Regular Paper

A Real-Time Stream Authentication Scheme for Video Streams

Shintaro Ueda,† Shin-ichiro Kaneko,† Nobutaka Kawaguchi,†
Hiroshi Shigeno† and Ken-ichi Okada†

Real-time streaming services are attracting attention. However, an adversary can compromise the safety of these services in ways such as data tampering, spoofing, and repudiation. In this paper we propose a real-time Stream Authentication scheme for Video streams called SAVe. Each packet in the stream is authenticated to correspond to the packet loss seen in UDP-based streaming. The amount of redundancy distributed to each frame is also adjusted according to the importance of each frame, to take account of the special characteristics of video such as differences in importance of and dependencies between frames. Since temporal and spatial compression techniques are adopted for video stream encoding, SAVe is efficient in terms of making important frames robust to packet loss. The simulation results show that the authentication rate is on average approximately equivalent to that of previously proposed schemes. An improvement of 50% in the playing rate over previously proposed schemes can be seen when the packet loss rate is 20%.

1. Introduction

As a result of the explosive growth of broadband networks, many users are able to enjoy services via the Internet. Real-time streaming services such as IP telephony and video conferencing are attracting particular attention [1]. However, these services are still affected by various issues such as data tampering, spoofing and repudiation. For example, using future high-performance computers, it will become possible to tamper with data in real time by methods such as inserting tampered frames into the data stream. Countermeasures are therefore needed so that these services can be used in critical situations, such as business negotiations.

In real-time streaming services, to maintain real-time transmission, real-time transmission protocols on top of connectionless best-effort services (e.g., the User Datagram Protocol) are generally used [2),3)]. As a result, packet loss is frequently seen [4)]. When dealing with stream authentication, one must consider the effects of packet loss, since consecutive authentication must be attained [5)]. To address the packet loss issue, each packet in the stream must be authenticated. Digital signatures are generally used for message authentication. However, when using these signatures, one must take account of the computation overhead. Authentication of each individual packet is possible by signing each packet with the sender's digital signature, but at the cost of computation overhead.

Video sequences are compressed by removing spatial and temporal redundancies between frames. Therefore there are dependencies between frames, and the importance of each frame is different. If a high-priority frame is lost as a result of packet loss, all frames referring to the lost frame become unplayable, even if they reach the receiver side. High-priority frames should therefore be robust to packet loss. However, as pointed out in Ref. 6), existing stream authentication schemes do not take account of the characteristics specific to video. Since all frames are handled at the same level, efficiency is poor.

In this paper we propose a stream authentication scheme where the amount of redundancy distributed to each frame is adjusted according to that frame's importance. Our scheme uses a technique called the Information Dispersal Algorithm to reconstruct lost packets so as to enable efficient authentication.

In the next section, we discuss related work. In Section 3, we explain our approach to real-time stream authentication for video streams. In Section 4, we evaluate our scheme. In Section 5, we discuss the simulation results. Finally, we offer some concluding remarks in Section 6.

2. Related Work

In this section, we will explain the Information Dispersal Algorithm (IDA) [7)] technique, which is used in our scheme, and review several existing stream authentication schemes.

† Faculty of Science and Technology, Keio University

2.1 Information Dispersal Algorithm

When packets are streamed via the Internet using UDP, packet loss is frequently seen. Since UDP only provides best-effort services, no error recovery that includes packet retransmission is carried out. In our scheme, therefore we use a forward error correction technique, namely, an erasure codes called IDA, to recover the lost packets. The basic idea of IDA is to distribute data with some amount of redundancy to multiple data during transmission. On the receiver side, if a sufficient number of parts of the original data are received, the original data can be reconstructed by using the received data, even when packet loss occurs. For example, say a data A of size F is distributed to n packets, and the receiver side must receive at least m $(0 < m \leq n)$ packets in order to reconstruct A. The executions on the sender side are shown as follows:

(1) Data A is divided into pieces of length m. The number of divided pieces B that are generated is F/m.
(2) Using all the divided pieces B for computation, n reconstruction data C_i $(i = 1, 2, \ldots, n)$ are generated.
(3) Each reconstruction data C_i is of size F/m; thus, the total size of all n number of C_i is Fn/m.

On the receiver side, if at least m number of reconstruction data C_i are received, data A can be reconstructed. The size of the distributed reconstruction data can be adjusted by changing the values of m and n.

2.2 Existing Stream Authentication Schemes

Several schemes that attempt to efficiently authenticate streamed media have been proposed.

Gennaro and Rohatgi propose a scheme called the Hashed Chain technique [8], where the verification of a packet is dependent on other packets. The basic idea is as follows. Each packet contains the hash value of the next packet, and only the first packet in the stream is signed. Only one signature is needed, since the hash values act as a chain between the packets. To reduce the sender side delay, a stream of packets is divided into blocks and the above process is repeated for each block. While this scheme reduces the computation overhead, it does not tolerate packet loss. Non-resistance to packet loss becomes a drawback in stream authentication.

Wong and Lam propose a scheme that uses Merkle's signature trees [9),10)] to sign streams [11]. We will call their scheme Authentication Tree. Their idea is to make each packet individually verifiable, in order to tolerate packet loss. To achieve this, however, each packet needs to contain the signature of the root node and all the hashes of the nodes necessary to compute the root. This causes a long sender-side delay and a large space overhead (the overhead of authentication information).

