A PROPOSAL OF BUTTERFLY-GRAPH BASED STREAM AUTHENTICATION OVER LOSSY NETWORKS

Zhishou Zhang[1,2], Qibin Sun[1], Wai-Choong Wong[1,2]
[1]*Institute for Infocomm Research, Singapore*
[2]*Department of ECE, National University of Singapore*
{zszhang, qibin, lwong}@i2r.a-star.edu.sg

Abstract

In this paper, we propose a butterfly-graph based stream authentication scheme for lossy networks where the streaming packets could be lost in both random and burst ways. Due to the nice properties of butterfly graph, the proposed scheme is quite robust and efficient. Theoretical analysis and simulation results show that the proposed scheme outperforms existing schemes in terms of overhead and authentication probability while maintaining the same levels of sender / receiver delay and robustness.

1. Introduction

Nowadays, as more and more applications require data/media streaming, it is very important to protect the authenticity of the streams in the aspects of integrity and non-repudiation. Digital signature is a natural solution for addressing such issues. However, directly applying the digital signature for the streams like video is impractical: one issue is the high communication overhead, as each packet will be appended with a signature; the other issue is the high computation overhead due to the complexity of signature generation and verification. Therefore, to design a satisfactory authentication scheme for digital streams, we may need to carefully balance the following requirements:

Computation overhead: It refers to the computation resources required to generate the signature at the sender site and to verify the signature at the receiver site. As the digital stream typically has a huge amount of continuous data, this requirement becomes even more critical when the streaming involves mobile devices with low capabilities such as cellular phone.

Communication overhead: It refers to the additional bytes to be transmitted along with the stream packets. These additional bytes include MAC (i.e., Message Authentication Code or Crypto Hash) values or digital signatures. It is also critical especially in wireless environments where the channel bandwidth is scarce.

Sender delay: It refers to the delay from the time when the packet is first processed to the time when it is actually sent out of the sender. In real-time streaming scenarios, a high sender delay often requires a large buffer at the sender.

Receiver delay: It refers to the delay from the time the packet is received to the time when it is authenticated by the receiver. Likewise, a high receiver delay often requires a large buffer at the receiver. When consuming a streamed media, usually each packet has its deadline after which it becomes useless, thus, a large receiver delay could cause a packet to miss out its deadline.

Robustness against packet loss: The packets of the stream should be able to be authenticated with high probability even if the stream is sent over lossy networks. This requirement is particularly useful for video/audio streams which can tolerate some packet loss (random and burst). The authentication probability is defined as the probability that a delivered packet can be successfully authenticated.

Obviously it is hard to meet all above-mentioned requirements, as some requirements conflict with each other. For instance, usually the sender delay conflicts with the receiver delay and the overhead conflicts with robustness. Therefore, the design of the stream authentication scheme is application-dependent.

The problem has been attempted mainly using two approaches: one is to generate a lightweight signature; the other is to amortize one signature over a group of packets. The first approach employs One-Time-Signature (OTS) [1] and extension to the Feige-Fiat-Shamir signature (eFFS) [2], which reduces the computation overhead at the expense of increased communication overhead (Huge size of the signature / key). The second approach can be further classified into graph based schemes [1,2,4,5,6,7] and erasure code based schemes [8]. Gennaro *et al* [1] proposed an authentication scheme using a simple hash chain. It has low overhead and low receiver delay, but it also has a

0-7803-9332-5/05/$20.00 ©2005 IEEE

high sender delay and cannot tolerate any packet loss. Wong *et al*'s [2] scheme is based on the Merkle authentication tree [3]. This scheme has very high communication overhead, although it can tolerate large packet losses. Perrig *et al* [4] proposed two schemes: TESLA and EMSS. The TESLA scheme relies on the loose-time synchronization between the sender and the receiver, which is sometimes hard to achieve. The EMSS scheme uses a hash chain, where each packet contains the hashes of previous packets and the signing is on the last packet. This scheme has a high receiver delay and a low sender delay. Golle and Modadugu's [5] scheme is based on augmented chain. Since the signing is still on the last packet, it also has high receiver delay. Song *et al* [6] proposed an authentication scheme based on the expander graph and further theoretically derived the lower bound of their authentication probability. However, it has a very large communication overhead which is unacceptable for a real application. Miner and Staddon's [7] scheme is based on the random graph. The signing is on the first packet, and each packet contains the hashes of every subsequent packet with certain probability. Therefore, it also has high communication overhead. Park *et al* [8] proposed to use erasure code for stream authentication. For each block, the digital signature is coded with erasure code and is then scattered into the packets. As long as the number of loss packets is less than a threshold, all received packets can be authenticated. This scheme has a high computation overhead due to the erasure coding. In addition, it also suffers from a high receiver delay, because the receiver has to wait for a minimum number of the received packets before authentication.

