Video Streaming Authentication over Lossy Channels

Stefano Chessa  
Dip. di Informatica  
Università di Pisa, and  
ISTI-CNR, Pisa  
ste@di.unipi.it

Gaetano Giunta  
Dip. di Elettronica Applicata  
Università di Roma Tre, and  
ISTI – CNR, Pisa  
giunta@ieee.org

Roberto Di Pietro  
Dip. di Matematica  
Università di Roma Tre  
dipietro@di.uniroma1.it

Gabriele Oligeri  
ISTI-CNR  
Area della Ricerca di Pisa  
gabriele.oligeri@isti.cnr.it

Abstract

We consider the problem of source and content authentication in video streaming applications in multicast satellite networks. In particular we propose a novel method which combines signature amortization by means of hash chain with watermarking techniques. This approach does not introduce bandwidth overhead and computational overhead is reduced using signature amortization by means of an hash chain. We present simulation results about authentication robustness to transmission over a lossy channel.

Keywords

Video, Multicast, Authentication, Satellite

INTRODUCTION

Video broadcasting/multicasting is one of the most important services in many environments. In particular it is a very popular application in satellite communications (due to their implicit broadcasting capability) as it is also witnessed by emerging video streaming standards such as DVB-S [1]. In order to enable a widespread and trusted streamed media dissemination are necessary in this scenario integrity and authenticity guarantees. In fact users need assurance that the data stream is originated from the correct sender and malicious entities cannot replace parts of the stream with different material. The classical approach to this problem makes use of cryptography (either symmetric or asymmetric). For example in [2] the authors present a solution for the multicast streams authentication based on shared key mechanisms using MAC (Message Authentication Code). Entities acting as sender and receivers use MAC code to authenticate the message through the shared key. In [3], [4] is given a solution based on symmetric cryptography, where only the members of a multicast group can decrypt (and thus access) the messages. These approaches are however inadequate for source authentication, since they don’t distinguish between different senders belonging to the same group (since all nodes share the same key). For this reason research has oriented towards systems based on asymmetric cryptography. A straightforward solution for the video stream multicast based on asymmetric cryptography consists in signing each packet of the stream through a digital signature [5]. This scheme has two main drawbacks: a high computational overhead both at the sender and at the receiver ends, and a high bandwidth overhead due to the length of current digital signatures. Hence the concept of signature amortization has been introduced [6], [7], [8]. It exploits a combination of digital signatures and hash functions to reduce the memory and computational overheads. In its basic scheme (the hash chain), each packet of a given packet sequence carries the authentication information for the successor (the hash of the successor), and only the first packet is authenticated. At the receiver side the hash chain is analyzed starting from the first packet which contains the information to guarantee for the next. In this paper we consider the use of signature amortization techniques within the satellite multicast transmission framework. Our idea is to implement signature amortization hiding the hash chain in the video stream. This approach is independent of the security protocols at the network and MAC layers, and avoids the overhead of security protocols introducing transparent modifications to video pictures. In particular we observe that the features of satellite communications are quite specific with respect to communications in other networks. Multicast satellite networks present characteristics such as like low bit error rate [9] and no packet re-ordering for which simple authentication chains could be very efficient. There are various techniques for embedding data into uncompressed video stream pictures but the frequency-based method has an advantage in terms of number of processing stages that an image may undergo [10]. In particular, spread spectrum watermarking allows to place the mark in perceptually insignificant regions of the image preserving the mark robustness to geometric distortions. The mark is inserted in the frequency domain of the host picture after its redundancy amplification techniques for embedding data into uncompressed video streams over a noisy link.

In our previous works we have presented the authentication procedure [26],[27] and simulation results [26] about mark extraction performances after MPEG-2 video coding, assuming a negligible bit error rate. However, although satellite links have low bit error rate, this rate cannot be completely neglected. For this reason in this paper, we present a novel approach dealing with packet loss. In particular we leverage image hash functions for guaranteeing content authentication in presence of weak picture changes. We present simulation results about authentication performances considering transmission of well known video streams over a noisy link.
RELATED WORK

A. Signature amortization
In [2] authors proposed a solution to multicast communication authentication through a shared key mechanism where each member has a different set of keys. Sender holds a set of l keys and attaches to each packet l MACs, each one computed with a different key. Each recipient holds a subset of the l keys and verifies the MAC according to the keys it holds. In this schema the communication overhead is important and the security is only defined up to a coalition of malicious recipients forging data for a chosen recipient. A different approach to multicast authentication is an efficient and simple scheme that guarantees authenticity, integrity and non repudiation using a digital signature for the first packet and propagating authentication through a hash chain. In [6] authors present a procedure in which the sender signs the first packet, then it ties in subsequent blocks in a way that guarantees the stream authenticity. For a finite and known stream, the sender inserts the hash of each block into the block that precedes it. This solution does not tolerate packet loss, because losses break the chain preventing next authentications. Moreover, this solution applies only if both sender and receiver are able to buffer the entire stream in advance. Various solutions have been proposed for tackling packet loss phenomena, in [13], authors propose an authentication scheme that amortizes a single signature operation over multiple packets encoding the hash values and signatures with IDA (Information Dispersal Algorithm) erasure code. Other solutions are given by redounded chains like in [8], [14], [15].