Park proposes a scheme called SAIDA [12),13)] (Signature Amortization using IDA). First, the hash values of each packet are concatenated. Then, the hash of this concatenated value is computed. This value is called as the group hash. In SAIDA, only the group hash is signed. Next, the reconstruction data of the group hash and the signature are generated using the IDA encoding process. The reconstruction data are then distributed to each packet in the group. Because it uses IDA, this scheme is tolerant to packet loss.

The common characteristics of these existing schemes are that they all handle multiple packets as one group and the packets in the group are handled at the same level. Handling packets that store the frame data of video streams at the same level is the same as handling each frame at the same level. However, as mentioned in Section 1, there are dependencies between frames, and the importance of frames differs. Consequently, the efficiency is poor when these schemes are used for video streams.

3. SAVe: A Real-Time Stream Authentication Scheme for Video Streams

In this section, we propose a real-time Stream Authentication scheme for Video streams called SAVe. SAVe uses digital signatures, hashes, and IDA to take account of the characteristics of video. The authentication information is appended to a high-priority frame and the reconstruction data for the authentication information concatenated with the frame data are created by employing IDA. These reconstruction data are distributed to the low-priority frames. In this way, high-priority frames are made robust to packet loss, and efficient stream authentication is enabled. The reconstruction data stated here are those mentioned in Section 2.1, which are used to reconstruct the data lost during transmission.

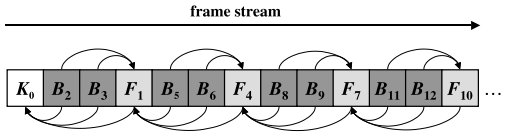


Fig. 1 Reference relation between frames.

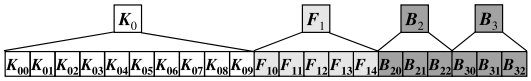


Fig. 2 Example of K, F, B frame data.

3.1 Assumptions

SAVe is based on the following assumptions:

- We assume that temporal compression techniques such as forward prediction and bi-directional prediction are used for video stream encoding, as in MPEG-2 [14].
- When a key-frame is lost on account of packet loss, all sub-frames dependent on the key-frame become unplayable on the receiver side.
- A frame can be carried in several UDP packets.

In this paper we call a frame coded with reference only to the current frame as key-frame, and frames coded with reference to both the current frame and other frames as sub-frames.

Figure 1 shows the reference relations between the key-frames K, the forward predicted sub-frames F, and the bi-directional predicted sub-frames B that are assumed in our scheme.

The arrows show the reference relationship between the frames. Frame K is the most important frame and frame B is the least important frame when decoding on the receiver side.

In our proposal, we also assume that an entire video stream consists of groups of frames. The characteristics of a frame group are as follows:

- One frame group consists of about 13 frames.
- The first frame in the frame group is a K frame.
- An F frame is present every 2–3 frames.

The size of each type of frame is as follows. F frames are 40–50% the size of K frames, and B frames are 20–30% the size of K frames. For example, if a K frame is stored in 10 UDP packets, F and B frames are stored in 5 and 3 UDP packets, respectively. **Figure 2** shows an example of the frame data when K, F, and B frames are stored in 10, 5, and 3 packets, respectively.

Figure 3 shows the order of generation of frames on the sender side and the order of playing on the receiver side. It can be seen that the generation order and the playing order are different. For example, F_1 is generated right after K_0 on the sender side, but played after B_2 and B_3 on the receiver side. This is because B_2 and B_3 are generated by using bi-directional prediction with reference to K_0 and F_1.

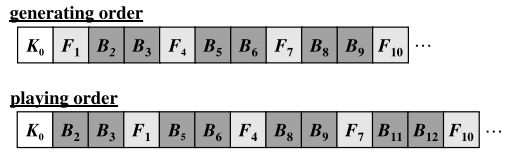


Fig. 3 Generation and playing order.

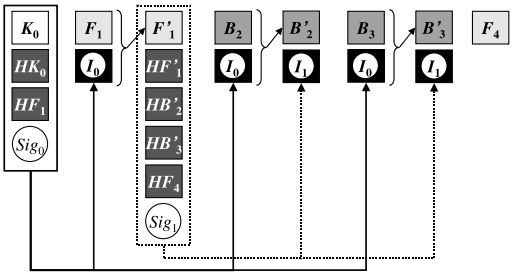


Fig. 4 Overview of the signature method of SAVe.

3.2 Signature Method

In SAVe, the reconstruction data of frames with high priority such as K frames are appended to frames with lower priority such as F and B frames. The reconstruction data of F frames are appended to B frames, which have the lowest priority. In this manner, the reconstruction data are appended hierarchically. Therefore, the most important K frames are made robust to packet loss and efficient authentication also becomes possible. **Figure 4** shows an overview of the signature method of SAVe.

- K, F, and B denote frame data.
- HK, HF, and HB denote the hash values of K, F, and B, respectively.
- Sig denotes the digital signature.
- I denotes the reconstruction data.
- F' and B' denote the data after the reconstruction data have been appended to F and B, respectively.

First, HK_0 and HF_1, the hash values of K_0 and F_1, are appended to the key-frame K_0. Then Sig_0, the digital signature for the above concatenated value, is generated. Sig_0 can be expressed as follows:

$$Sig_0 = Enc(KEY_s, Hash(K_0 \parallel HK_0 \parallel HF_1)) \quad (1)$$

where K_s is the private key used in public-key cryptography.

