Signal Processing: *Image Communication* 24 (2009) 825–833

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

A secure media streaming mechanism combining encryption, authentication, and transcoding

Luntian Mou [a,b], Tiejun Huang [c,*], Longshe Huo [c], Weiping Li [c], Wen Gao [c], Xilin Chen [a]

[a] *Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China*
[b] *Graduate University of Chinese Academy of Sciences, Beijing 100039, China*
[c] *Institute of Digital Media, Peking University, Beijing 100871, China*

ARTICLE INFO

Article history:
Received 1 June 2008
Received in revised form
9 November 2008
Accepted 3 March 2009

Keywords:
Media streaming
Media-aware encryption
Sender authentication
Secure transcoding
Intelligent mid-network proxy

ABSTRACT

Technology advancements are allowing more and more new media applications and services to be delivered over the Internet. Many of these applications and services require flexibility in media distribution as well as security in protecting the confidentiality of media content and ensuring its authenticity. However, achieving flexibility and achieving security are conventionally conflicting with each other. This is mainly because that security is traditionally implemented in a media-unaware manner that naturally prevents flexible handling of the media content during its distribution. This paper shows that the two goals can be simultaneously accomplished by making an overall plan and taking into consideration all the factors: coding, encryption, packetization, authentication, and transcoding. It emphasizes sender authentication, which is a big security issue for media streaming, but somehow left unresolved. We propose a secure media streaming mechanism which supports media-aware encryption, sender authentication and secure transcoding. Multimedia Application Routing Server (MARS) is especially used as an intelligent mid-network proxy with the ability to perform tasks of sender authentication and secure transcoding. A prototype system for securely streaming AVS media using MARS demonstrates the performance, thus proves the practicality of the proposed secure media streaming mechanism.

1. Introduction

In recent years, the rapid development of the Internet and media coding has made various media streaming applications and services available on the Internet. The typical media streaming applications include IPTV, video conferencing and distance education. Many of these applications and services require flexibility in media distribution as well as security in protecting the confidentiality of media content and ensuring its authenticity. Flexibility means that flexible media handling and processing should be allowed at various stages in the media distribution chain. It is because that the available bandwidth for streaming a particular media at a point in time is decided by the heterogonous network it goes through and the total traffic over the network at that moment. Moreover, different terminals may vary greatly in the capability of receiving and processing streaming media. Therefore, media adaptation, especially video adaptation, is needed both at a sender and at any mid-network proxies so as to provide adaptive quality of service (QoS) for users. Scalable coding is designed to meet the flexibility requirement and seems most suitable for media streaming services over the Internet. But for the reason of performance, scalable coding techniques are yet to be deployed in today's mainstream codec. Although non-scalable coding is comparatively less flexible, it still provides a certain degree of scalability, for example, the frame layer scalability or temporal scalability. This temporal scalability can be best exploited so that a simple

* Corresponding author.
E-mail address: tjhuang@pku.edu.cn (T. Huang).

0923-5965/$ - see front matter © 2009 Published by Elsevier B.V.
doi:10.1016/j.image.2009.03.001

but effective media adaptation strategy can be used. For security needed by media streaming applications, it includes encryption and authentication, which meet the three information security requirements of confidentiality, authenticity and non-repudability. Confidentiality is well achieved in the way that media content is encrypted and kept in ciphertext during its distribution, thus only authorized users can consume the protected content. Authenticity means that the media content received is trustworthy in terms of its integrity and alleged sender. Non-repudability ensures that the sender cannot deny the action of sending the media. Authenticity and non-repudability can be achieved simultaneously by employing the technique of digital signature.

However, achieving flexibility and achieving security are conventionally conflicting with each other. On the one hand, security is traditionally implemented in a media-unaware manner that naturally prevents flexible handling of the media content during its distribution. For instance, media encryption is often done in the same way as data encryption that makes some valuable structural information of the media completely unavailable during its distribution. This media-unaware protection can satisfy the requirement of end-to-end security, but it denies any flexible handling and processing to the media during its delivery. On the other hand, flexible handling to the media often compromises the security. For example, transcoding the encrypted media at a mid-network node usually needs to decrypt the media first, which leaves the media in plaintext and opens an obvious security hole for attackers.

In this paper, we show that flexibility and security can be simultaneously achieved by using media-aware protection. For a non-scalable coder, the structural information of its output bitstream such as group of pictures (GOP) and picture start code can be used for video adaptation to achieve network adaptive or terminal adaptive media streaming. In other words, adaptation-friendliness is achieved by encrypting actual media content instead of the entire bitstream. Based on such structural information, a selective frame-dropping algorithm can be adopted as a media adaptation technique by a sender. According to specific security requirements, all frames or selectively some frames are encrypted, with a signal indicating whether a frame is encrypted or not. And each encrypted frame should be able to be decrypted independently. Then an encrypted frame is encapsulated into RTP [1] packets, with a few bytes being inserted right after each RTP header to signal the encryption as well as other digital rights management (DRM) information. After packetization, each RTP packet with protected content is authenticated by a digital signature algorithm. As a result, a fixed length signature together with a unique identifier of the stream is appended to each RTP packet as its authentication tag. An intelligent mid-network proxy can easily decide to discard or pass on a RTP packet by verifying its digital signature. Therefore, attacks to the media streaming can be detected, and no tampered or illegal media packets have the chance of prevailing over the Internet and reaching end users.

