AUTHENTICATION OF MPEG-4-BASED SURVEILLANCE VIDEO

Michael Pramateftakis, Tobias Oelbaum and Klaus Diepold

Institute for Data Processing, Dept. of Electrical Engineering
Technische Universität München
80290 Munich, Germany

ABSTRACT

The industry is currently starting to use MPEG-4 compressed digital video for surveillance applications. The transition from analog to digital video raises difficulties for using surveillance video in court. Since it is fairly easy to make hard-to-detect modifications to the video stream captured by the cameras, e.g. mask out a specific event or person, a system for proving authenticity and integrity of video streams is needed. This paper presents such a system based on digital signatures embedded in the video stream. Our concept provides a means for proving authenticity and integrity of MPEG-4 digital video streams, while leaving compatibility with standard media players untouched.

1. INTRODUCTION

Traditionally, closed-circuit television (CCTV) systems for surveillance have been recording and storing analog video. With the industry progressing towards digital video for surveillance applications, new problems arise when taking digital video streams to court [1]. In the past, it was very difficult to modify analog video in order to e.g. mask out an event or make a person's face unrecognizable, also because of the limited availability and high cost of respective tools. Apart of that, the change would probably be easily noticeable. Today, for digital video, there is a variety of low-cost, easy-to-use tools for making such modifications in a way that the change is not noticeable. Thus, while analog video can be used as proof before court without considerable problems, digital video needs a mechanism that can prove that the video stream presented is indeed the video stream captured by the camera without any modifications.

This paper presents a concept based on digital signatures for proving authenticity and integrity of MPEG-4 digital video streams. The signatures are generated within the camera in real-time during recording. The camera itself is considered a secure environment. The digitally signed video streams remain compatible with standard media players, as authentication information is placed in ancillary data fields within the video stream, a feature already existent in the MPEG-4 standard. A special player can access and verify the authentication information, providing the viewer with an indication of authenticity of the video stream along with additional useful information, e.g. camera identification numbers, timestamps etc. Standard media players can decode the video stream normally, but will not interpret the authentication data. It is expected that only special players which are aware of the presented concept will be used in applications where an authenticity indication is needed, e.g. before court.

2. ABOUT DIGITAL VIDEO AUTHENTICATION

Digital video authentication in the context of surveillance applications proves that any or all of the following facts are true:

A. No frame in the video stream has been modified.
B. No new frame was inserted in the video stream.
C. No frame was removed from the video stream.
D. The video being viewed actually came from the indicated camera.

Figure 1 shows examples of the first three facts. Our proposal for a solution to the fourth fact requires authentication information calculations and encoding to occur within the camera. The obvious alternative of calculating all authentication information at a central server which receives video streams from multiple cameras has the major drawback of opening the way to attacks while the video stream is underway from the camera to the server. Such attacks require access to the network that connects the cameras to the server, which, in CCTV applications, should be adequately protected, but the trend goes toward connecting cameras and central storage through the Internet or a corporate LAN. By using specialized camera chipsets that can calculate and encode authentication information in the video stream immediately after it has been captured, we obtain secured streams that are cryptographically bound to the camera before they are fed to the central storage server. Thus, modification attempts on the streams while they are in

0-7803-8554-3/04/$20.00 ©2004 IEEE.

transit do not make any sense, as the attack would be immediately noticeable at the storage server, which expects correctly signed video.

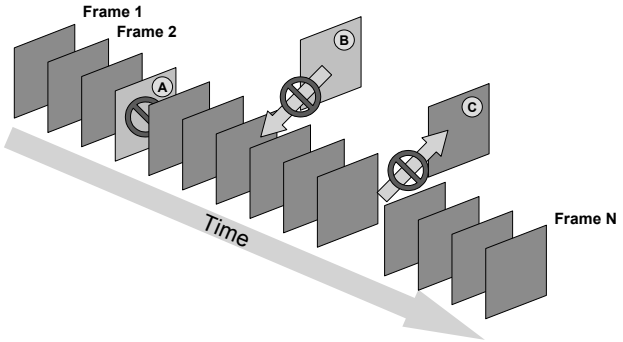

Figure 1: Authentication of digital video ensures that no frame was modified (A), no new frames were inserted into the stream (B) and no frames were removed from it (C).

3. EMBEDDING AUTHENTICATION INFORMATION INTO VIDEO STREAMS

Two ways of embedding external information into a video stream are commonly used: Information is either embedded into special syntax fields defined in the stream format and reserved for this reason, or through a watermark applied to the video frames (see e.g. [7] and [8]). Conceptually, both ways could be used for realizing our system. We chose to avoid using watermarks for several reasons. The steps that leaded us to this decision are outlined below.