I_0, the reconstruction data for the data enclosed in the box in the left side of Fig. 4, are then generated by using the IDA encoding process. Therefore, the reconstruction data of the authentication information concatenated with the frame data are generated in our scheme. Next, I_0 is distributed to frames F_1, B_2, and B_3. As mentioned above, F_1', B_2', and B_3' denote the data after I_0 has been appended to F_1, B_2, and B_3, respectively.

Next, HF_1', HB_2', HB_3', and the hash value of the next frame , HF_4 are appended to F_1'. Then Sig_1, the digital signature for the concatenated value, is generated. The reconstruction data I_1 are then generated by using the IDA encoding process. Finally, I_1 are distributed to B_2' and B_3'. In this manner, the reconstruction data of the high-priority frames are distributed hierarchically to the low-priority frames, thus facilitating the reconstruction of high-priority K frames. Furthermore, distributing the reconstruction data in an interleaved fashion is an effective technique for reducing the effects of burst loss. However, interleaving packets in the stream necessitates an extra delay [15),16)]. Therefore, interleaving techniques are not used in our scheme.

In our scheme, all processes are carried out at a packet level instead of a frame level. Details are given in the following subsections.

3.2.1 Signature Method of *K* Packets

In this section the signature method of K packets is explained. The reconstruction data of the authentication information and the frame data of K packets are distributed to F and B packets. K, F, and B packets are those that store the frame data of K, F, and B frames, respectively. Each K frame K_i is divided into L_k packets (K_{i0}, ..., K_{iL_k}). Each F frame F_{i+1} is divided into L_f packets ($F_{(i+1)0}$, ..., $F_{(i+1)L_f}$). Each B frame B_{i+2} is divided into L_b packets ($B_{(i+2)0}$, ..., $B_{(i+2)L_b}$). The flow of the following explanation is shown in **Fig. 5**. The intersection of arrows denotes the concatenation of data.

First, the hash values of all packets K_{i0}, ..., K_{iL_k} are computed and concatenated with each other. The hash value of this concatenated value is then computed and expressed as $H(K_{i0-iL_k})$.

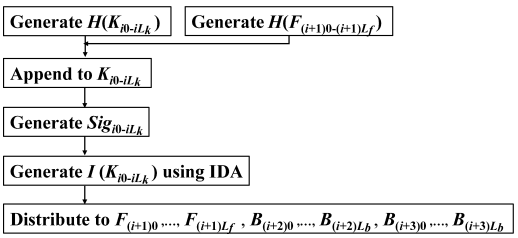


Fig. 5 Signature method of K packets.

$$H(K_{i0-iL_k}) = Hash(H(K_{i0}) \parallel H(K_{i1}) \parallel \cdots \parallel H(K_{iL_k})) \quad (2)$$

In the same manner, $H(F_{(i+1)0-(i+1)L_f})$ is computed. In order to maintain concatenation between packets in terms of authentication, $H(F_{(i+1)0-(i+1)L_f})$ is concatenated with the formerly generated $H(K_{i0-iL_k})$. Next, this is appended to K_{i0-iL_k}. Here, K_{i0-iL_k} denotes packets K_{i0} through K_{iL_k}. Then, the digital signature shown in the following equation is generated:

$$Sig_{i0-iL_k} = Enc(KEY_s, Hash((K_{i0-iL_k}) \parallel H(K_{i0-iL_k}) \parallel H(F_{(i+1)0-(i+1)L_f}))) \quad (3)$$

The IDA encoding process is then carried out using the data with the digital signature, to create the reconstruction data $I(K_{i0-iL_k})$. Finally, this $I(K_{i0-iL_k})$ is distributed to packets $F_{(i+1)0}$, ..., $F_{(i+1)L_f}$, $B_{(i+2)0}$, ..., $B_{(i+2)L_b}$, and $B_{(i+3)0}$, ..., $B_{(i+3)L_b}$. Here we will call the packets that carry the reconstruction data as $F'_{(i+1)0}$, ..., $F'_{(i+1)L_f}$, $B'_{(i+2)0}$, ..., $B'_{(i+2)L_b}$, and $B'_{(i+3)0}$, ..., $B'_{(i+3)L_b}$.

We will call the number of F' and B' packets which carry the reconstruction data $I(K_{i0-iL_k})$ K reconstruction-packets and define it as n_k. We will also call the number of packets needed to reconstruct the K packets in the case of packet loss the K reconstruction-threshold and define it as m_k.

3.2.2 Signature Method of *F* packets

In this section the signature method of F packets is explained. The reconstruction data of the authentication information and the frame data of F packets are only distributed to B packets. The flow of the following explanation is shown in **Fig. 6**.

First $H(F'_{(i+1)0-(i+1)L_f})$, $H(B'_{(i+2)0-(i+2)L_b})$, $H(B'_{(i+3)0-(i+3)L_b})$, and $H(F_{(i+4)0-(i+4)L_f})$ are computed. Next, these hash values are concatenated and appended to $F'_{(i+1)0-(i+1)L_f}$.

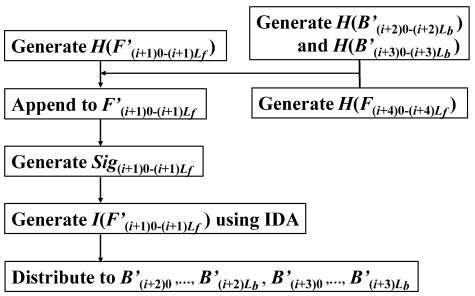


Fig. 6 Signature method of F packets.

