to authenticate the received packets depends on where a burst loss occurs, its length and the block size of the authentication scheme. If a burst loss includes a signature packet, the necessary buffer size and delay are increased.

Let b_i, $i = 1, 2, \cdots, n$, denote the length of the burst loss i, where n is the number of bursts. Also, let θ and α_1 denote the number of consecutive signatures loss and the number of packets a receiver needs to hold in the buffer. Then, α_1 is equal to receiver's delay.

If two consecutive signature packets $P_{sig_{j-1}}$ and P_{sig_j} are received, where $1 < j \leq \gamma$, the number of packets a receiver must wait for until he is able to authenticate the received packets is $\alpha_1 = kc - \sum_{i=1}^{n} b_i$. The reason is that there are kc packets preceding a signature one and the total number of lost packets is subtracted.

When P_{sig_j} is received, provided that all the signatures $(P_{sig_{j-\theta}}, \cdots, P_{sig_{j-1}})$ are lost, the delay that a receiver must wait for is equal to his buffer capacity and is given as:

$$\alpha_1 = (\theta + 1)ck - \sum_{i=1}^{n} b_i \tag{19}$$

Equation (19) holds whether the receivers have different packet losses and different buffer capacities or the same. Also it shows that α_1 increases linearly as ck and θ increase.

7 Evaluation

In this section we apply our scheme to the two case studies proposed by the authors of the EMSS [Perrig, et al., 2000] and compare our scheme in terms of hash chain construction, loss resistance, authentication probability and sender's delay with previously proposed schemes, AC [Golle et al., 2001] and Signature Amortization using Information Dispersal Algorithm (SAIDA) [Park et al., 2003].

7.1 Case Studies

The first case study proposed by the authors of EMSS is as follows:

A municipality broadcasts data over the Internet of its traffic sensors that are distributed over streets. The requirements of such a system are:

- The data rate of the stream is about 8 Kbps, about 20 packets of 64 bytes each are set every second.
- The packet drop rate is at most 5%, where the average length of burst drops is 5 packets.
- The verification delay should be less than 10 seconds.

To construct MC model, we choose $\nu = 2$, so that at least one packet of $A(c, \nu)$ arrives with probability equal to $1 - 0.05^2 = 0.9975$. The value of μ is also chosen as 2, while the number of chains is chosen as $c = 20$ and the number of slices $k = 10$. Being able to determine these

parameters we can measure the overhead per packet δ from Equation (5) and the authentication probability $P_r\{P_i\}$ from Equation (14). Considering the number of packets in a stream $N = 10000$, the hash type is $SHA-512$ and sending the signature of type $RSA-1024$ twice in each packet, then $\delta \cong 129.15$ and $P_r\{P_i\} = 0.9971$. Using the same hash value the EMSS is using in their case studies, which is 10 bytes, then $\delta \cong 21.26$. While the overhead of EMSS in the first case study is 22 bytes and verification probability is 98.7%. Using the same hash value that EMSS used in this case study, our scheme achieves higher authentication probability with less overhead.

The second case study is a real-time video broadcasting with the following requirements:

- The data rate of the stream is about 2 Mbps, about 512 packets of 512 bytes each are set every second.
- Some receivers experience packet drop rates up to 60%, where the average length of burst drops is 10 packets.
- The verification delay should be less than 1 seconds.

Choosing $\nu = 3$, then at least one packet of $A(c, \nu)$ arrives with probability equal to $1 - 0.6^3 = 0.784$. The value of μ is chosen as 4, while the number of chains remains $c = 20$ and the number of slices $k = 10$. Considering the number of packets in a stream $N = 10000$, the hash type is $SHA-512$ and sending the signature of type $RSA-1024$ twice in each block, then $\delta \cong 192.89$ and $P_r\{P_i\} = 0.9763$. Using the same hash value the EMSS is using in their case studies, which is 10 bytes, then $\delta \cong 31.22$. While the overhead of EMSS in the second case study is 55 bytes and verification probability is 97%. When the loss rate and burst loss increase and using the same hash value that EMSS uses in this case study, our scheme achieves higher authentication probability and lower overhead.

7.2 Hash Chain Construction

The EMSS and AC schemes do not specify how many hashes nor what packets should have hashes appended to the signature packet. EMSS determines the best case of the chain to achieve strong resistance against packet loss by simulation.

AC does not give a clear method to determine the number of packets to be inserted between every two packets of the original chain.

Instead of appending the hash values of each packet to other packets, SAIDA encodes the hash values and appends the n encoded segments to n packets of a block. A receiver needs at least m packets among the n sent packets to be able to reconstruct the authentication data for the whole

received packets of a block. Although the communication overhead is reduced, the computation resources are high. SAIDA uses the same technique to encode the signature packets.