3.1. Design aspects considered

Since there is no best way of embedding that is optimal for all applications, the appropriate method can only be deduced by looking at the characteristics of the specific application. For defining the optimal method for our surveillance scenario, we considered various related aspects.

A Boolean indication of authenticity and integrity is required for this application. When using digital video as evidence before court, it must be proven that the video is completely untouched. Effectively, either no modification has happened at all and the video can be used as proof, or the video was modified somehow and is useless before court. This implies that the indication of authenticity is fragile, so that even the slightest modification is detected. Yet, the system should be able to distinguish between modification attacks and eventual transmission errors correctly. Apart of the authenticity indication, additional information like the identifier of the camera that captured the stream and the date and time of recording is important.

Authenticity and integrity must be provable even if only parts of the video stream are taken. This is essential as only small parts (maybe minutes) of a larger recording (maybe a day) will be used in court in most cases.

We propose that the authentication information calculations occur within the camera. The camera is considered a secure environment, i.e. an attacker can not tamper with it, or security personnel will immediately be alerted. Since the embedded systems within the camera have limited processing power and MPEG-4 encoding is a computation-intensive task, the overhead induced by authentication calculations should be constrained to a minimum. This also holds for the overhead induced in the video stream by the physical size of the authentication information. These goals should be achieved without raising the cost of a single camera unit too much.

3.2. Deciding on the embedding method

After considering the above facts, we concluded that digital signatures over the video stream placed in ancillary data fields of the MPEG-4 video stream is the best choice for our scenario. The reasons for this decision are as follows.

The use of digital signatures fulfills our fragility requirement, since a digital signature remains valid only if the data it was calculated over is absolutely untouched. Apart of that, verification of the signature provides the Boolean authenticity indication we require. Another important feature is that, depending on the requirements of a particular implementation, a single digital signature can authenticate a variable time span, since it can be calculated over a variable amount of frames. Thus, if parts of a stream are to be used, a granularity of signatures per time unit can be specified depending on the requirements. So, even small parts of the video can be taken and proven to be authentic over their entire duration.

We chose to avoid watermarks in this concept, even though they would also be a solution. Watermarks change the content of video frames in very subtle ways, such that degradation in video quality is not noticeable when the video is viewed by a human viewer. However, the degradation may be disadvantageous to image processing algorithms that may be applied to the video material. For example, the material may be fed to a pattern-matching system in an attempt to recognize or identify a certain person appearing in the video. Such image processing algorithms require the highest possible image quality to work reliably. By embedding the signatures in the ancillary data fields of the stream, the frame content is left unchanged. Another less important reason is that if we would want to embed digital signatures through watermarks, we would possibly run into capacity

problems, due to the fact that watermarks have mostly limited data capacity per time unit.

An advantage of using watermarks is of course the fact that the mark is bound to the content of the frames and cannot be removed; at least not with the same ease that ancillary data can be. However, since the cameras in a CCTV scenario are connected to a central storage server permanently, the server expects correctly signed video at all times. So, if an attacker tries to remove the signatures from the stream while they are underway to the server, the attack becomes immediately noticeable, since the attacker can not recalculate valid signatures as he doesn't have the camera's private key. The result is that we can keep the simple digital-signature-based system and embedding scheme without the binding problem of authentication information to content.

4. MPEG-4 VIDEO STREAM STRUCTURE

An MPEG-4 video stream can be divided into five hierarchical layers (see Figure 2). The lowest layer is the Video Object Plane (VOP) layer, which normally corresponds to a single frame of the video stream. One or more frames are grouped together in Group of Video Object Planes (GOP) objects. A GOP object has three attributes that are interesting for our system:

- The frames within a GOP object can be decoded independently of frames outside the GOP.
- This is the lowest layer where user data can be added to the stream.
- The time code of the video is stored at this level.

Several GOPs are grouped together in the Video Object Layer (VOL). This layer contains information needed to display the video, such as the size of the video frames in pixels, the frame rate of the video and the color matrices. The two top layers, the Visual Object Sequence Layer and the Visual Object Layer, contain additional information about the type of the MPEG-4 objects, as well as the profile and level indicators of the encoded bit stream. These layers are not important to our system, as they only appear once in each bit stream and do not contain important data fields needed for authenticating the video stream.

5. A CONCEPT FOR VIDEO AUTHENTICATION

Our system provides an indication of authenticity and integrity of an MPEG-4 video stream by embedding digital signatures in the Group of Video Object Planes objects within the stream. In the particular application of CCTV, the signatures are generated within the camera, while video is being recorded, so that the encoded video stream leaving the camera is already signed.