We design a secure media streaming mechanism by making use of the existing highly studied cryptographic techniques because of their proved security, both theoretically and practically. The novelty in the mechanism is that these cryptographic techniques are used in new applications of media streaming, and in a different manner than they are typically used. What is more, encryption, authentication, and transcoding are so coherently designed that flexibility and security can be simultaneously achieved. We stress that sender authentication is critical for both monitoring media content streamed over the Internet and preventing any illegal or evil media content from proliferating.

The rest of the paper is organized as follows. Section 2 reviews related works. Section 3 proposes a secure media streaming mechanism. In Section 4, we bring up a prototype system using Multimedia Application Routing Server (MARS) as an intelligent mid-network proxy for AVS [2] streaming. Section 5 gives out experimental results to demonstrate the practicability of the proposed secure media streaming mechanism. Finally, the paper is completed with some conclusions in Section 6 and an acknowledgement in Section 7.

2. Related works

In the research field of media streaming, some research works focus on the flexibility requirement, some address the security requirement, and only a few jointly consider both.

2.1. Adaptive streaming

Video adaptation is the most important aspect in flexible media streaming. Various coding strategies with corresponding streaming mechanisms address the problem of serving heterogeneous clients with adaptive video quality. Simulcast [3,4] is a widely used method for video adaptation. A single video source is encoded into multiple independent streams, each with different bitrate and quality suitable for a set of clients. A client can select one of them according to its access bandwidth. However, simulcast is not suitable to be used at mid-network nodes, and can only switch between several different video streams with different bit-rates. What is more, in case of secure streaming, it needs to encrypt the multi-copies of the same content, which implies a waste of computing and storage resources. Scalable coding [5,6] could be the most promising coding scheme for media streaming over the Internet. In MPEG-2 and -4, several layered scalability techniques, namely, SNR scalability, temporal scalability, and spatial scalability, have been included. In such a layered scalable coding technique, a video sequence is coded into a base layer and several enhancement layers. Video adaptation is achieved by adding or dropping layers from the stream. Yet, scalable coding techniques are still not in widespread use. One main reason is that for a few targeted bit-rates, coding individual streams yields better quality than coding multiple layers. Another approach is multiple description coding (MDC) [7], in which a video is encoded in two or more independently decodable layers. The decoded video quality is proportional to the number

of layers decoded. Transcoding [8] is also suggested to be performed at a video server or some mid-network nodes to convert a video to an appropriate format, with expected quality, form and rate. While the expensive computational cost it incurs may not be acceptable to certain real-time applications, a simplified version of transcoding, namely, selective frame-dropping can be used as a simple but effective technique for video adaptation. A network adapted selective frame-dropping algorithm for media streaming is proposed in [9]. The paper aims to address the problem of random packet losses resulted from network bandwidth mismatching. The basic idea of the algorithm is that to determine a sending window for each GOP according to its rendering time interval, and then selectively drop some frames with low priority to ensure that other more important frames in this GOP can be reliably delivered to the receiver within the time limit of the GOP's sending window. For non-scalable coders, such an algorithm is quite suitable.

2.2. Secure streaming

Secure RTP (SRTP) [10] was developed to address the security of RTP flows that provides confidentiality, message authentication and replay protection for RTP traffic, as well as for its associated control traffic real-time transfer control protocol (RTCP) [1]. SRTP provides the basic security services required for secure streaming between a sender and a receiver, but does not provide the ability to securely adapt the protected media, and it leaves sender authentication unresolved for consideration of computation cost and bandwidth overhead.

Internet Streaming Media Alliance (ISMA) [11] has opted for an end-to-end security model for media streaming over potentially lossy channel and through untrusted intermediaries. For untrusted intermediaries to be able to distribute media, the formats standardized are such that sufficient metadata is available at all stages so that delivery can be done without accessing actual media content. In other words, encryption is done at the content level rather than at the transport level, making the protection transport-independent and therefore end-to-end secure. However, it only provides integrity authentication, and does not solve sender authentication.

MS MAF [12] is a MPEG standard aiming at standardizing the format for distribution of governed media content to protect rights of holders and solve the interoperability issue that is worsened by the many existing proprietary DRM systems. Security is achieved by delivering encrypted content and performing mutual authentication between devices involved and integrity authentication of governed content. It supports secure distribution of user-generated content as well as enterprise content. However, adaptation and other flexible handlings of multimedia content are out of the scope of the standard.