Then the digital signature $Sig_{(i+1)0-(i+1)L_f}$ is generated. The IDA encoding process is then carried out, using the data with the digital signature, to create the reconstruction data $I(F'_{(i+1)0-(i+1)L_f})$. Finally, these $I(F'_{(i+1)0-(i+1)L_f})$ are distributed to packets $B'_{(i+2)0}, \ldots, B'_{(i+2)L_b}$ and $B'_{(i+3)0}, \ldots, B'_{(i+3)L_b}$ and the resulting data are denoted as $B''_{(i+2)0}$, $\ldots$, $B''_{(i+2)L_b}$ and $B''_{(i+3)0}, \ldots, B''_{(i+3)L_b}$.

We will call the number of B'' packets that carry the reconstruction data $I(F'_{(i+1)0-(i+1)L_f})$ F reconstruction-packets and define it as n_f. We will also call the number of packets needed to reconstruct the F' packets in the case of packet loss the F reconstruction-threshold and define it as m_f.

Similar processes are carried out for the F packets of the next F frames in the same frame group. These F packets do not carry reconstruction data of K packets. We will call the number of B packets that carry the reconstruction data $I(F_{(i+4)0-(i+4)L_f})$ F_{next} reconstruction-packets and define it as n_{fnext}. We will also call the number of packets needed to reconstruct the F packets in the case of packet loss the F_{next} reconstruction-threshold and define it as m_{fnext}.

3.3 Verfication Method

The verification method of SAVe is explained in this section. The explanation is divided into two cases: (i) verification when there is no packet loss and (ii) verification when there is packet loss. The explanation uses the examples from Section 3.2.

3.3.1 Verfication When There is No Packet Loss

When there is no packet loss, ordinary verification using digital signature is carried out. On receiving K_{i0-iL_k}, the receiver verifies K_{i0-iL_k} using the authentication information appended to K_{i0-iL_k}. Using KEY_p, the public key of the public-key cryptography, Sig_{i0-iL_k} is decrypted and the hash value of $H(K_{i0-iL_k})$ and $H(F_{(i+1)0-(i+1)L_f})$ appended to K_{i0-iL_k} is computed. The decrypted value and the computed hash value are compared, and if the two are equal, K_{i0-iL_k} is verified. Together, $F_{(i+1)0-(i+1)L_f}$ is also verified.

3.3.2 Verification When There is Packet Loss

When packet loss exists, SAVe uses the reconstruction data created by the IDA encoding to reconstruct the lost packets.

In this explanation we will assume that several K packets are lost during transmission. Furthermore, we will divide the explanation into two cases: (i) that in which the number of F' and B' packets received exceeds the K reconstruction-threshold and (ii) that in which the number of F' and B' packets received is below the K reconstruction-threshold.

If several packets $K_{i0}, \ldots, K_{iL_k}$ are lost, reconstruction of the lost packets is possible if at the minimum m_k packets out of $F'_{(i+1)0}, \ldots, F'_{(i+1)L_f}$, $B'_{(i+2)0}, \ldots, B'_{(i+2)L_b}$, and $B'_{(i+3)0}, \ldots, B'_{(i+3)L_b}$ are received on the receiver side. Since the authentication information is included in the reconstructed data, consecutive authentication of the packets is enabled. Verification is carried out in the manner explained in Section 3.3.1.

When the number of packets received is below m_k, direct reconstruction of $K_{i0}, \ldots, K_{iL_k}$ is not possible. Therefore, as the first step of reconstruction, packets $F'_{(i+1)0}, \ldots, F'_{(i+1)L_f}$, which carry the reconstruction data of $K_{i0}, \ldots, K_{iL_k}$ need to be reconstructed. If the number of $B''_{(i+2)0}, \ldots, B''_{(i+2)L_b}$ and $B''_{(i+3)0}, \ldots, B''_{(i+3)L_b}$ packets which carry the reconstruction data $I(F'_{(i+1)0-(i+1)L_f})$ exceeds m_f, reconstruction and authentication of $F'_{(i+1)0}, \ldots, F'_{(i+1)L_f}$ are possible. The reconstruction data of $K_{i0}, \ldots, K_{iL_k}$ are included in the reconstructed data. Therefore reconstruction and authentication of $K_{i0}, \ldots, K_{iL_k}$ are enabled.

As described in this section, in SAVe the reconstruction data of high-priority packets are distributed hierarchically to low-priority packets. This enables a high reconstruction rate of high-priority packets.

4. Evaluation

To evaluate SAVe, we ran simulations. Since the packet loss probability changes over time, it

Table 1 Simulation parameters.

Parameter	Value
Total number of packets sent: N	10,008 packets
K reconstruction-packets: n_k	16 packets
K reconstruction-threshold: m_k	11, 15 packets
F reconstruction-packets: n_f	10 packets
F reconstruction-threshold: m_f	4, 9 packets
F_{next} reconstruction-packets: n_{fnext}	5 packets
F_{next} reconstruction-threshold: m_{fnext}	4 packets
Packet loss probability: π	0%, ..., 40%
Expected burst loss length: β	8 packets
Size of K frames: S_k	2.5 Kbytes
Size of F frames: S_f	1.25 Kbytes
Size of B frames: S_b	625 bytes
Size of packet: S_p	1,500 bytes

is difficult to evaluate our scheme by using real networks. We therefore use a packet loss model and ran simulations over virtual networks.