This paper proposes a new stream authentication scheme based on the butterfly graph, where one signature is amortized among a group of packets connected with a butterfly graph. Owing to the nice properties of the butterfly graph, our scheme has lower overhead and higher authentication probability, while maintaining the same level of delays and robustness against packet loss.

The paper is organized as follows. Section 2 describes the butterfly-graph based authentication scheme. Section 3 compares the proposed scheme with the existing schemes. Section 4 concludes this paper.

2. Proposed Butterfly-graph based scheme

Assume the stream is divided into a number of blocks and each block contains M packets, where only one signature is generated for each block, and the M packets and the signature packet are connected using the butterfly graph. Assuming $M = N(\log_2 N+1)$, the definition of the graph is given below.

Definition: *A butterfly authentication graph is a directed acyclic graph (DAG) containing one signature packet* ***S*** *and* $M = N(\log_2 N+1)$ *data packets. The M data packets are divided into* $(\log_2 N+1)$ *stages, and each stage has N packets. The packet is denoted as* $P(s,j)$*, where* $s \in \{0,1,\ldots, \log_2 N\}$ *indicates the stage and* $j \in \{0,1,\ldots, N-1\}$ *indicates the packet in a stage. In this graph, there exists a directed edge* $\vec{e}(P(s_1, j_1), P(s_2, j_2))$ *from packet* $P(s_1,j_1)$ *to packet* $P(s_2,j_2)$ *if either of the following conditions is met:*

1. $s_1 = s_2 + 1$ *and* $j_1 = j_2$
2. $s_1 = s_2 + 1$ *and* $j_1 = \bar{j}_2^{s_2}$, *where* $\bar{j}_2^{s_2}$ *is different from* j_2 *only at the bit position* s_2.

In addition, there also exists a directed edge from all packets in stage 0 to the signature packet ***S****.*

In the butterfly authentication graph, each directed edge $\vec{e}(P(s_1, j_1), P(s_2, j_2))$ is realized by appending the hash of the packet $P(s_1,j_1)$ to the packet $P(s_2,j_2)$. Fig. 1 gives an example of the butterfly authentication graph, with 4 stages and 8 data packets in each stage. The signature packet ***S*** contains the signature and hashes of all packets in stage 0. All packets in stage 0 to $\log_2 N$-1 have two hashes, and the packets in the last stage do not have any hash. Assuming each hash has h bytes, each signature has g bytes, and each block has $M = N(\log_2 N+1)$ packets, the communication overhead per packet on average O_{avg} is:

$$O_{avg} = 2h - \frac{h}{\log_2 N + 1} + \frac{g}{M} \qquad (1)$$

We can also see that the computation overhead is quite low, including 1 signing operation for the whole block and 1 hashing operation for each packet.

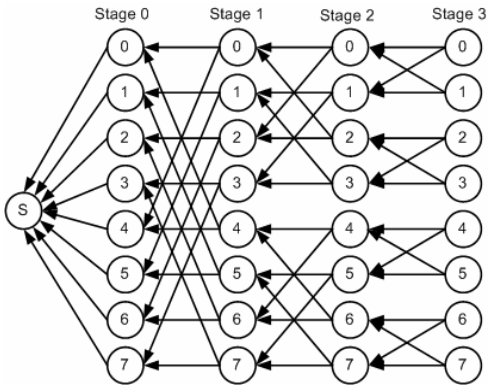

Fig. 1 – An example of butterfly authentication graph

Before we analyze the authentication probability, we assume that the signature packet ***S*** is always

received, and all data packets have equal and independent loss probability ε (i.e., random loss). A packet $P(s,j)$ cannot be authenticated unless there is a path to the signature packet $\boldsymbol{S}$ at the receiver. The authentication probability $\phi(P(s,j))$ is equivalent to probability that such path exists, as shown in Eq. (2).

