IEEE International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, Utah, May 2001.

SECURE SCALABLE VIDEO STREAMING FOR WIRELESS NETWORKS

Susie J. Wee and John G. Apostolopoulos

Streaming Media Systems Group
Hewlett-Packard Laboratories, Palo Alto, CA USA

ABSTRACT

We present a wireless video streaming system that securely and efficiently streams video to heterogeneous clients over time-varying communication links. Clients may differ in their display, power, communication, and computational capabilities and wireless channels may have time-varying bandwidths and quality levels that depend on channel usage and channel conditions. End-to-end system efficiency is achieved by placing transcoders at intermediate network nodes; these transcoders can easily adapt the video stream for particular client capabilities and network conditions.

This system uses our proposed method of secure scalable streaming (SSS) to simultaneously achieve scalability, efficiency, and security. Specifically, an SSS coder encodes video into secure scalable packets by using jointly designed scalable video coding, packetization, and progressive encryption techniques. This allows downstream SSS transcoders to transcode the secure scalable packets by simply truncating or eliminating packets, and without decrypting the coded video. A key feature of SSS is that it enables low-complexity transcoding operations to be performed at intermediate network nodes without compromising the security of the end-to-end wireless streaming system.

1. INTRODUCTION

Wireless streaming environments present many challenges for the system designer. For instance, clients can have different display, power, communication, and computational capabilities. In addition, wireless communication links can have different maximum bandwidths, quality levels, and time-varying characteristics. A successful wireless video streaming system must be able to stream video to heterogeneous clients over time-varying wireless communication links, and this streaming must be performed in a scalable, efficient, and secure manner. Scalability is needed to enable streaming to a multitude of clients with different device capabilities. Efficiency is needed to maximize the usage of the available network and device resources. Security is particularly important in wireless networks to protect content from eavesdroppers.

In order to achieve scalability and efficiency in wireless streaming environments, one must be able to easily adapt or transcode the compressed video stream at intermediate network nodes. A transcoder takes a compressed video stream as the input, then processes it to produce another compressed video stream as the output. Sample transcoding operations include bitrate reduction, rate shaping, spatial downsampling, frame rate reduction, and changing compression formats [1, 2]. Network transcoding can improve system scalability and efficiency for example by adapting the spatial resolution of a video stream for a particular client's display capabilities or by dynamically adjusting the bitrate of a video stream to match a wireless channel's time-varying characteristics [3].

While network transcoding facilitates scalability and efficiency in video streaming systems, it also presents a number of challenges. First, while computationally efficient transcoding algorithms have been developed, even these are not well-suited for processing hundreds or thousands of streams at intermediate wired network nodes or even a few streams at intermediate low-power wireless networking relay nodes. Furthermore, network transcoding poses a serious threat to the security of the streaming system because transcoding operations performed on encrypted streams generally require decrypting the stream, transcoding the decrypted stream, and then re-encrypting the result. Since every transcoder must decrypt the stream, each network transcoding node presents a possible breach in the security of the entire system.

We present a wireless video streaming system that simultaneously achieves three goals of scalability, efficiency, and security despite these challenges. This is accomplished with our proposed method of secure scalable streaming (SSS). SSS encodes video into secure scalable packets that are streamed to heterogeneous clients through hybrid wired and wireless networks. SSS allows transcoding operations to be performed at intermediate network nodes with low complexity and without decrypting the packets; thus, SSS enables low-complexity network transcoding without compromising the security of the system.

This paper is organized as follows. Section 2 describes two types of wireless streaming systems that deliver streaming video to heterogeneous clients and discusses how network transcoding can increase the end-to-end efficiency of these systems. Section 3 describes conventional approaches to secure video streaming and shows that these approaches do not allow network transcoding without compromising the security of the overall system. Section 4 presents our proposed method of Secure Scalable Streaming (SSS), which enables wireless streaming systems to securely stream video to heterogeneous clients while allowing low-complexity network transcoding to be performed without decryption. Finally, section 5 discusses system design issues that must be considered when designing SSS systems.