2.3. Secure and adaptive streaming

Several approaches [13,14] were proposed to avoid decryption of protected media content at mid-network nodes. Particularly, Wee and Apostolopoulos [13] proposed a secure scalable streaming (SSS) framework that supports end-to-end delivery of encrypted media content while enabling adaptive streaming and transcoding to be performed at intermediate, possibly untrusted, nodes without requiring decryption and therefore preserving the end-to-end security. However, the SSS framework does not specify in detail how to ensure the sender authenticity, which may imply vulnerability to malicious attacks such as insertion of illegal packets or complete replacement of the stream with another fake stream.

In the sense that sender authentication should be done using proved cryptographic services instead of creating new techniques or protocols, to our knowledge, very few papers address the issue of sender authentication in media streaming. Nevertheless, sender authentication is very critical to the security of media streaming. Without sender authentication, end users may be endangered by the risk of consuming illegal contents that certainly harms the rights and interests of end users.

Obviously, by using digital signature algorithms, sender authentication can be achieved as well as the integrity of the content. The reason why digital signature techniques were not chosen for sender authentication is mainly because of worry about their computational costs. Public-key cryptographic algorithms traditionally imply high computational complexity. However, we have noticed that the implementation has already achieved great advancements in recent years. Due to its intrinsic security, public-key cryptography is gaining more and more applications. A hot topic is elliptic curve cryptography (ECC) [15], which is remarkable for its high security. Elliptic curve cryptosystems offer the highest strength-per-key-bit of any known public-key system. With a 160-bit modulus, an elliptic curve system offers the same level of cryptographic security as DSA or RSA with 1024-bit moduli. The smaller key sizes result in smaller system parameters, smaller public-key certificates, bandwidth savings, faster implementations, lower power requirements, and smaller hardware processors [16]. Therefore, we have exploited ECC digital signature algorithms to fulfill the task of sender authentication in our secure media streaming system.

3. A secure media streaming mechanism

It is difficult for media streaming services over the Internet to provide QoS guarantees because that the bandwidth requirements of media streaming services cannot be guaranteed due to time-varying traffic over the Internet. Most routers are QoS-enabled, but QoS is not used at all. The reason is that routers cannot set priority of traffic. Instead, applications are media-aware and can set priority of media data. However, applications are not trusted by routers. As a result, real-time services such as audio and video streaming services are treated the same as data services by routers. Apparently, an interface between applications and routers is needed. Intelligent mid-network proxy can serve as such an interface. It should understand media protocols and be able to process

media data if needed. It also should be able to set priority of media data and its priority setting should be trusted by routers. Therefore, by the introduction of intelligent mid-network proxies into the Internet, together with usage of video adaptation techniques, QoS for media streaming services can be provided. By deployment of intelligent mid-network proxies into the Internet, an application layer multicast can be set up among mid-network proxies that will certainly save up much bandwidth resources.

Another reason for the introduction of intelligent mid-network proxies is that we have extended them with security features so that we can use them for sender authentication. Sender authentication is a key feature of our secure media streaming mechanism. Why should we focus on sender authentication? The reason is that sender authentication could be an efficient way to achieve content governance in the context of media streaming over the Internet. Streaming media can be easily distributed and, by P2P-like distribution techniques, it can reach a large number of users rapidly. Therefore, streaming media is highly proliferating and poses a severe security challenge to content governance. The failure in preventing an illegal or evil streaming media from proliferating could lead to very serious consequences. For example, a forged speech of a statesman may affect the stability of the society, while a piece of pornographic video can do harm to the spiritual health of teenagers.

Traditionally, content governance on the Internet is manually done by public security authorities, which is very inefficient and slow in response. And new techniques based on content analysis are not yet applicable. Therefore, we decide to implement sender authentication based on digital signature techniques that are widely used in e-Government and e-Commerce, hence, with proved security. The reason why digital signature techniques are not chosen by ISMA or SRTP for sender authentication is that the computational cost for signature generation and verification might be expensive and, the bandwidth overhead for delivering signatures could be high. It is true for their streaming models, in which intermediates are taken as untrusted and clients or end users should do signature verification themselves, which may be unaffordable for them. However, the situation is quite different in our model. Here, sender authentication is done by powerful mid-network proxies, instead of being performed by clients. For alleviating overhead of signatures, we choose a proper cryptographic algorithm that produces a signature of acceptable length.