4.1 Simulation Environment

Simulations were run on a Pentium 4 2.80 GHz CPU, 2.0 GB RAM processor. The simulation program is written in JDK 1.4.2. We used 160 bit SHA-1 hash functions and 1,024 bit RSA for digital signatures, though our authentication scheme is independent of the type of hash functions and digital signatures.

4.1.1 Packet Loss Model

In commonly used networks, packet loss occurs in bursts rather than at random intervals. The two-state Markov Chain Loss Model is widely used to express burst packet losses [17)~19),21)], especially in evaluating stream authentication schemes [13),24)]. In our simulation, therefore, we use the two-state Markov Chain Loss Model as the packet loss model.

4.1.2 Simulation Parameters

We carried out comparative evaluations by running the simulation program using the packet loss model mentioned in Section 4.1.1. We compare the results of SAVe and the existing scheme. The parameters of the simulation program are shown in **Table 1**.

We assume the use of SAVe when the size of the video is about 120 × 160 pixels, like NetMeeting by Windows. S_k, S_f, and S_b are set to 2.5 Kbytes, 1.25 Kbytes, and 625 bytes, respectively, because these are the sizes of each frame of such video when MPEG-2 is used. Under normal conditions, the sizes of sub-frames are obtained by actually encoding the video. However, in the simulation experiments, virtual packets were used and therefore we set the size of the frames to typical actually measured sizes.

The maximum value of π was set to 40%, since it is highly unlikely that the packet loss probability via the Internet will be greater than 40%. Much research has been done on measurements of packet loss over the Internet [20)~23)]. The results from Ref. 23), state that the average packet loss probability is about 7%. β is set to 8, since the average burst packet loss length over the Internet is less than 8 packets.

4.1.3 Evaluation Criteria

The evaluation items are authentication rate, playing rate, communication overhead, and delay time. The authentication rate is the number of packets authenticated on the receiver side divided by the total number of packets sent from the sender side. The playing rate is the number of playable authenticated packets on the receiver side divided by the total number of packets sent from the sender side. The playing rate must be evaluated, since in existing schemes, there are packets that are unplayable on the receiver side, even though they are authenticated. The communication overhead is the average size of the reconstruction data of the authentication information and frame data appended per packet. The delay time is the average delay time for a packet to be authenticated after it has been created by the sender side. This delay time includes the time necessary for IDA computation, if this is required due to packet loss.

We will compare SAVe with SAIDA, since SAIDA is the only scheme that uses erasure codes.

4.2 Results

4.2.1 Authentication Rate

Figure 7 shows the relation of the packet loss rate and the authentication rate. The legend of the graph are shown as "scheme name-n_k, m_k, n_f, m_f, n_{fnext}, m_{fnext}".

The authentication rates of SAVe and SAIDA are approximately equivalent. The value of m_k

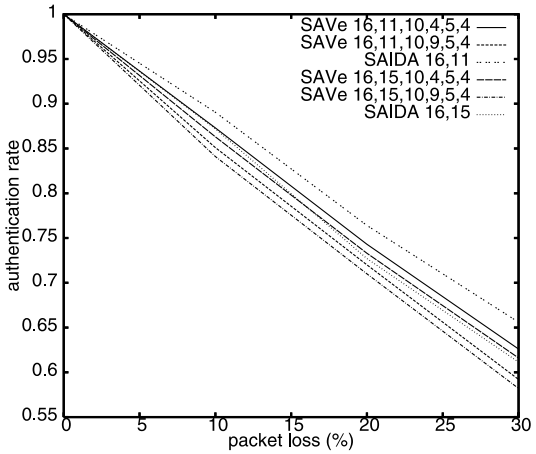


Fig. 7 Authentication rate.

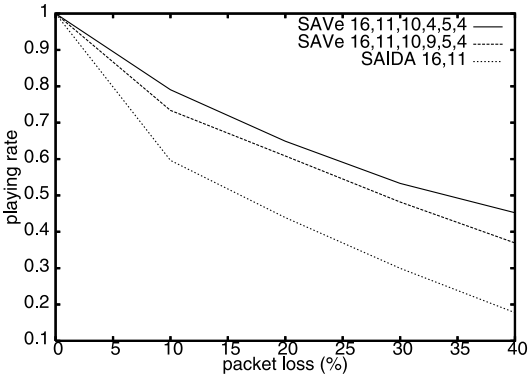


Fig. 8 Playing rate.

is the reconstruction-threshold of K packets, and represents the reconstruction conditions. In general, the greater the value of m_k, the stricter the reconstruction condition becomes, since the redundancy of the reconstruction data distributed to each packet is small.

First, for the easy reconstruction condition ($m_k = 11$), the authentication rate of SAIDA is slightly higher than that of SAVe. In SAIDA, all packets in a group are handled at the same level. Therefore the reconstruction data of the authentication information are distributed to all the packets in the group. In SAVe, on the other hand, the distribution of the reconstruction data is changed according to the priority of the packet. Therefore, when a low-priority packet is lost, the reconstruction rate of the packet is low and thus the authentication rate becomes lower than that of SAIDA.

Next, for the strict reconstruction condition ($m_k = 15$), the authentication rate of SAVe is higher than that of SAIDA. This shows that SAVe is more effective when the reconstruction condition is strict. When high-priority K packets are lost, SAVe reconstructs the authentication information of K packets more easily than SAIDA, since hierarchical reconstruction of K packets is possible in SAVe.