B. Watermarking
Hartung and Girod [16] proposed methods for embedding additive digital watermarks into uncompressed and compressed video streams. The first method regards inserting sequences of repeated bits (the redounded mark) in the spatial domain of the video image after their amplification and modulation by pseudo noise sequences. They also presented a marking procedure in the MPEG-2 bit stream domain concluding that in this case watermarking scheme is less robust than its counterpart in the pixel domain. Cox et al. [17] proposed spread spectrum watermarking scheme by embedding data in perceptually insignificant DCT (discrete cosine transform) coefficients of the cover image and presented a methodology that can be generalized to audio, video and multimedia data. The basic idea is to spread narrow band watermark signal over many frequency bins of the host image using pseudo random spreading sequences so that the watermark energy content for each bin becomes small and could hardly be detected. At the same time, any attempt to remove watermark causes image impairment to an extent that fails to preserve the acceptable quality of the watermarked image.

C. Features extraction
Data integrity issues are generally addressed by cryptographic hashes, which are sensitive to every bit of the input message. As a result, the message integrity can be validated when every bit of the message is unchanged. The information carried by media data (multimedia content) is mostly retained also after moderate levels of filtering, geometric distortion, or noise corruption. Therefore, bit level verification is no longer a suitable way to authenticate video streaming, and multimedia content authentication by means of feature extraction [28][29][30] is becoming more and more attractive.

SYSTEM MODEL
Multicast satellites networks present some significant security challenges due to their broadcast nature which makes eavesdropping and active intrusion easier than in terrestrial fixed or mobile networks. Satellite systems exhibit low packet loss most of the time, with typical project constraints of 10^{-8} bit error rate 99% of the time [9], except for a few days a year when the atmospheric conditions are so bad that they heavily affect data transmissions.

![Fig. 1: Multicast video streaming over satellite network](image-url)

We consider a broadcast architecture is constituted by a group of satellite stations. Each station can broadcast an MPEG-2 [18] video stream to the other stations in the group (as shown in Fig. 1). To ensure source authentication we assume that each station has its own pair of public/private keys and that it uses the private key to sign its transmissions. In turn the receiving stations authenticate the source and the transmissions using the source public key.

THE AUTHENTICATION PROCEDURE
In this section we present the authentication procedure used for the source authentication of a video stream. The authentication procedure exploits a signature amortization scheme based on single hash chains, because the bit error rates of satellite transmissions is in most cases very low, and the probability that a hash chain gets broken due to packet losses is negligible. It should be observed that in presence of lossy links multiple hash chains could be employed, however we judge their use unnecessary in the considered scenario and, in any case, the extension of the algorithm to include multiple hash chains is straightforward. The au-
The authentication procedure exploits hash functions and signatures. For our purposes, good candidates are the SHA-1 hash function [19] which produces a 160-bit digest from a message with a maximum size of $2^{64}$ bits and a standard RSA algorithm which computes signatures of 1024 bits. Hereafter, we use the functions $H(\cdot)$, $\text{Signature}(\cdot)$ and $\text{VerSignature}(\cdot)$ which compute the SHA-1 hash of a data, the RSA signature of a data and perform the authentication test, respectively.

We also assume the existence of two functions $W(\cdot)$ and $E(\cdot)$ which perform data embedding and data extraction from a video sequence, respectively. In particular, given a sequence $B$ of picture and a value $H$ function $W(B, H)$ computes a video sequence $B'$ which is obtained embedding $H$ into $B$, and function $E(B')$ extracts $H$ from $B'$. Functions $W(\cdot)$ and $E(\cdot)$ are described in Section V.

Transmitter is provided with an uncompressed video stream $S$ and interprets it as a sequence of block pictures $B_1, B_2, \ldots, B_n$. In the single hash chain signature amortization scheme, given a sequence of packets $B_1, B_2, B_3$, each packet is augmented with a witness which is the hash of the successor. Sender starts computing features extraction from the last block of the video sequence as $f_i = F_E(B_i)$. The construction of the hash chain starts from the block of index $n-1$. The hash value $h_i$ is calculated over the extracted features from the previous block $h_i = H(f_i)$. This value $h_i$ is inserted into the block $B_{i+1}$ by means of the watermarking embedding function $B_{i+1} = W(B_i, h_i)$. When the first block is reached source computes the signature $S$ of $H(B_n)$ as $S = \text{Signature}(H(B_n))$. The block sequence is now encoded with $G_i = \text{MPEG2}(B_i)$ and transmitted as $[S, G_1, \ldots, G_n]$.