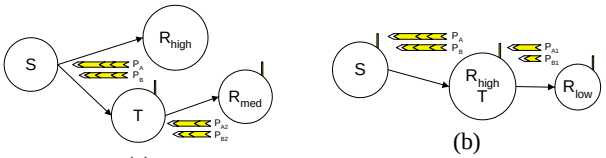
2. WIRELESS STREAMING SYSTEMS

Wireless streaming systems are limited by wireless bandwidth and client resources. Wireless bandwidth is scarce because of its shared nature and the fundamental limitations of wireless spectrum. Client resources are often practically limited by power constraints and by display, communication, and computational capabilities; for example, wireless transmission and even wireless reception alone typically consume large power budgets. In order to make the most efficient use of wireless bandwidth and client resources, it is desirable to send clients the lowest bandwidth video streams that match their display and communication capabilities [4]. In wireless streaming

systems where a sender streams video to a number of heterogeneous clients with different resources, network transcoders can be used to help achieve end-to-end system efficiency and scalability.

In hybrid wired/wireless networks, it is often necessary to simultaneously stream video to fixed clients on a wired network and to mobile clients on a wireless network. Figure 1a shows a hybrid wired/wireless network which consists of a wired sender, a wired high-resolution receiver, and a wireless medium-resolution receiver. In this system, the sender generates a full-bandwidth, high-resolution video stream that is sent to the fixed wired client. A transcoder, placed at the sender or the wired/wireless gateway, transcodes this stream into a lower-bandwidth, medium-resolution video stream which is then sent to the mobile wireless receiver.

In wireless appliance networks, mobile senders and receivers communicate with one another over wireless links. A sender's coverage area is limited by the power of the transmitted signal. Relay devices can be used to extend the wireless coverage area when intended receivers are beyond the immediate coverage area of the sender. In the case of heterogeneous clients, transcoders can be used to adapt a video stream for a particular client or communication link. Transcoding can be performed in a relay device or in a receiver which also acts as a relay. Figure 1b shows a wireless appliance network that consists of a wireless sender and a high- and low-resolution wireless receiver. In this system, the high-resolution receiver receives and decodes the high-resolution video stream; in addition, it transcodes it and relays the resulting lower-bandwidth stream to the low-resolution receiver.



(a) (b)

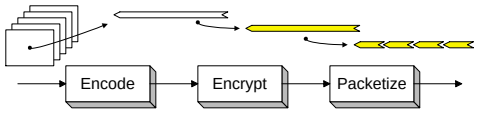
Fig. 1. Wireless streaming systems: (a) Hybrid wired/wireless network with heterogeneous clients and an intermediate transcoding node; (b) Wireless appliance network with heterogeneous clients with transcoding and relay capabilities.

Both these systems use network transcoders to transcode video streams into lower bandwidth streams that match the display capabilities of the target wireless clients. Generally speaking, these systems illustrate how network transcoding can enable efficient use of wireless spectrum and client resources by transcoding video streams into formats better suited for transmission over particular channels and for capabilities of target clients. Thus, we consider transcoding to be a critical part of a wireless streaming system.

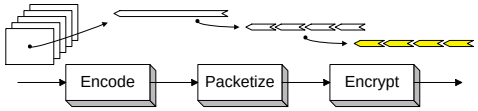
3. CONVENTIONAL APPROACHES TO SECURE VIDEO STREAMING

This section discusses two conventional approaches for secure video streaming. Figure 2a shows a secure video streaming system that uses application-level encryption. The video is first encoded into a bitstream using interframe compression algorithms such as MPEG or H.263 or intraframe compression algorithms such as JPEG or JPEG2000. The resulting bitstream is encrypted, and the resulting encrypted stream is packetized and transmitted over the network using a transport protocol such as UDP. The difficulty with this approach occurs when a packet is lost. Specifically, error recovery is difficult because without the data from the lost packet, decryption and/or decoding may be difficult if not impossible.

Figure 2b shows a secure video streaming system that uses network-level encryption. This system can use the same video compression algorithms as the previous system. However, in this system the packetization can be performed in a manner that considers the content of the coded video and thus results in better error recovery, a concept known to the networking community as application-level framing. For example, a common approach is to use MPEG compression with the RTP transport protocol which is built on UDP. RTP provides streaming parameters such as time stamps and suggests methods for packetizing MPEG payload data to ease error recovery in the case of lost or delayed packets.



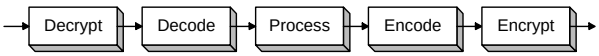
(b) Conventional approach #1: Application-level encryption.