We now give the results for the authentication rate when not only m_k but also m_f is changed. Figure 7 shows that when m_f is a smaller value, the authentication rate becomes higher. The difference in the authentication rate is also greater when m_f is changed instead of m_k. This shows the effectiveness of the hierarchical reconstruction ability of K packets in SAVe.

From the above results, it is possible to say that SAVe has the same tolerance to packet loss as SAIDA in terms of authentication.

4.2.2 Playing Rate

Figure 8 shows the relation of the packet loss rate and the playing rate. The legend of the graph are shown as "scheme name-n_k, m_k, n_f, m_f, n_{fnext}, m_{fnext}".

SAVe maintains a higher playing rate than SAIDA. For example, when the packet loss rate is 20%, the playing rates of SAVe and SAIDA are 0.65 and 0.45, respectively. This shows that SAVe has a 50% better playing rate than SAIDA.

In SAVe the reconstruction of K packets is easily carried out in the case of packet loss, and the received F and B packets are not wasted. What we mean by waste of packets is, inability to play the received sub-frames because the key-frames are lost.

The playing rate of SAVe is lower than the authentication rate. In SAIDA, however, there is a significant difference between the playing rate and the authentication rate, because if m_k packets are received, all packets in the group are authenticated. This means that packets of sub-frames are authenticated even when packets of key-frames are lost. This leads to the presence of frames that are unplayable even though they are authenticated. Thus the authentication rate of SAIDA is significantly higher than the playing rate. Consequently, SAIDA is not tolerant to packet loss in terms of playing and is therefore not fit for authenticating streams when there are dependencies between packets, since many authenticated packets are wasted.

4.2.3 Communication Overhead

Figure 9 shows the relation between the communication overhead and the playing rate.

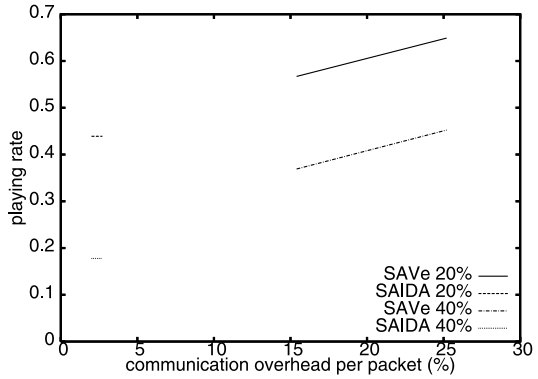


Fig. 9 Communication overhead.

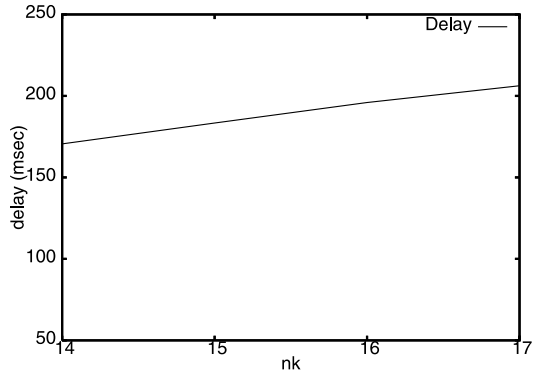


Fig. 10 Delay time.

The legend of the graph are shown as "scheme name-π". The value of the playing rate is shown when π is 20% and 40%.

Figure 9 shows that the communication overhead per packet in SAVe is larger than that in SAIDA. Since our scheme distributes the reconstruction data of not only the authentication information but also the frame data, the overhead is larger than in SAIDA. However, this enables a higher playing rate than in SAIDA.

4.2.4 Delay Time

Figure 10 shows the delay time of SAVe according to the size of n_k. The delay time is the average time of all possible m's for a single n_k needed for a packet to be authenticated on the receiver side after the packet has been created by the sender side. This includes the time needed for signing and verification and for IDA encoding and decoding.

The relation between n_k and the delay time is that, the larger n_k becomes, the longer the delay time becomes. This is mainly due to the characteristic of the IDA process that the computing time is dependent to the size of n, that is, the number of packets to which the reconstruction data are distributed. The delay time of SAVe in our assumed use is under 200 msec. It is stated in the Technical Reports on IP Network Technology [25] by the Ministry of Internal Affairs and Communication that the delay time for IP telephony should be under 400 msec. Considering the fact that the auditory sense of the human being is more sensitive to delay than the visual sense, the delay time of SAVe is within the acceptable limits.

The delay is mainly caused by the calculation of the IDA process. Our program was written in Java, and therefore the delay time is likely to be shortened by a large amount when a hardware IDA codec is used.

5. Discussion

In this section we will discuss the simulation results of our scheme in terms of long burst packet loss, packets arriving out of order, and frame size. We will also discuss the acceptability of our scheme.

5.1 Long Burst Packet Loss

Research results have shown that packet loss on the Internet occurs in bursts. The average packet loss probability, as stated in Section 4.1.2, is approximately 7%. In our simulation, the expected burst length was set to 8 packets, since the average burst packet length is less than 8 packets. In these normal conditions our scheme is tolerant to packet loss. If a K packet is lost, m_k packets are required to reconstruct the lost K packet. In the simulation, m_k is set to 11 packets. Therefore, if the expected burst loss is set higher than m_k packets, the authentication and playing rate will decrease. On the Internet there are cases when longer bursts, such as several tens of packets, occur [20]. Our simulations do not take these characteristics of long burst packet loss into consideration, since they are rare cases. Furthermore, frames in such burst loss are unplayable and therefore authentication is not necessary anyway.