An important goal of our secure media streaming mechanism is to establish a secure network over the Internet. This secure network is made up of trusted intelligent mid-network proxies. Trust between any two intelligent mid-network proxies can be achieved by mutual authentication. Mutual authentication is performed by exchanging and verifying certificates, which are issued by a certain certificate authority (CA) [17]. Certainly, each mid-network proxy in this secure network needs to hold the root certificate of the CA and a certain amount of certificates issued by the CA to identify legal senders. Here, we make a presumption that a sender is legal only if the sender has a certificate issued by the CA. Yet, to a specific mid-network proxy, only those senders whose certificates are stored locally on the proxy (be done by a previous step before streaming service) will be treated as legal senders. Once this secure network is established, every RTP packet, carrying protected or unprotected media, should be authenticated before transmission using a digital signature algorithm and the sender's private key [18]. The signature and an identifier of the stream, i.e., a globally unique stream identifier (StreamId), are attached to the end of a RTP packet as an authentication tag. Here, the StreamId is assigned by a mid-network proxy when a RTP session is being established and the sender has sent its certificate to the mid-network proxy. Therefore, a specific StreamId is associated with a particular certificate of a sender (compared with CertID of a certificate, StreamId is much more compact). When a RTP packet arrives at a mid-network proxy, the mid-network proxy will find the sender's certificate as indicated by authentication tag, and then verify the RTP packet's signature using the sender's public key. If the certificate cannot be found or the verification fails, the RTP packet will be discarded immediately. While the signature is valid, the mid-network proxy will decide whether to pass on the packet by performing a media adaptation algorithm, which may be a selective packet-dropping algorithm based on the unencrypted information of RTP payload. Therefore, RTP packets with fake or no signature have no chance of being transmitted on the secure network. Obviously, the more intelligent mid-network proxies deployed, the better security can be achieved. Cost for deploying intelligent mid-network proxies could be a consideration, but as analyzed above, at least three arguments are in favor of the worthiness of deploying such a secure network. First, intelligent mid-network proxies can help routers to resolve the QoS issue. Second, intelligent mid-network proxies are capable of many flexible media processes. Third, intelligent mid-network proxies are efficient at content-security governance. Of course, in practice, the secure network is preferably to be built over public-key infrastructure (PKI) [17] that can make the media streaming system more secure and trustworthy. But without PKI, the secure network can still accomplish the task of secure media streaming.

Senders, typically service providers, prefer that the content transmitted be kept in ciphertext until it reaches intended receivers. In order to avoid decrypting the protected content when transcoding is performed at intelligent mid-network proxies, we use a media-aware encryption scheme similar to that of ISMA. That is, we encrypt the media content frame by frame, using a selective encryption algorithm when necessary. A stream cipher or block cipher in counter mode (CTR) [18] is used, so that each encrypted media frame can be independently decrypted. Frame level encryption has some advantages. The first is that the encryption is transport-independent. Since encryption is done to the actual content, the media content only has to be encrypted once, and the encrypted version can be adapted to many transport protocols. The second is that structural information about the media is not encrypted and can be used by transcoding process. For example, if only conversion of bitrate from high to low is

the task of transcoding, then it can be done by simply dropping frames according to their frame types judged from the unencrypted structural information. Therefore, transcoding is securely performed without the decryption of protected media content.

For media streaming, key distribution is another challenge. Typically, a key management system (KMS) should be scalable and based on multi-level keys. We will not give further discussion on KMS, since it is very practical and application dependent.

Fig. 1 illustrates the proposed secure media streaming mechanism. The key processes through a sender, via multiple mid-network proxies, to a receiver are as follows.

(1) *Transcode (at a sender)*: transcode a media stream from a high bitrate to a low bitrate to meet the access bandwidth constraint of a certain receiver. It is done by dropping some less important media frames, for example, *P* frames in the bitstream. Since the bitstream is in plaintext at this phase, frame dropping is an easy task.
(2) *Encrypt*: encrypt each frame or some frames in the bitstream using a stream cipher or a block cipher in CTR mode, for example, AES in CTR mode, with a symmetric key. For overview on cryptography, refer to [18]. As discussed above, the structural information of the bitstream is not encrypted, so it can be used by later process of packetization. Encryption can also be done after packetization, by encrypting the payload of each RTP packet. See SRTP for an example in [10].
(3) *Packetize*: packetize an encrypted media frame as well as some valuable structural information into one or more RTP packets using corresponding packetization rules and protection signaling rules specified for a certain media type.
(4) *Authenticate*: sign the entire RTP payload using a digital signature algorithm, for example, ECDSA [19], with the sender's private key; then, append the authentication tag, i.e., the signature and the StreamId, to the end of the RTP packet. We only authenticate the payload of a RTP packet because that the RTP header might be modified during the transmission and we only want to ensure the authenticity of the media content itself. The payload is composed of unencrypted DRM information and media structural information such as picture start code, and encrypted media content. By authenticating the entire RTP payload, the integrity of the unencrypted information is also ensured, and any tampering operations done to the unencrypted information can be detected by verifying the signature.
(5) *Verify*: verify the signature of a RTP packet using the same digital signature algorithm as used by the sender, with the sender's public key in its certificate stored locally on the mid-network proxy. By signature verification, both the integrity and sender authenticity of the RTP payload can be ensured. On mid-network proxies, verification is done first to guarantee that transcoding is only done on authenticated media content.
(6) *Transcode (at a mid-network proxy)*: transcode a media stream from a high bitrate to a low bitrate according to feedback on consumable bitrate from the receiver. Transcoding is securely performed without the decryption of protected media content. It is done by simply dropping frames or all RTP packets for some frames according to a selective frame-dropping algorithm that is mainly based on the well-known knowledge of the different importance of different media frames. Note that, provided that the mid-network proxy is a trusted node of the secure network that we discussed above, transcoding can also be done in the conventional way. That is, transcoding will be