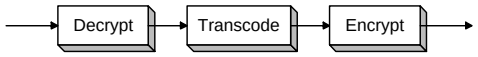
(a) Conventional approach #2: Network-level encryption.

Fig. 2. Conventional approaches to video streaming.

Both these approaches are secure in that they transport the video data in encrypted form. However, if network transcoding was needed, it would have to be performed with the method shown in Figure 3a. The transcoding operation is a decrypt, decode, process, re-encode, and re-encrypt process. The computational requirements of this operation can be reduced by incorporating efficient transcoding algorithms in place of the decode, process, and re-encode modules as shown in Figure 3b. However, even improved transcoding algorithms have computational requirements that are not well-suited for transcoding many streams in a network node. Furthermore, a more critical drawback stems from the basic need to decrypt the stream for every transcoding operation. Each time the stream is decrypted, it opens another possible attack point and thus increases the vulnerability of the system. Thus, each transcoder further threatens the security of the overall system.



(a) Conventional transcoding.



(b) Conventional transcoding with improved efficiency.

Fig. 3. Conventional approaches to transcoding.

4. SECURE SCALABLE STREAMING

This section describes our proposed method of Secure Scalable Streaming (SSS).

4.1. SSS Coding

The SSS coder encodes the input video frames into secure scalable packets that can be streamed to heterogeneous clients over wireless

networks. The SSS coder was developed by jointly designing the compression, packetization, and encryption modules of the coder. More specifically, scalable coding and packetization modules were designed in conjunction with progressive encryption techniques. The resulting SSS video streams have the feature that subsequent transcoding operations such as bitrate reduction and spatial downsampling can be performed without decrypting the video and thus while maintaining the security of the system.

Our SSS coding method is shown in Figure 4. First, the video frame is segmented into tiles. Then, each tile is coded into two portions: header data and scalable video data. Next, the scalable video data is encrypted with progressive encryption techniques. Finally, a packet is created by combining the unencrypted header data with the progressively encrypted scalable video data. The resulting secure scalable packets are streamed over the network to the receiving clients.

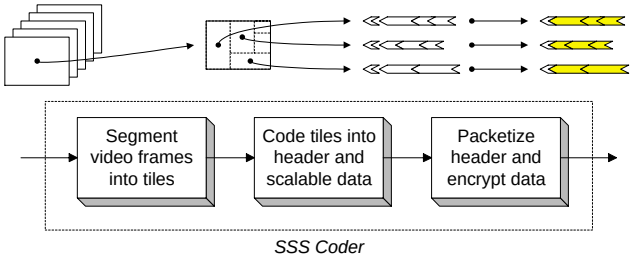


Fig. 4. SSS Coding.

Standard scalable video coding techniques are used to code each tile into scalable video data. Scalable video data has the property that the first small portion of the data can be decoded into baseline quality video, and larger portions can be decoded into improved quality video. It is this property that allows packets to be transcoded to lower bitrates or spatial resolutions simply by truncating the packet. Progressive encryption techniques include cipher block chains or stream ciphers. These methods have the property that the first portion of the data is encrypted independently, then later portions are encrypted based on earlier portions. When properly matched with scalable coding and packetization, progressive encryption preserves the ability to transcode packets with simple packet truncation. Additional details about the joint scalable coding and packetization method are discussed in Section 5.1 and about the progressive encryption in Section 5.2.

4.2. SSS Transcoding

In SSS coding, the scalable coding and packetization modules are combined with progressive encryption modules. It is this combination that allows subsequent SSS transcoding operations to be performed by packet truncation or elimination and without decryption. The resulting SSS transcoder is shown in Figure 5. SSS transcoders can transcode packets by reading the unencrypted header data at the beginning of each packet, then truncating packets at the appropriate locations that may be specified by the unencrypted header.

The SSS transcoder can be compared with the conventional transcoder shown in Figure 3. Notice that the conventional approach requires decryption while the proposed approach does not. Also, the conventional transcoder has much higher computational requirements due to the computations needed for decryption, conventional transcoding, and encryption. However, while insecure,

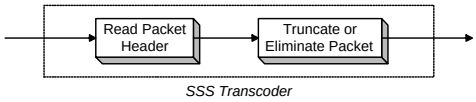


Fig. 5. SSS Transcoding.

the advantage of the conventional approach is that it can be performed on standard-compliant streams. Thus, if one does not have the freedom to encode the video content, the conventional approach is the only alternative. On the other hand, if one does have the freedom to encode the video content, SSS coding offers a number of benefits. Specifically, an SSS system provides end-to-end security while enabling very low complexity transcoding to be performed at intermediate, possibly untrusted, nodes without compromising the security of the system.