![Signature amortization using a simple hash chain](image)

Note that in our scheme the witness of each block is embedded in the block itself using a watermarking technique. This is possible because the blocks carry video streams.

### THE EMBEDDING PROCEDURE

#### A. Embedding information as watermark

In this section, we describe the technique used to embed the hash chain into the video stream. In particular, we describe the functions $W(\cdot)$ and $E(\cdot)$ used in the previous section. It should be observed that our use of watermarking techniques is not standard. Classical watermarking is used to embed a logo (generally a picture) within a video stream. This embedding is hidden in the video and it can be extracted to witness the ownership of the video. The extraction is in general lossy, that is, the logo can be extracted with errors, provided it remains recognizable. In our case instead we use watermarking to embed hashes of blocks.

The extraction of the hashes must not be affected by errors, otherwise the hash chain would result broken and the receiver would be unable to authenticate the source of the video stream. For this reason, in this section we propose an embedding technique which ensures the extraction of the hashes without errors, and we evaluate the performance of this technique in the next section. The proposed embedding algorithm is based on spread spectrum watermarking [17], and could potentially exploit Reed-Solomon codes to improve mark extraction performances. A video stream can be considered as a three-dimensional signal whose components are bi-dimensional matrices (the picture) and the time. Each picture is formed by a $w \times h$ matrix of pixels; each pixel expresses a colour which can be represented by three values: $R$ (red), $G$ (green), $B$ (blue) or, equivalently, by $Y$ (luminance) and $U$, $V$ (chrominance) components. In our algorithm we consider the $Y$-U-V representation, and we embed a mark (hash) on each picture of the video stream on its luminance component ($Y$).

We use a Fourier domain method based on the DCT (discrete cosine transform) ensuring the robustness with regard to various type of coding algorithms like MPEG-2 or MPEG-4 [20], [21]. This procedure is based on modifications of the image frequency domain coefficients; it thus have a minimal impact on the whole picture in the spatial domain. Given a block $B_i$ (with $1 \leq i \leq n$), let $H$ be the hash of the next block which should be embedded in $B_i$. In our algorithm $H$ is embedded in each picture of the block using redundant codes.

Note that, due to specific features of the mark extraction which will be discussed in Section V-B the mark must be represented as a sequence of $\{-1, 1\}$. For this reason $H_{\text{AS}}$ (which is a sequence of $\{0, 1\}$) is first trivially converted into a sequence of $\{-1, 1\}$. Then, for each picture $p_i$ (with $1 \leq i \leq k$) in $B_i$, the mark $H_{\text{AS}}$ is embedded in $p_i$ using the following equation:

$$R_i = \text{DCT}(p_i(0))$$  

(1)
\[ P_i^H = P_i^{RF} + [P_i^{RF}] \ast \alpha \ast n_i \ast H_{RS} \]  

Equation 1 computes the DCT transformation of the luminance (Y) component \( P_i \) of \( R_i \). Equation 2 embeds \( H_{RS} \) in the high frequency of \( P_i \), denoted \( P_i^{RF} \), using parameters \( \alpha \), an amplification factor, and \( n_i \), a pseudo-noise sequence which is changed for each picture. In particular \( n_i \) are matrices of binary pseudo noise sequences whose items belong to \{-1, 1\}. The spread sequence \( n_i \ast H_{RS} \) is multiplied by an amplification factor (\( \alpha \)), replicated according to a replication factor, and finally reshaped to a matrix of the picture dimensions. The mark is now scaled by the luminance frequency coefficients, so that it is spread proportionally to the DCT coefficients. Finally, the resulting sequence is summed to the highest frequencies coefficients of the DCT transformed picture \( P_i \). The pseudo code of the embedding algorithm is shown in Table III.

**TABLE III**

Function \( W(s) \) embedding hashes into video stream pictures

- Let \( B \) be a video block belonging to the stream \( S_i \).
- Let \( H \) be the hash to be embedded in \( B \).
- Let \( \alpha \) be an amplification factor.
- Let \( PRSG() \) a pseudo random sequence generator algorithm:
  \[ H_{RS} = \text{ERASUREENCODING}(H) \]
  \[ H_{RS}^{\text{Rep-M}} = [H_{RS}, H_{RS}, H_{RS}, \ldots, H_{RS}] \] // concatenation of the \( H_{RS} \) replicas
  \[ H_{RS}^{\text{Rep-M}} = \text{Reshape to a square matrix} \] // \( H_{RS} \) replicas
  \[ H_{W} = \begin{pmatrix} 0 & 0 \\ 0 & H_{RS}^{\text{Rep-M}} \end{pmatrix} \] // Frequency DCT domain

  for \( i = 1 \) to \( k \)
  