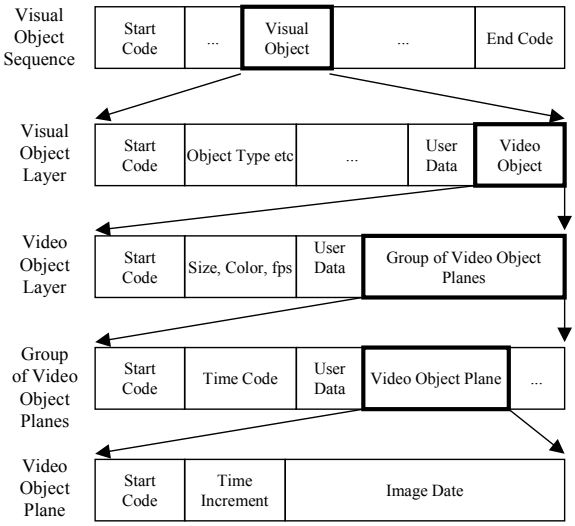

Figure 2: Logical structure of an MPEG-4 video stream. The layers relevant to our concept are the two lowest ones. The lowest layer (Video Object Plane) contains the video frames which must be authenticated and the layer above (Group of Video Object Planes) is the lowest layer that can accommodate user data.

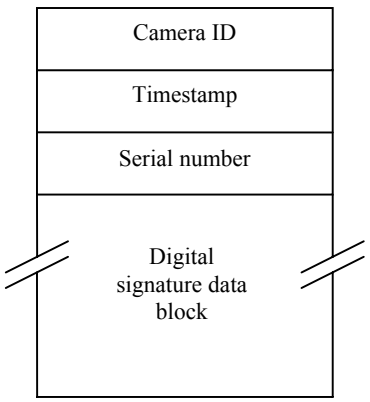

Figure 3: Structure of the authentication information embedded in the stream. Included is a camera identifier, a timestamp, the serial number of the structure and the digital signature.

The embedded information can be easily removed from the stream unless the stream is encrypted. Yet, encryption is not necessary, due to the fact that the storage server in the CCTV system expects correct digital signatures in the stream. The server knows the public keys of all connected cameras and can periodically check for correct signatures. This can be done in a scheme that keeps overhead low and protects the system from signature removal attacks.

An authentication information structure as shown in Figure 3 is embedded in the user data field of each Group

of Video Object Planes object. The structure contains a camera identifier field, a timestamp field, a serial number field and a digital signature block. The camera ID field contains the name of the camera that recorded the video stream and is used for identifying the correct public key for signature verification. The timestamp field contains the date of recording and complements the timestamp within the Group of Video Object Planes object, which contains a frame-accurate time code between 00:00 and 23:59 without date information. The exact date and time where each and every frame was taken can be deduced by the two timestamps, provided that the camera has a clock of acceptable accuracy. The serial number field contains an integer number that can be used to verify that every object is in its original sequence. This provides a means of proving that the sequence of the frames in the video stream has not been tampered with. An alternative method would be digital signatures linked to one another, but this would require unnecessary overhead. Since the number of signatures in a video stream can grow very large, an appropriate length for the serial number field must be selected. In our prototypes we have used a field length of 48 bits (6 bytes), but any length can be used depending on the particular application.

The length of a GOP object in frames is a parameter of the MPEG-4 encoder, so the number of frames per GOP is configurable based on how many signatures per time unit a particular application requires (Remember that there is one digital signature per GOP). We believe that 5 signatures per second are a good tradeoff between stream overhead and shortest video part that can be signed separately, and should be sufficient for any application. In our prototype, we constrained the number of signatures per second to 5 when the video had a frame rate of 5 fps or higher. When the frame rate was lower, the number of signatures per second was reduced accordingly.

Each digital signature value is calculated over the whole GOP object, including the timestamp and serial number fields of the authentication structure. Thus, every piece of information in the object is protected. The digital signature formed over a GOP object is placed in the authentication structure in the user data field of the following GOP. To facilitate easy embedding and verification of the signatures, a single signature calculation extends over the boundaries of a GOP and includes a small part of the next GOP, as Figure 4 shows. In that way, a hash value can be calculated from the end of a signature block to the beginning of the next one. When the beginning of a signature block has been reached, the currently accumulated hash is signed and the signature is placed in the stream. The process starts from the beginning immediately afterwards. A similar procedure is used for verification.

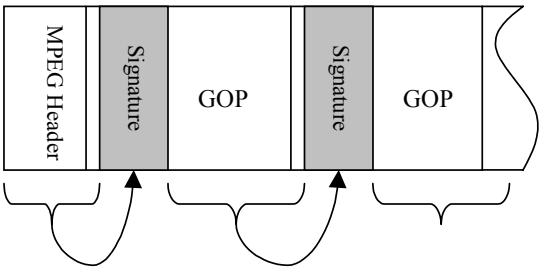

Figure 4: Signature embedding in the stream. The signature calculated over a GOP is placed in the next GOP object's user data field. The first GOP in the stream contains a signature formed over all parts of the stream preceding the GOP, i.e. headers of higher layers.