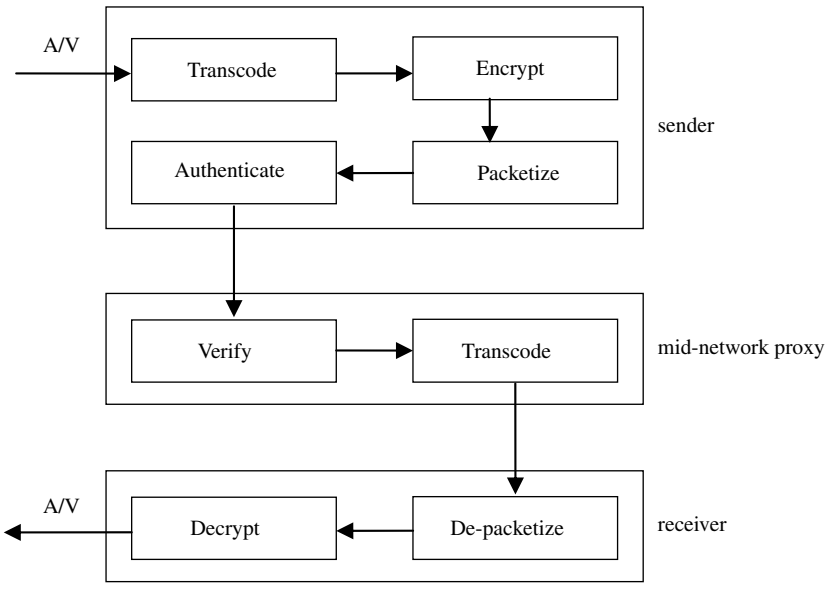

Fig. 1. Illustration of secure media streaming.

performed with the process of decrypt-transcode-encrypt. Consequently, re-packetization and re-authentication will be performed since media data has been changed after the transcoding, which means extra computational cost to mid-network proxies. So, whether this kind of transcoding is applicable depends much on the performance of mid-network proxies.

(7) *De-packetize*: de-packetize a RTP packet according to corresponding packetization rule. Since the RTP packet is from the mid-network proxy, the authenticity of this RTP packet can be trusted and the authentication tag can be simply ignored. But, if the computational power of the receiver is strong enough, it can also do verification itself.

(8) *Decrypt*: decrypt the content using the same stream cipher or block cipher in CTR mode and the same key as used by the encryption process. Preferably, decryption is done only by the destination receiver of the media as demanded by the end-to-end security requirement. But as discussed above, mid-network proxies are trusted devices, so decryption can also be done on mid-network proxies for the purpose of transcoding.

Details about the mechanism and the performance of sender authentication can be found in next section, where we describe a secure media streaming system based on MARS.

4. MARS P2P: a secure media streaming system for AVS

MARS is a hardware infrastructure device managed together with routers. MARS is powerful since it employs DSP chips for computing. It is capable of various media processing such as transcoding between media formats, automatic bandwidth detection, and transcoding from high to low bitrate, automatic determination of terminal capability and transcoding from large to small picture size, as well as region of interest video splitting in bitstream domain for split-screen video display. MARS is media-aware, so it is able to set priority and its priority setting can be trusted by routers. By extending it with features of authentication and secure transcoding, we turn MARS into the desirable intelligent mid-network proxy as discussed in previous sections. Distributed MARS units form a P2P network, with each MARS serving multiple receivers. Like a common P2P system, the performance of MARS P2P gets increasingly better with the growth of its scale. Besides its intrinsic scalability for large-scale deployment, MARS P2P now provides the important feature of security, namely, authentication and secure transcoding. Although both legal and illegal senders (without a qualifying certificate issued by CA) can stream media onto the Internet, only streaming media from legal senders has the possibility to pass the authentication enforced by MARS. Therefore, the MARS P2P network is actually a secure network over the Internet. Obviously, the more MARS deployed, the broader range of secure media streaming can be achieved over the Internet. The topology of MARS P2P is demonstrated in Fig. 2.

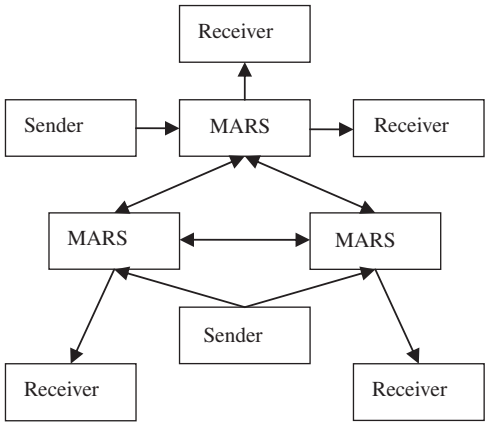

Fig. 2. Topology of MARS P2P.

