2B-4

New Hash Chain for Signature Amortization Schemes

Qusai Abuein Susumu Shibusawa *
Ibaraki University

1 Introduction

Recently, authenticating multicast streams in real-time environment using signature amortization has a great concern. More studies on amortization schemes are still necessary such as where to place the signature packet, how to determine the packets that its hashes to be appended to the signature one, how many hashes to append to the signature packets, in addition to hash chain analysis to show its effect on overhead, loss resistance and authentication probability. How to lengthen the path between a packet and the signature one, so as to increase loss resistance is another research point.

We introduced a solution of the mentioned issues in [1]. The solution used basically three hashes to append to the signature packet. In this paper we generalize that number and introduce equations to determine the appropriate number so as to achieve the desired results. Accordingly, the equations to compute the overhead and the loss probabilities are derived. We also study the relation between the number of hashes appended to the signature packet and the overhead.

2 Chain Construction

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. Our model consists of multiple c chains, each packet P_i is connected to P_{i+1}, P_{i+c}, P_{i+2c} so as to increase robustness to packet loss. Connecting P_i to another packet means concatenating the hash of P_i with the data of the other packet before computing the hash of the other packet as follows: 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 || represents concatenation and → represents the hash of the packet is obtained from the computation of the left value. The signature packet P_{sig_j}, where $j \geq 1$, is appended with some hashes of non-contiguous packets denoted as a chosen from the last c packets preceding the signature one 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 signature packet is 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.

Figure 1 depicts a construction of our multiple connected chains (MC) 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.

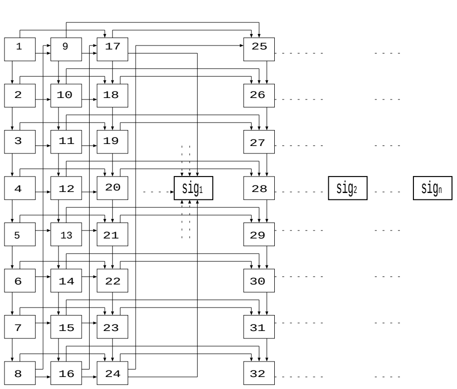

Figure 1: MC model: $c = 8, p = 3c$.

3 Theoretical Analysis

According to our MC model, each packet of the first c packets except the first one, contains only a single hash, that is, in total there are $c-1$ hashes. While each packet of the second c packets contains 2 hashes of the previous packets, in total there are $2c$ hashes. Each packet of the rest of the packets contains 3 hashes. Accordingly the number of hashes β in the stream is computed as follows:

$$\beta = 3c + 3(N-2c) - 1 = 3(N-c) - 1 \qquad (1)$$

where N represents the total number of packets in the stream. When h represents the hash size, the total size of all hashes H in the stream is as follows:

$$H = h\beta \qquad (2)$$

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

$$\gamma = \lceil \frac{N}{p} \rceil \qquad (3)$$

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 and computed as follows:

$$\delta = \frac{H + \gamma(s + ah)}{N} \qquad (4)$$

Since a packets are chosen non-contiguously from the last c packets preceding the signature one in our model, the value of a is chosen as follows:

$$a \leq \lceil \frac{c}{2} \rceil \qquad (5)$$

*Department of Computer and Information Sciences, Ibaraki University, Hitachi, Ibaraki 316-8511, Japan

Loss resistance ℓ is the maximum number of lost packets the scheme can resist so as to be able to authenticate the received 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:

$$\ell = 2c - 1 \tag{6}$$

Since the number of chains of MC plays a main role in its efficiency we introduce a measure regarding burst packet loss length b and the loss resistance ℓ. The model must resist the expected burst loss b, accordingly:

$$c \geq \lceil \frac{b+1}{2} \rceil \tag{7}$$

According to Gilbert model, the loss probability ρ_1 for the a packets appended to the signature one within the range $\{P_{(k-1)c}, P_{(k-1)c+1}, \ldots, P_{kc}, P_{sig_1}\}$ as non-contiguous is as follows:

$$\rho_1 = (1-r)^{c-2a+1} \cdot r^a \cdot q^a \tag{8}$$

where 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 is opposite to r.

While the loss probability ρ_2 for a in case of contiguity:

$$\rho_2 = (1-r)^{c-a} \cdot r \cdot (1-q)^{a-1} \cdot q \tag{9}$$

4 Simulation Results

The hash chain construction in our model mainly depends on the number of the chains c. The effect of c is shown in Figure 2. While the effect of a on the overhead is depicted in Figure 3.

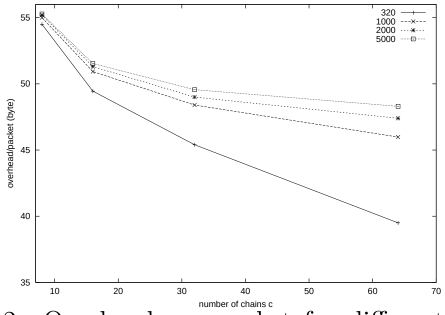

Figure 2: Overhead per packet for different streams where p is after $3c$ and $a = 3$.

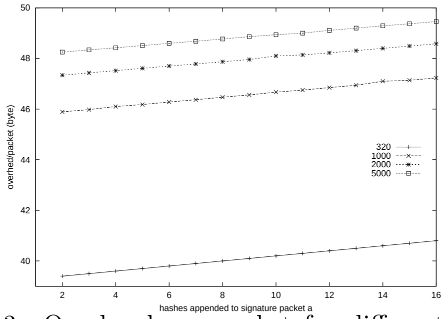

Figure 3: Overhead per packet for different streams regarding a, where $c = 16$ and p is after $3c$.

5 Performance Evaluation

We compare our solution to two previously proposed schemes, EMSS [2] and Augmented Chain (AC)[3], as shown in Table 1.

Table 1: Comparison of the Authentication Schemes

	Hash chain	Loss resistance
MC	Specifies clearly that number in addition to its theory	Longer loss resistance reduces overhead
EMSS	Specifies the number of hashes appended to other packets by simulation only	Longer loss resistance increases overhead
AC	Does not specify the number of packets to be merged in the original one	Longer loss resistance increases overhead

Our model achieves similar performance to EMSS and AC concerning the following criteria: sender delay, receiver delay, computation and communication overhead, and verification rate.

6 Conclusion

We introduced a measure to determine the number of hashes to be appended to the signature one. The loss probability of the packets that its hashes are appended to the signature one in case of non-contiguity and contiguity were also introduced. The effect of the number of hashes that are appended to the signature packet on the overhead for different streams is also studied.

Our scheme achieves longer loss resistance and lower overhead by increasing the number of chains. Mathematical equations to determine the appropriate number of chains are introduced.

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] Q. Abuein and S. Shibusawa, “Efficient multicast authentication scheme using signature amortization,” Proc. of the IASTED Int. Conf. on CIIT, Nov. 2004.

[2] A. Perrig et al, “Efficient authentication and signing of multicast streams over lossy channels,” IEEE Symp. on Security and Privacy, pp.56-73, May 2000.

[3] P. Golle and N. Modadugu. “Authenticating streamed data in the presence of random packet loss,” Proc. of ISOC Network and Distributed System Security Symp., pp.13 – 22, 2001.