5.2 Packets Arriving Out of Order

On the Internet there are cases when packets are received out of order due to changes in the routing path. In UDP, packets with a certain delay are considered lost. According to the results of Ref. 23), the majority of packets that are received out of order are out by a single packet. Therefore, if the decoder has a buffer of one packet, receiving packets out of order is not an issue in most cases. In our scheme,

when a K packet is lost, to reconstruct and authenticate that packet, m_k out of n_k packets of the reconstruction data must be received. It is readily seen that this is equivalent to having a buffer on the decoder side, since our scheme needs at the most a buffer of n_k packets. Our scheme is tolerant to packets received out of order if the number within n_k. In our scheme, therefore, if the packets containing the reconstruction data are received out of order by over n_k packets, the lost packet is not reconstructed or authenticated, since the packets with the reconstruction data are determined to be lost.

5.3 Frame Size

In the simulation, the frame sizes of K, F, and B frames were set to 2.5 Kbytes, 1.25 Kbytes, and 625 bytes respectively. As mentioned in Section 4.2.4, the delay time of SAVe is mainly caused by the computing time of the IDA process, which is dependent on the size of n_k. Therefore, when high-resolution video is used, the frame sizes of K, F, and B increase and the number of reconstruction-packets n_k, n_f, and n_{fnext} also increases. This leads to a greater delay. The sizes of frames should be set according to the acceptable delay limits of the intended purpose.

5.4 Acceptability

Two types of acceptability due to packet loss in our scheme can be discussed. One is the authentication rate and the other is the video quality. As shown in Section 4, our scheme is acceptable in the terms of the authentication rate, since it is equivalent to the previously proposed scheme. We will discuss the playing rate, which is the number of playable authenticated packets divided by the total number of packets sent. There are results whereby a packet loss rate as low as 3% can translate into a frame error rate as high as 30% in MPEG-2 [23]. The frame error rate mentioned in this paper is the degree to which a frame is affected by loss in the current frame or the frame from which the current frame is predicted. This frame error rate is equivalent to the unplayable rate (1 − playing rate) in our scheme, since in our scheme, the current frame is unplayable if the frame from which the current frame is predicted is lost. At a packet loss rate of 3%, the unplayable rate in our scheme is less than 10%. Therefore, our scheme compares favorably with MPEG-2 in this respect.

However, it is difficult to generalize the acceptability of the video quality due to packet loss, since when measuring the acceptability of video quality, factors such as the purpose of the video, the scene content, and human perceptions come into play. Therefore, there are no general values for the acceptability of video quality [26].

6. Conclusion

In this paper we have proposed SAVe, a real-time stream authentication scheme that takes account of the characteristics of video. Through simulation results, we show the scheme's effectiveness. The authentication rate of SAVe is on average approximately equivalent to that of SAIDA, but it is higher than that of SAIDA in strict reconstruction conditions. At a packet loss rate of 20%, which is higher than the average packet loss rate of the Internet, the playing rate of SAVe is 50% better than that of SAIDA. Therefore, SAVe enables efficient real-time authentication of video streams. It is possible to say that our scheme is a necessary technology for the spread of safe real-time streaming services.

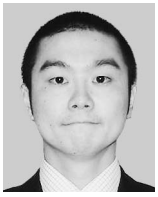
Acknowledgments This work is supported in part by JSPS Research Fellowships for Young Scientists.

References

1) Varshney, U., Snow, A., McGivern, M. and Howard, C.: Voice Over IP, *Comm. ACM*, Vol.45, No.1, pp.89–96 (Jan. 2002).
2) Shahbazian, J. and Christensen, K.J.: TSGen: A tool for modeling of frame loss in streaming video, *International Journal of Network Management*, pp.315–327 (2004).
3) Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V.: RTP: A transport protocol for real-time applications, RFC 1889 (1996).
4) Floyd, S. and Fall, K.: Promoting the Use of End-to-End Congestion Control in the Internet, *IEEE/ACM Transactions on Networking*, Vol.7, No.4, pp.458–472 (1999).
5) Ueda, S., Eto, S., Kawaguchi, N., Uda, R., Shigeno, H. and Okada, K.: Real-time Stream Authentication Scheme for IP Telephony, *IPSJ Journal*, Vol.45, No.2, pp.605–613 (Feb. 2004).
6) Kaneko, S., Ueda, S., Kawaguchi, N., Ogino, T., Shigeno, H. and Okada, K.: Proposal of Real-time Stream Authentication Scheme for Motion Pictures, *IPSJ SIG Technical Reports DPS–122*, pp.211–216 (2005).
7) Rabin, M.: Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance, *J. ACM*, No.2, pp.335–348 (1989).