$$\phi(P(s,j)) = \left(1-\varepsilon^2\right)^s \qquad (2)$$

We can see that $\phi(P(s,j))$ depends only on s and ε, and all packets in the same stage have the same $\phi(P(s,j))$. As we travel from stage 0 to stage $\log_2 N$, the authentication probability decreases, because a packet in the later stage has more dependency than that in the earlier stage. However, this trend is slowed down by the butterfly graph where a packet in the later stage has more paths to the signature packet $\boldsymbol{S}$. For instance, a packet in stage 0 has only one path to $\boldsymbol{S}$, while a packet in the last stage has N paths to $\boldsymbol{S}$. The minimum authentication probability $\phi_{\min}$ under random packet loss can be derived according to Eq. (3).

$$\phi_{\min} = \left(1-\varepsilon^2\right)^{\log_2 N} \qquad (3)$$

Regarding the burst packet loss, our scheme is able to resist up to $N/2^{s+1}$ consecutive packet losses at stage s, where $0 \le s < \log_2 N$. In the example graph depicted in Fig. 1, the maximum length of burst loss is 4 and 2 in stages 0 and 1, respectively.

Note that our butterfly-graph based authentication scheme also has a high sender delay M, because the sender has to compute the hashes and the signature of the block before sending the first packet. Such delay can be further reduced to 1 by pre-processing the packets (e.g., pre-compute hashes and signature) before the streaming starts. However, our scheme does not have any receiver delay, i.e., the received packet can be authenticated immediately. As mentioned before, low receiver delay means small buffer space at the receiver, which is required in most mobile applications.

3. Comparison with existing schemes

In this section, we compare the proposed scheme with other existing schemes. Table 1 summarizes the 8 authentication schemes based on the aforementioned evaluation criteria. The results in Table 1 are obtained by the following reasonable settings / assumptions:

- The block size is M, the hash is h bytes and the signature is g bytes. The parameter d denotes the distance between the current packet to the signature packet.
- The degree of Merkle's authentication tree used in scheme [2] is set to 2.
- In the scheme [6], every two consecutive levels form a $(n/a,n)$-bipartite expander graph of degree(da,d) which is $(ad/8,d/8a)$-expanding.
- In the scheme [7], a packet contains the hashes of every subsequent packet with equal probability ρ.
- For the scheme [4], each packet has n hashes, and maximum edge length is a.
- For the scheme [5], a is the maximum edge length and p is the size of packet buffer at the sender.
- For the scheme [8], m is the minimum number of received packets to recover the hashes and the signature in a block.
- For our scheme, we assume $M = N(\log_2 N+1)$. The parameter s refers to the stage number.

Table 1. Comparison with existing schemes

	Comp. Overhead	Sender delay	Receiver delay	Max. burst loss
Tree-Chain [2]	*(2M-1), 1*	*M*	*1*	*any*
Simple Hash Chain [1]	*M, 1*	*M*	*1*	*0*
Expander Graph [6]	*M, 1*	*M*	*1*	*Unconsidered*
Random Graph [7]	*M, 1*	*M*	*1*	*Unconsidered*
EMSS [4]	*M, 1*	*1*	*M*	*a-1*
Augmented Chain [5]	*M, 1*	*p*	*M*	$p\times(a-1)$
Erasure Code [8]	*M,1,2*	*M*	*[m, M]*	*M-m*
Butterfly-graph	*M, 1*	*M*	*1*	$N/2^{s+1}$

In terms of computation overhead, the Tree-Chain scheme performs M-1 more hash operations than the rest, and the Erasure Code scheme perform 2 additional erasure coding operations (Table 1). All other schemes including ours have roughly the same computation overhead.

In terms of sender and receiver delay, the Erasure Code scheme and the Augmented Chain scheme have the lowest performance. Note that the sender and receiver delays are closely correlated in the graph-based schemes, for instance, signing the first packet incurs a high sender delay and low receiver delay, while signing the last packet incurs a low sender delay and high receiver delay (Table 1).

In terms of the robustness against burst loss, the Tree-Chain scheme has the best performance. Our butterfly-graph scheme is able to resist up to $N/2^{s+1}$ consecutive packet losses at stage s.

The Tree-Chain, Expander Graph and Random Graph scheme have much higher communication

overhead than the others. In particular, the Expander Graph has unacceptable overhead, for instance, to achieve the lower bound of authentication probability of 69%, the following conditions must be satisfied: $d \geq 5120$ and $a \geq 36$, where $d \times a$ is the number of hashes per packet.