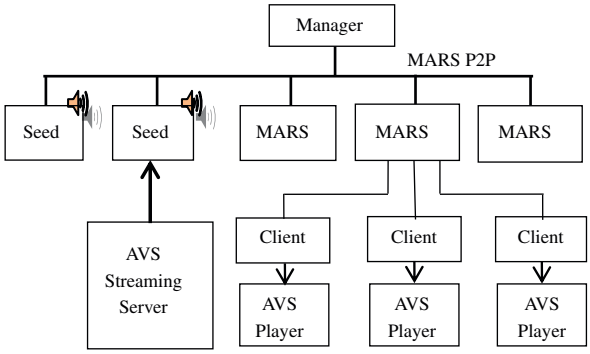

Fig. 3. A MARS P2P secure streaming system.

Fig. 3 illustrates the architecture of a MARS P2P secure media streaming system based on the mechanism proposed in Section 3. It is a P2P system for providing secure AVS [2] audio and video streaming services over the Internet. MARS P2P is much different from other ordinary P2P system in that it has very strong security features. That is, it can provide both content protection and sender authentication.

There are six types of nodes in MARS P2P. Their roles and main functionalities are described below.

(1) *AVS streaming server*: streams encrypted or unencrypted AVS audio and video, from a real-time AVS encoder or a file (see AVS Part 9: AVS File Format [20]). Transcoding can be done at this server before packetization of the media for streaming.

(2) *Seed*: publishes AVS streams as program channels or cancels a channel; authenticate each channel by computing digital signature for each received RTP packet from a server using its own private key. Then the signed RTP packets are delivered over MARS P2P. AVS streaming server, together with Seed, is equivalent to the sender node in Fig. 2 from a functional perspective.

(3) *Manager*: a server that manages all configuration information of MARS units and Seed nodes, information of media streams, information of routing among

MARS units, and Seed nodes as well as routing topology for each stream.

(4) *MARS*: relays streams; manage all the clients associated to it; verify digital signature for each received RTP packet and pass on legal RTP packets to a downstream MARS or its clients. Transcoding can also be done on MARS if necessary.
(5) *Client*: requests a media stream from its home-MARS directly associated with it; when media data is available, passes it to media player.
(6) *AVS player*: depacketizes RTP packets; reconstructs a media frame; decrypts the media frame if it is encrypted; decodes and renders the frame.

While there are various processes in the MARS P2P system, we only want to address three key points.

The first is encryption and packetization. In order to provide flexibility for packetization at the streaming server and transcoding at MARS, we adopt an encryption scheme similar to that of ISMA security standard [12]. That is, we apply AES in CTR mode to encrypt Access Units (AUs), i.e. a video or audio frame, instead of encrypting the whole media. In this way, structural information such as start code for a video sequence or a picture can be preserved in plaintext and used for later packetization and transcoding. The RTP format for content protection is shown below in Fig. 4.

RTP header is conformant to RFC 3550 [1], while AVS RTP DRM header is defined as follows (note that all the numbers in Figs. 5 and 6 are of bit measurement).

The first two fields of HeadLen and DataIsEncrypted are mandatory, while the following five fields are there only when DataIsEncrypted has a value of 1, which means the payload of media data is encrypted. For more information on the AVS RTP DRM header or key derivation algorithm, please refer to AVS DRM [21]. Media data is

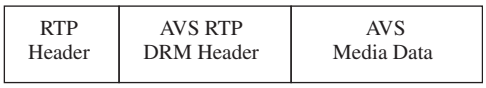

Fig. 4. AVS RTP payload format for protected content.

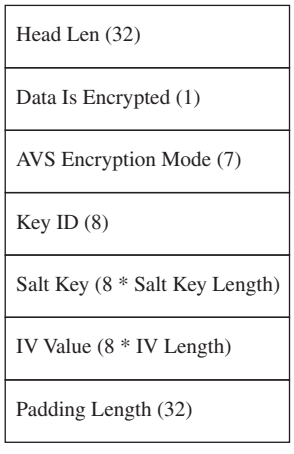

Fig. 5. AVS RTP DRM header.

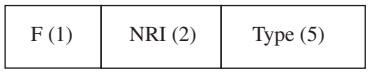

Fig. 6. NALU header.

organized into Network Abstraction Layer Units (NALUs) when it is packetized for transmission over IP network [22]. Each NALU is made up of a one-byte NALU header followed by a piece of media data, for example, a picture header or a slice. The format of NALU header is illustrated by Fig. 6. F is a forbidden zero bit, which should always be set to 0. NRI value indicates relative transmission priority. media aware network element (MANE) such as MARS can use this information to make better performance in protecting important NALUs. So, all NALU headers should be kept unencrypted even when media content is encrypted. The priority values are 11b, 10b, 01b, and 00b in a high to low order. When NALU is of sequence header or I frame, it would be appropriate that its NRI value is 11b. NAL unit type (Type) is a 5-bit unsigned integer that gives out the type of data structure in a NAL unit according to the start code value followed and (or) information contained in the picture header. This implies that the start code value should also be in plaintext.

The second is transcoding. MARS performs transcoding according to its dynamic detection of bandwidth and terminal capability. Judging from the NRI value and Type value in a NALU header, MARS will perform transcoding simply by dropping all the RTP packets for some frames if necessary, without decryption of the protected media content. Transcoding is also securely done on streaming servers using the network adapted selective frame-dropping algorithm [9]. With encryption done at frame level and authentication done at RTP packet level, transcoding actually has no impact on the decryption of any frame received at a client and on the authentication of any RTP packet received by a MARS. And because transcoding is done in a simple and efficient way, it will not affect the real-time characteristics of the media streaming services.