Our model consists of multiple connected chains and clearly specifies ν, the number of packets that contain the hash of p_i in terms of number of chains c. Also our solution specifies what packets should have hashes μ appended to a signature packet and how to choose them. The value of the number of chains c is determined in terms of expected burst length. Being able to determine the values of the parameters (ν, μ, c) and construct the hash chain achieve an efficient performance.

7.3 Loss Resistance

Loss resistance achieved by an 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 an augmented chain $C_{a,p} = C_{3,6}$, the loss resistance is equal 12 packets, where p packets are inserted between every two packets of the original chain C_a. 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.

SAIDA achieves loss resistance equal to $n-m$, where n is the number of packets in a block and m is the minimum needed number of packets to reconstruct the authentication information of the received packets. SAIDA achieves stronger loss resistance by adjusting n and m keeping the space overhead fixed, but computation resources are still high.

Our scheme on the other hand, achieves loss resistance equal to $\ell = (\nu-1)c-1$ 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. Stronger resistance against loss is achieved by increasing c, which reduces the overhead. The major advantage of our scheme over those previously proposed is achieving stronger resistance against loss and reducing the overhead in the same time.

7.4 Authentication Probability

A recurrence authentication probability formula for EMSS and AC had been derived in case of independent packet loss. The authentication probability of EMSS and AC depend on two factors, the number of hashes appended to a signature packet and the number of packets contain the

hash of a packet P_i. Increasing these two factors achieves higher authentication probability, which increases the overhead.

The authentication probability of SAIDA depends on the block size. Increasing the block size achieves higher authentication probability.

The authentication probability of our scheme had been derived using 2-state Markov model and it depends on ν, μ and c. Increasing ν and μ achieve higher authentication probability, also increasing c makes it possible to choose more values of μ, which in turn achieves higher authentication probability and reduces the overhead in the same time.

7.5 Sender's Delay

The delay on a sender using EMSS scheme is equal to one packet since the signature packet is the last one of a block and depends on previously computed hashes. The sender experiences a delay equal to p packets using AC scheme, where p represents the sender buffer size in AC scheme. Using SAIDA, the sender experiences a delay equal to n packets, where n represents the number of packets in a block. Our scheme, on the other hand, using MC model signs the last packet of a block. Therefore, the sender experiences a delay of a single packet.

8 Related Works

Digital signature achieves non-repudiation and authenticity for the received messages. Relying on digital signature, several schemes have been introduced to authenticate multicast streams. The schemes that sign each packet separately are impractical due to high computation and communication overhead and delay on both sender and receivers [Rohatgi, 1999], even if faster signing algorithms are used [Wong, 1999].

The alternative is to use Message Authentication Codes (MAC), such as TESLA introduced by [Perrig et al., 2000]. TESLA requires time synchronization between the sender and the receivers, which might not be feasible in large multicast groups.

Another alternative schemes is to divide the stream into many blocks, sign a single packet in each block and link the rest of the packets in the block to the signed one using multiple hashes links, such as Authentication Tree [Wong, 1999], EMSS [Perrig et al., 2000] and AC [Golle et al., 2001] schemes. These schemes are known as signature amortization, which its security is proved by [Gennaro et al., 1997].

The Authentication Tree computes the hash of each packet in a block to form the leaf nodes of a tree. The parent nodes are computed as the hashes of their children. A signature is computed as the root. Since each packet contains a signature with the authentication information so as to

be individually verifiable, that requires high amount of overhead.

EMSS overcomes weak loss resistance of the schemes introduced by the authors of [Gennaro et al., 1997], by storing the hash of each packet in multiple locations and appending multiple hashes to the signature packet. This method according to [Chan, 2003] and [Miner et al., 2001], increases the overhead. The EMSS determines the block size and the number of hashes to append to each packet by experiments. It also chooses the location of these hashes randomly.

The AC scheme was introduced to achieve stronger burst loss resistance by using a similar strategy to that of the EMSS, but the locations of the appended hashes are deterministic. The AC does not include a means of choosing the number of packets to be inserted between each pair of the original chain. More details about the AC analysis is found in [Alain et al., 2002], where it is applied to two case studies and compared to the EMSS.

The authors of [Chan, 2003] and [Miner et al., 2001] give an analysis of hash chains 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.

In Piggybacking scheme [Miner et al., 2001], a group of n packets are partitioned into r equal-sized subgroups called classes S. These classes have different priorities. The first packet in the highest priority class is signed. The main aim of Piggybacking is to resists multiple bursts x_i of size at most b.

To reduce the overhead of amortization schemes, SAIDA [Park et al., 2003] uses erasure codes to achieve that goal. The computation resources, on the receivers, of the Forward Error Correction (FEC) in SAIDA is high comparing to that of hashes in other amortization schemes [Cucinotta et al., 2003]. The fast computation and communication cost of the hashes [Wong, 1999], [Perrig et al., 2003], [Stallings, 2003] makes amortization schemes widely adopted.