In terms of authentication probability, the Tree-Chain scheme is most robust, as the received packets can be authenticated with probability 1. In most schemes, the authentication probability and communication overhead conflict with each other, that is, increasing the overhead will increase the authentication probability, and vice versa. Fig. 2 shows the authentication probabilities under different communication overheads, assuming equal and independent loss probability $\varepsilon=0.3$, $M=1024$, $h=16$ and $g=128$. For EMSS scheme, the length of each edge is uniformly distributed in the interval [1,128]; For the Augment Chain Scheme, $a=15$ and $p=7$. Fig. 2 shows that our butterfly-graph based scheme outperforms all other schemes except the Erasure Code scheme in terms of overhead and authentication probability.

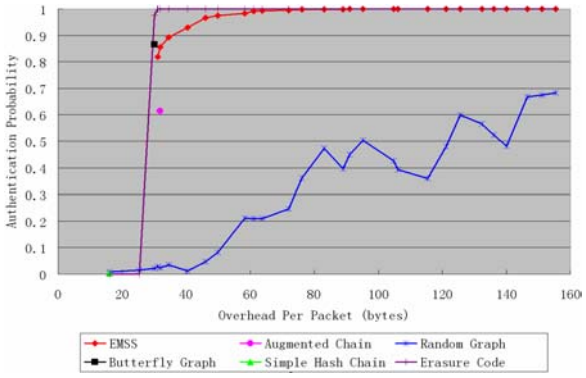

Fig. 2. Authentication probabilities at different overheads. (Loss probability is fixed at 30%)

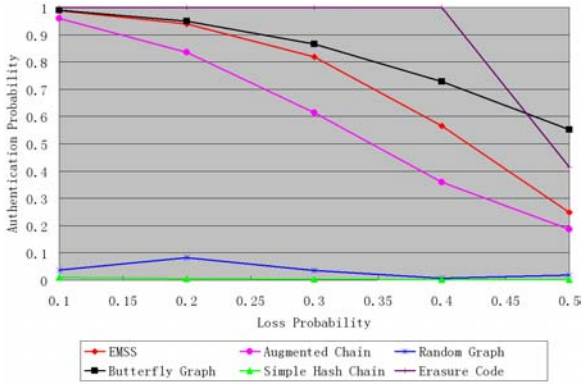

Fig. 3. Authentication probabilities at different loss probabilities. (The overhead is 32 bytes per packet)

If we only check from Fig. 2, the performance gap is not significant between the Butterfly-graph based scheme and the EMSS scheme. However, with the increase of loss probability, the butterfly-graph based scheme performs much better than the EMSS scheme, as shown in Fig. 3. We can see that this performance gap increases with the loss probability ε, at high loss probability.

4. Conclusions

In this paper, we proposed a butterfly-graph based authentication scheme, which aims to achieve low overheads and high authentication probability. The scheme is robust against both random and burst packet losses. By analysing its performance and comparing with other existing schemes we have shown that the butterfly-graph based scheme outperforms existing schemes in terms of overheads, authentication probability or receiver delay.

References

[1] R. Gennaro and P. Rohatgi. “How to Sign Digital Streams”, In Advances in Cryptology--CRYPTO '97.

[2] C. K. Wong and S. Lam, "Digital Signatures for Flows and Multicasts", The University of Texas at Austin, Dept. of Comp. Sc., Tech. Report TR-98-15. July 1998

[3] R. Merkle, “A Certified Digital Signature”, In Advances in Cryptology - CRYPTO'89, pp 218-238, 1990. Springer-Verlag, LNCS 435.

[4] A. Perrig, R. Canetti, J. Tygar and D. Song, “Efficient Authentication and Signing of Multicast Streams over Lossy Channels”, Proc. of IEEE Symposium on Security and Privacy, 2000.

[5] P. Golle and N. Modadugu, “Authenticating Streamed Data in the Presence of Random Packet Loss”, ISOC Network and Distributed System Security Symposium (2001), pp13--22.

[6] D. Song, D. Zuckerman, and J. D. Tygar, “Expander Graphs for Digital Stream Authentication and Robust Overlay Networks”, Proc. of IEEE Symposium on Research in Security and Privacy, pp 258-270, May 2002

[7] S. Miner and J. Staddon, “Graph-based Authentication of Digital Streams”, Proc. of IEEE Symposium on Research in Security and Privacy, pp 232--246, 2001.

[8] J. M. Park, E. K. P. Chong, and H. J. Siegel, "Efficient Multicast Stream Authentication using Erasure Codes," ACM Trans. Inf. Syst. Secur., vol. 6, no. 2, 2003

[9] M. S. Borella, D. Swider, S. Uludag and G. Brewster, "Internet Packet Loss: Measurement and Implications for End-to-End QoS", Proc. of International Conference on Parallel Processing, Aug. 1998.