  \( n_i = \text{PRSG}() \)
  \( P_i = \text{DCT}([n_i (Y)]) \)
  \( P_i^{H} = P_i + [P_i] \ast \alpha \ast n_i \ast H_{RS} \)
  \( P_i^{Y}(Y) = \text{DCT}^{-1} \{ P_i^{H}(Y) \} \)

**B. Retrieval of information**

Once the receiver receives a block, it performs the mark extraction. Since the mark has been embedded in each picture of the block, the receiver extracts the mark from each picture. This is necessary because the extraction from a single picture is, in general, affected by errors, thus the mark is obtained combining the information extracted from all the pictures in the block. Given a picture \( p_i \) in an incoming block, the receiver performs a DCT transform over the luminance component (Y) of the picture denoted \( P_i \). Then it extracts the mark from the high frequencies of \( P_i \) (which are the frequencies where it had been embedded). This is achieved using the following equations:

\[ P_i^{RF} \ast n_i = P_i^{RF} \ast n_i + [P_i^{RF}] \ast \alpha \ast n_i \ast H_{RS} \ast n_i \]

\[ P_i^{RF} \ast n_i = P_i^{RF} \ast n_i + [P_i^{RF}] \ast \alpha \ast H_{RS} \ast n_i \]

Let \( H_{RS}(i) \) be the hash replica inside picture \( i \) and \( \text{Red} \) be the number of hash replicas in each picture. The receiver combines the mark extracted from each picture using the following equation:

\[
\sum_{i=1}^{\text{Red}} \sum_{j=1}^{\text{Red}} \left\{ P_i(i) \ast \alpha \ast H_{RS}(i) \right\} + \left\{ P_i(i) \ast \alpha \ast H_{RS}(i) \right\} = \sum_{i=1}^{\text{Red}} \sum_{j=1}^{\text{Red}} \left\{ P_i(i) \ast \alpha \ast H_{RS}(i) \right\}
\]

where the inner sum accumulates the component extracted from each replica within a given picture, and the outer sum accumulates all the marks extracted from all the pictures in the block. Since the first term in Equation 3 \( (P_i(i) \ast n_i(i)) \) expresses uncorrelated signals, their sum tends to 0, thus, if the number of pictures in a block is sufficiently high, Equation 3 becomes:

\[
\sum_{i=1}^{\text{Red}} \sum_{j=1}^{\text{Red}} \left\{ P_i(i) \ast \alpha \ast H_{RS}(i) \right\} = H_{RS}
\]

Thus the mark extraction is performed over 4 as:

\[
H = \text{sign} \left( \sum_{i=1}^{\text{Red}} \sum_{j=1}^{\text{Red}} \left\{ P_i(i) \ast \alpha \ast H_{RS}(i) \right\} \right)
\]

The extracted sequence is now decoded according to Reed-Solomon coding parameters used by the sender. The pseudo code of the extraction algorithm is shown in Table IV.

**RESULTS AND DISCUSSION**

We have performed simulation experiments aimed at evaluating mark extraction performances from a corrupted video stream. In the simulations we considered three well known video samples (Akiyo, Coastguard and Foreman) and we have simulated video streaming over a noisy satellite channel. The raw videos have been coded with MPEG-2 algorithm and then corrupted using a Bernoulli loss process with three different bit error rates: \(5 \times 10^{-6}, 10^{-5}, 5 \times 10^{-5}\). The use of Bernoulli loss process is a common choice [24] for simulating loss process in satellite links and bit error rate less than \(10^{-5}\) are typical for satellite link because of DVB-S project constraints [25]. Fig. 1 shows mark extraction performances of akiyo video sample after MPEG-2 coding process, simulation transmission over a noisy channel and finally MPEG-2 decoding process. Note that, using a GOP length of 5 picture, the maximum number of extraction \(300/5=60\) is reached only using a perfectly reliable channel. Nevertheless bit error rates of \(5 \times 10^{-6}\) and \(10^{-5}\) decrease the performances up to 20%. Fig. 2 and 3 show simulation result for coastguard
and *foreman* video samples respectively. Note that in these cases bit error rates of $5 \times 10^{-4}$ and $10^{-3}$ do not affect the number of mark extractions.

**CONCLUSIONS**

We have presented the performance evaluation of a new authentication procedure over a simulated satellite link. Our authentication algorithm combines signature amortization based on hash chain and a watermarking algorithm for embedding authentication information in the video stream.

We have improved a previous algorithm for dealing with packet loss. We have tested the new procedure simulating the streaming of authenticated well known video samples over a simulated bernoullian noisy link, and we have evaluated authentication performances with different bit error rate using three different video streams.

Future works involves the study of erasure codes in the watermarking algorithm to improve the ability to correctly extract the mark in presence of errors.

**REFERENCES**