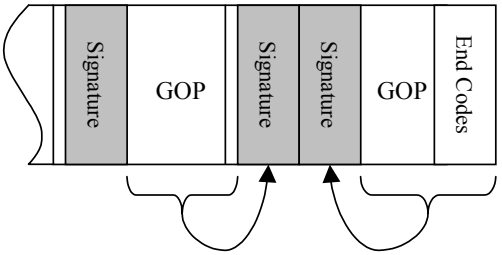

Figure 5: Signature embedding at the end of the stream. The last GOP object contains two signature data blocks: One from the previous GOP and one from its own content. This signals the end of the stream. If any end codes are present, they are included in the final signature.

The first GOP object in the stream contains a signature that was formed over the data of the MPEG-header preceding it (Figure 4). The user data field of the last GOP object in the sequence contains two signatures, one from the preceding GOP object and one from the current GOP object, since there are no others following. The existence of two signature blocks within a single GOP indicates the end of the stream, as shown in Figure 5. In some cases, MPEG end codes may be present after the last GOP in the stream. These codes are not mandatory. In case any are present, they are included in the final digital signature.

6. IMPLEMENTATION ASPECTS

The most important aspects that have to be taken into account for particular implementations of the concept are calculation overhead for the digital signatures and stream overhead from the authentication blocks. In order to simulate a typical embedded system for MPEG-4 encoding in cameras, e.g. the MPEG-specialized MAP-CA DSP by Equator Technologies, Inc. [3] running at 400 MHz, we implemented our concept on a Pentium-III CPU

running at 850 MHz. In order to have a model of typical MPEG-4 video streams used in surveillance applications, we used video streams with a bit rate of 512 kbit/s at 12 fps. The bit rate is high enough for MPEG-4 to produce high quality video and the frame rate of 12 fps is a good value for surveillance applications. Real application frame rates are mostly considerably lower.

For our prototype we stated the following requirements: First, the load on the processor for digital signature calculations should not exceed 12.5% (i.e. one eight of the whole load). This is what we consider acceptable for a CPU which must accomplish MPEG-4 encoding in real time. Second, stream overhead should not exceed 10% of the bandwidth. With the above parameters of 512 kbit/s for the video data, we have up to 51 kbit/s for authentication information at our disposal.

We used the MD-5 algorithm for hashing and the RSA algorithm with keys of 512 bit length for digital signatures. The length of the GOP objects was set to 5 frames, which led to 2 signatures per second at the frame rate of 12 fps (approx. 2 GOPs per second).

Our prototype reached our goals satisfactorily. The processor load for signature calculations was only 3%, which is well below our limit. Yet, a test with keys of 1024 bits length raised the load to over 20%. For real applications, a specialized cryptographic processor should be used. The stream overhead is particularly low. With a key length of 512 bits, the signature is 64 bytes long. The whole authentication structure is thus in our configuration about 80 bytes long. At 2 signatures per second, we obtain an overhead of 1.2 kbit/s, which is well under the specified limit. Even with much higher key lengths, stream overhead is still not a problem.

7. CONCLUSION

Our concept provides a simple and effective solution to the problem of authentic digital video. In cases where digital video streams have to be used as proof before court, this system is a great benefit. Its simplicity and performance makes it an attractive solution for real implementations.

8. REFERENCES

[1] House of Lords, “Digital Images as Evidence, Government Response”, Select Comittee on Science and Technology, Eighth Report, June 1998.

[2] ISO/IEC International Standard 14496-2:2001.

[3] http://www.equator.com

[4] R. Gennaro and P. Rohatgi, “How to Sign Digital Streams”, Proceedings of Crypto’97.

[5] S. Haber and W.S. Stornetta, “How to Timestamp a Digital Document”, Journal of Cryptology (2) 3 (1991), pp. 99-111.

[6] P.C. Van Oorschot, S.A. Vanstone and A. Menezes, *Handbook of Applied Cryptography*, CRC Press, 1996.

[7] D. Nicholson, P. Kudumakis and J.F. Delaigle, “Watermarking in the MPEG4 context”, ECMAST’99, Proceedings of the European Conference on Multimedia Applications Services and Techniques, Madrid, Spain, May 1999, pp. 472-492.

[8] Stefan Katzenbeisser and Fabien A.P.Petitcolas, *Information Hiding techniques for steganography and digital watermarking*, Artech House, 2000