8) Gennaro, R. and Rohatgi, P.: How to Sign Digital Streams, *CRYPTO 1997*, LNCS1294, pp.180–197 (1997).
9) Merkle, R.: A Certified Digital Signature, *Proc. Conference on Advances in Cryptology*, pp.218–238 (1989).
10) Merkle, R.: A Digital Signature Based on a Conventional Encryption Function, *Proc. Conference on Advances in Cryptology*, pp.369–378 (1987).
11) Wong, C. and Lam, S.: Digital Signature for Flows and Multicasts, *IEEE/ACM Transactions on Networking*, Vol.7, No.4, pp.502–513 (1999).
12) Park, J., Chong, E. and Siegel, H.: Efficient multicast packet authentication using signature amortization, *Proc. IEEE Symposium on Research in Security and Privacy*, pp.227–240 (2002).
13) Park, J., Chong, E. and Siegel, H.: Efficient Multicast Stream Authentication Using Erasure Codes, *ACM Trans. Inf. Syst. Security*, pp.258–285 (May 2003).
14) ISO/IEC 13818-2:2000, Information Technology-Generic Coding of Moving Pictures and Associated Audio Information (2000).
15) Perkins, C., Hodson, O. and Hardman, V.: A Survey of Packet Loss Recovery Techniques for Streaming Audio, *IEEE Network*, pp.40–48 (Sep./Oct. 1998).
16) Sinha, R., Papadopoulos, C. and Kyriakakis, C.: Loss Concealment for Multi-Channel Streaming Audio, *International Workshop on Network and Operating System Support for Digital Audio and Video*, pp.100–109 (June 2003).
17) Widmer, J., Boutremans, C. and Boudec, J.: End-to-End congestion control for TCP-friendly flows with variable packet size, *ACM SIGCOMM Computer Communication Review*, Vol.34, Issue 2, pp.137–151 (2004).
18) Shmueli, R., Huber, R. and Hadar, O.: Effects of frame rate, frame size and MPEG2 compression on the perceived compressed video quality tranmitted over lossy IP networks, *2nd International Conference on Information Technology: Research and Education*, pp.49–54 (June 2004).
19) Ma, L. and Ooi, W.: Retransmission in Distributed Media Streaming, *International Workshop on Network and Operating System Support for Digital Audio and Video*, pp.117–122 (June 2005).
20) Loguinov, D. and Radha, H.: Measurement Study of Low-bitrate Internet Video Streaming, *Proc. 1st ACM SIGCOMM Workshop on Internet Measurement*, pp.281–293 (2001).
21) Yajnik, M., Moon, S., Kurose, J. and Towsley, D.: Measurement and modeling of the Temporal Dependence in Packet Loss, *Proc. IEEE Conference on Computer Communications*, pp.345–352 (1999).
22) Paxson, V.: End-to-End Internet Packet Dynamics, *IEEE/ACM Transactions on Networking*, Vol.7, No.3, pp.277–292 (June 1999).
23) Boyce, J. and Gaglianello, R.: Packet Loss Effects on MPEG Video Sent Over the Public Internet, *Proc. 6th ACM international conferecne on Multimedia*, pp.181–190 (1998).
24) Perrig, A., Canetti, R., Tygar, J. and Song, D.: Efficient Authentication and Signing of Multicast Streams over Lossy Channels, *IEEE Symposium on Security and Privacy*, pp.56–73 (2000).
25) Technical Reports on IP Network Technology, The Ministry of Internal Affairs and Communication, 2002.
26) Reibman, A., Kanumuri, S., Vaishampayan, V. and Cosman, P.: Visibility of individual packet loss in MPEG-2 video, *International Conference on Image Processing IEEE ICIP'04*, pp.171–174 (Oct. 2004).

(Received May 9, 2005)
(Accepted November 1, 2005)
(Online version of this article can be found in the IPSJ Digital Courier, Vol.2, pp.70–80.)



Shintaro Ueda received the B.S. in Department of Information and Computer Science from Keio University, Japan in 2002, the M.S. degree in Open and Environment Systems from Keio University in 2005. He is currently working toward the Ph.D. degree in Open and Environment Systems at Keio University. His research interests includes Network Security.

Shin-ichiro Kaneko received the B.S. in Department of Information and Computer Science from Keio University, Japan in 2004. He is currently working toward the M.S. degree in Open and Environment Systems at Keio University. His research interests are Network Security.

Nobutaka Kawaguchi received the B.S. in Department of Information and Computer Science from Keio University, Japan in 2003, the M.S. degree in Open and Environment Systems from Keio University in 2005. He is currently working toward the Ph.D. degree in Open and Environment Systems at Keio University. His research interests includes Network Security.

Hiroshi Shigeno received the B.S., M.E. and Ph.D. degree in instrumentation engineering from Keio University, Japan in 1990, 1992 and 1997 respectively. Since he has been with the Department Information and Computer Science, Keio University, where he is currently an assistant professor. His current research interests include computer networking architecture and protocols, mobile and ubiquitous computing, and agent computing and communications. He is a member of IPSJ.

Ken-ichi Okada received the B.S., M.E. and Ph.D. degree in instrumentation engineering from Keio University, in 1973, 1975 and 1982 respectively. kHe is currently an professor in the Department of Information and Computer Science at Keio University. His research interests include CSCW, groupware, human computer interaction and mobile computing. He has published over 100 technical papers and reports and books entitled "Collaboration and Communication", "Intelligent Inspired Information Society", "Designing Communication and Collaboration Support Systems" and "Introduction to Groupware". He is a member of IEEE, ACM, and IPSJ. He was a chair of SIGGW, a chief editor of IPSJ Journal, and an editor of IEICE Transactions. Dr. Okada received the IPSJ Best Paper Award in 1995, 2001 and IPSJ 40th Anniversary Paper Award in 2000.