The third is sender authentication. We transplant the open source software implementation of ECDSA from OpenSSL [23]. Authentication, or signature generation, is done on Seed by a PC, while signature verification is done in DSP on MARS. The processes for signature generation on Seed and signature verification on MARS are briefly described below.

(1) *Signature generation on Seed*: receives a RTP packet from a streaming server; signs the payload of the RTP packet using the Seed's 192-bit private key and produces a 48-byte (384 bit) signature; appends the signature and a 4-byte StreamId to the RTP packet as an authentication tag and pass on the RTP packet. See Fig. 7.
(2) *Signature verification on MARS*: receives a RTP packet from a Seed or an upstream MARS; finds the certificate of the Seed using the StreamId contained in the authentication tag; verifies the signature associated with it using the public key of the Seed, on which this RTP packet is signed; drops the packet if its signature

Fig. 7. A RTP packet with authentication tag.

is verified as invalid and gives out alarming signals according to certain predefined policies; if the signature is verified as valid, the RTP packet is delivered to the transcoding process that will decide whether to discard or pass on the RTP packet by performing a packet-dropping algorithm.

The cryptographic details of key generation, signature generation, and signature verification procedures for ECDSA can be found in [19].

Experimental results for the performance of authentication, encryption as well as secure transcoding in this MARS P2P system are given out and analyzed in the following section.

5. Experimental results

Experiments show that signature generation can be done about 1000 times per second and signature verification can be done 50 times per second per DSP. In case that average RTP payload size is about 1024 bytes (8 kb) for video streaming, a single DSP can verify 50 such packets in one second. That is, a single DSP can verify a video stream with an approximate bitrate of 400 kb (50*8 kb = 400 kb). The verification power of MARS is proportional to the number of DSP chips incorporated in it. For the lowest configuration of 4 DSP chips, 4 video streams of 400 kb can be simultaneously verified by a single MARS unit; while for a MARS containing 29 DSP, more than 20 video streams can be verified. Since signature generation is less complex than signature verification for ECC, a PC can sign 20 video streams with the same bitrate of 400 kb. (See Table 1 below).

The system configurations for MARS and PC are concisely described in Table 2.

As bandwidth overhead for signatures (48 bytes each) and StreamIds (4 bytes each) is concerned, it should be less than 5% (see the calculation below).

$$(48+4)/(1024+48+4)100\% < 5\%$$

We decrease the overhead by deliberately sending RTP packets with payload size larger than 1024 bytes, while maintaining the 1500 bytes MTU constraint of IP packet. So, less bandwidth overhead for authentication has been achieved. For example, the calculation could be as follows:

$$(48+4)/(1400)100\% \approx 3.7\%$$

By dedicatedly designing an illegal Seed by turning its authentication feature off, we find that a video stream distributed as a program channel by the Seed is not played on the client requesting it while another stream from legal Seed is normally played. In fact, the entire stream from the illegal Seed has been discarded by the first MARS that this stream reaches on MARS P2P network.

Table 1
Performance of authentication.

Signature generation (on PC)	1000 times/s	20 video streams (400 kb)	
Signature verification (on MARS)	50 times/s (1 DSP)	1 video stream (400 kb) (1 DSP)	4 video streams (400 kb) (4 DSP)

Table 2
System configurations for MARS and PC.

System configuration	MARS	PC
CPU	Motorola powerPC 200 MHz	Intel pentium D CPU 2.8 GHz
OS	Linux 2.4	Windows XP professional
DSP	TMS320C6416	

Therefore, the goal of preventing illegal contents from being consumed by end users is realized. Meanwhile, from the quality and fluency of the playback of protected and authenticated video streams with bit-rates of about 500 kb, we find that our secure mechanism can meet the real-time requirement of media streaming. Tests also show that secure transcoding feature works well when bitrate of the video stream drops from 500 kb to less than 200 kb.

For better authentication performance, fast implementations are already available from cryptographic products providers. The reported high-performance ECC SoC can perform signature generation more than 2000 times per second and signature verification more than 1000 times per second. With this SoC integrated, a single MARS can at least simultaneously verifies 20 video streams with bitrate of 400 kb, or 10 video streams with bitrate of 800 kb.

There is still a concern about possible bit error in RTP payload during its transmission that can result in failing the signature verification of some legal RTP packets. In our practice, these packets are dropped. Some forward error correction (FEC) algorithms can be applied to protect the media payload by using a RTP payload format with FEC feature [24]. Since authentication is done in a packet-independent manner, the authentication scheme is naturally robust to packet losses.