9 Conclusion

We introduced a multiple connected chains MC model for signature amortization to authenticate multicast streams and showed how to determine the values of the parameters of our scheme that influence the performance of the authentication scheme. We also showed how to measure the efficiency metrics such as the authentication probability, the loss resistance, the overhead and the sender and the receivers buffer capacities and delays. The loss probabilities of the packets that have their hashes appended to the signature packet have been introduced. Being able to determine the values of the parameters made it possible to construct the best construction of our model in advance that achieves the desired performance. Our scheme achieves higher authentication probability by increasing the number of appended hashes to other packets and to the signature packet. Our

scheme achieves stronger loss resistance against burst packet loss and reduces the overhead in the same time by increasing the number of chains of MC model.

As future works, we will discuss the optimal values of ν, μ, c and k. More derivation and analysis of the authentication probability of our scheme still necessary. It is also interesting to achieve empirical study to compare the theoretical results to the experimental ones. Compare the performance of our scheme with that of other schemes is our next research attempt.

References

Abuein, Q. and Shibusawa, S., (2004a) Efficient multicast authentication scheme using signature amortization. Proc. of the IASTED Int. Conf. on CIIT, pp.133-140.

Abuein, Q. and Shibusawa, S., (2004b) New chain construction for multicast stream authentication. Proc. of the ICENCO Int. Conf. on NTIS, pp.174-179.

Abuein, Q. and Shibusawa, S., (2005a) The performance of amortization scheme for secure multicast streaming. Proc. of the 6th Int. Workshop on Information Security Application, pp.169-184.

Abuein, Q. and Shibusawa, S., (2005b) Signature amortization using multiple connected chains. Proc. of 9th IFIP TC-6 TC-11 Int. Conf. on CMS, pp.65-76.

Alain, P. and Refik, M., (2002) Authenticating real time packet stream and multicast. Proc. of 7th IEEE Symposium on Computers and Communications.

Chan, A., (2003) A graph-theoretical analysis of multicast authentication. Proc. of the 23rd Int. Conf. on Distributed Computing Systems.

Christophe, T. and Huaxiong, W., (2005) Efficient multicast stream authentication for the fully adversarial network model. Proc. of the 6th Int. Workshop on Information Security Application.

Cucinotta, T., Cecchetti, G. and Ferraro, G., (2003) Adopting redundancy techniques for multicast stream authentication. Proc. of the $9th$ IEEE Workshop on FTDCS.

Gennaro, R. and Rohatgi, P., (1997) How to sign digital streams. Advances in Cryptology - CRYPTO'97, pp.180-197.

Golle, P. and Modadugu, N., (2001) Authenticating streamed data in the presence of random packet loss. Proc. of ISOC Network and Distributed System Security Symposium, pp.13-22.

Jiang, W. and Schulzrinne, H., (2000) 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.

Lysyanskaya, A., Tamassia, R., and Triandopoulos, N., (2004) Multicast authentication in fully adversarial networks. Proc. of IEEE Symposium on Security and Privacy, pp.241-255.

Miner, S. and Staddon, J., (2001) Graph-based authentication of digital streams. Proc. of the IEEE Symposium on Research in Security and Privacy, pp.232-246.

Park, J., Chong, E. and Siegel, H., (2003) Efficient multicast stream authentication using erasure codes. ACM Trans. on Information and System Security, vol.6, nr. 2, pp.258-258.

Perrig, A., Canetti, R., Tygar, J. D., and Song, D., (2000) Efficient authentication and signing of multicast streams over lossy channels. IEEE Symposium on Security and Privacy, pp.56-73.

Perrig, A. and Tygar, J. D., (2003) Secure broadcast communication in wired and wireless networks. Kluwer Academic Publishers, Norwell, MA.

Rohatgi, P., (1999) A compact and fast hybrid signature scheme for multicast packet authentication. Proc. of the 6th ACM Conf. on Computer and Communications Security.

Sanneck, H., Carle, G., and Koodli, R., (2000) A framework model for packet loss metrics based on loss runlengths. SPIE/ACM SIGMM Multimedia Computing and Networking Conf.

Stallings, W., (2003) Cryptography and network security principles and practices, Prentice Hall, Upper Saddle River, NJ.

Wong, C. K. and Lam, S. S., (1999) Digital signatures for flows and multicasts. IEEE/ACM Trans. on Networking, vol.7, nr. 4, pp.502-513.

Yajnik, M., Kurose, J., and Towsley, D., (1996) Packet loss correlation in the mbone multicast network. Proc. of IEEE Global Internet.