6. Conclusions and future work

In this paper, we propose a secure media streaming mechanism which combines encryption, authentication, and transcoding to address content protection, sender authentication, and media adaptation, respectively. and coherently. By introduction of MARS, we implement a secure media streaming system. MARS performs sender authentication by using an ECC digital signature algorithm with high security feature. Performance analysis shows the practicality of the sender authentication scheme. And the rapid development of fast implementation for ECC

algorithms implies a promising future for its application in media streaming sender authentication as well as solving other security issues. Nevertheless, the drawback of this sender authentication scheme is also obvious. That is, the cryptographic hash function used by digital signature algorithm is not robust to bit change and a single bit error can fail signature verification. To overcome this weakness, researchers have brought up perceptual hash algorithms [25,26] that extract robust, discriminative, and compact audio visual features as the digest or fingerprint of a media. Yet, the security of perceptual hash is controversial. And in the context of media streaming, it is difficult for content-based algorithms to extract out such kind of ideal features from a media segment which is of the small granularity as payload of a RTP packet. So, authentication based on perceptual hash is hard to be performed at RTP packet level.

The future work should be research on error correction techniques to reduce bit error occurrence for RTP payload, so that digital signature verification can be done more effectively. Perceptual hash should be studied as this approach implies possible breakthrough for multimedia authentication. And many interesting issues related to scalable video coding and its corresponding secure streaming mechanism need to be resolved so that more secure and adaptive media streaming can be achieved.

Acknowledgment

This work is supported by the Key Technologies R&D Program of China under Grant no. 2006BAH02A10.

References

[1] RTP: A Transport Protocol for Real-Time Applications, 〈http://www.ietf.org/rfc/rfc3550.txt〉, July 2003.
[2] GB/T 20090.2-2006, Information technology—Advanced coding of audio and video, Part 2: Video, 2006.
[3] B. Furht, R. Westwater, J. Ice, Multimedia broadcasting over the Internet: part II–video compression, IEEE multimedia 6 (1) (1999) 85–89.
[4] A. Lippman, Video coding for multiple target audiences, in: Proc. SPIE Visual Communications and Image Processing, San Jose, CA, USA, 1999, pp. 780–782.
[5] Weiping Li, Overview of fine granularity scalability in MPEG-4 video standard, IEEE Trans. Circuits Syst. Video Technol. 11 (2001) 301–317.
[6] Y. Wang, J. Ostermann, Y. Zhang, Video Processing and Communications, Prentice-Hall, New Jersey, 2002, pp. 368–393.
[7] A.R. Reibman, H. Jafarkhani, Y. Wang, M.T. Orchard, R. Puri, Multiple-description video coding using motion-compensated temporal prediction, IEEE Trans. Circuits Syst. Video Technol. 12 (2002) 193–204.
[8] A. Vetro, C. Christopoulos, H. Sun, Video transcoding architectures and techniques: an overview, IEEE Signal Process. Mag. 20 (2) (2003) 18–29.
[9] Longshe Huo, Qiang Fu, Yuanzhi Zou, Wen Gao, Network adapted selective frame-dropping algorithm for streaming media, IEEE Trans. Consumer Electron. 53 (2) (2007) 417–423.
[10] SRTP: the secure real time transport protocol, 〈www.ietf.org/rfc/rfc3711.txt〉 March 2004.
[11] ISMA 2.0: Internet Streaming Media Alliance Implementation Specification, Version 2.0, April 2005.
[12] Filippo Chiariglione, Tiejun Huang, Hyon-Gon Choo. Streaming of governed content—Time for a standard, in: Proceeding of the 5th IEEE Consumer Communications and Networking Conference (CCNC 2008), Las Vegas, USA, January 10–12, 2008.
[13] S.J. Wee, J.G. Apostolopoulos, Secure scalable video streaming for wireless networks, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, UT, May 2001.
[14] C. Venkatramani, P. Westerink, O. Verscheure, P. Frossard, Securing Media for Adaptive Streaming, in: ACM Conference on Multimedia, November 2003.
[15] D. Hankerson, A. Menezes, S.A. Vanstone, Guide to Elliptic Curve Cryptography, Springer, Berlin, 2004.
[16] A. Jurisic, A.J. Menezes, Elliptic Curves and Cryptography, Doc. Dobbs J., 1997.
[17] http://en.wikipedia.org/wiki/Certificate_authority.
[18] B. Schneier, Applied Cryptography, Wiley, New York, 1996.
[19] Accredited Standards Committee X9, American National Standard X9.62-2005, Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA), November 16, 2005.
[20] AVS working group, Information technology—Advanced coding of audio and video, Part 9: File Format, FCD, 2007.
[21] AVS working group, Information technology—Advanced coding of audio and video, Part 6: Digital Rights Management of Media, FCD, 2007.
[22] AVS working group, Information technology-Advanced coding of audio and video, Part 8: AVS over IP, FCD, 2007.
[23] http://www.openssl.org.
[24] http://www.ietf.org/rfc/rfc2733.txt.
[25] A. Mucedero, R. Lancini, F. Mapelli, A novel hashing algorithm for video sequences, 2004.
[26] Job Oostveen, Ton Kalker, Jaap Haitsma, Visual hashing of digital video: applications and techniques, 2001.