5. SYSTEM DESIGN CONSIDERATIONS

5.1. Scalable Coding – Packetization

The scalable video coding and packetization modules of the SSS coder were jointly designed to enable downstream transcoding operations to be performed by simple packet truncation. SSS coding is similar to bitstream scalable video coding, but it further partitions the video frames into scalable packets that correspond to predetermined regions or tiles in the video sequence. The JPEG 2000 image compression standard has many of these characteristics of independently coded tiles and scalability within the tile. We build upon these concepts by extending this level of scalability to video frames and by combining it with the packetization process; furthermore, we jointly design this smart packetization with the encryption process.

Separate regions or tiles of a video frame are encoded into one or more packets. Each packet contains header data and scalable video data. The header describes the region that the packet represents and other information needed for subsequent SSS transcoding and decoding operations. Such information may include a series of recommended truncation points for packet transcoders. The scalable video data contains the actual coded video. In the case of intraframe coding, the video data may be the coded pixels; while in the case of interframe coding, it may be the motion vectors and coded residuals that result from motion-compensated prediction. Scalable coding techniques can be used in both cases to create an embedded or scalable packet that can be truncated to lower the resolution or fidelity of the coded video data.

5.2. Scalable Coding – Packetization – Encryption

If the entire packet was encrypted with one long block code, it would not be decodable unless it was received in its entirety. Since we are using scalable packets and we wish to be able to transcode the stream by packet truncation, it is useful and necessary to encrypt the packets in a similarly progressive manner. Thus, SSS coding performs encryption by using progressive encryption methods such as cipher block chains or stream ciphers [5].

Progressive encryption methods have the property that smaller blocks of data are encrypted progressively. While block code encryption with small block sizes is not very secure, progressive encryption methods add a degree of security by feeding encrypted data of earlier blocks into the encryption of a later block. Decryption can then be performed progressively as well. The first small

block of ciphertext can be decrypted into plaintext by itself while later blocks of ciphertext depend on the decrypted plaintext from earlier blocks. Thus, earlier blocks of ciphertext can be decrypted without knowledge of the entire ciphertext segment. This progressive nature of cipher block chains and stream ciphers matches nicely with the progressive or embedded nature of scalable coding. It is this combination that enables efficient secure transcoding operations to be performed in SSS.

While the payload data is encrypted progressively, the header data is left unencrypted so that transcoding nodes can use this information to make transcoding decisions. For example, the unencrypted header can contain information such as recommended truncation points within the encrypted packet. This can be used to achieve near RD-optimal bitrate reduction by intermediate transcoding nodes. This is discussed further in Section 5.3.

5.3. Rate-Distortion Optimality

A highly desirable feature of a scalable system is the ability to transcode the compressed stream to different rates, each of which is rate-distortion (RD) optimal or near-RD optimal. Achieving this property is rather straightfoward if an entire frame is coded into a single embedded bitstream that is sent within a single packet; then, any truncation point of the single packet (embedded bitstream) is nearly RD optimal by design. However, this is not possible in a system where frames are coded into multiple packets, unless transcoders accumulate all the packets of a frame and de-packetize, decrypt, process, re-encrypt, and re-packetize these packets.

RD optimal coding is achieved by generating an RD plot for each tile of an image, and then operating all tiles at the same slope λ that generates the desired total bitrate. We achieve near-optimal transcoding at the packet level by placing the optimal RD cutoff points for a number of quality levels in the unencrypted headers of the packets. Then, a transcoder can truncate each packet at the appropriate cutoff point; thus the resulting packets will contain the appropriate number of bits for each region of the video for the new desired quality level. Notice that the transcoder simply needs to read each packet header, then truncate the packet at the appropriate point. This is illustrated in Figure 6 where three tiles in an image are coded into separate packets, and for each tile three RD optimal points are identified and their locations placed in the respective packet header. A transcoder can choose to operate at any of the three RD points (or points in-between) and then truncate each packet at the appropriate cutoff point.

5.4. Drift vs. Compression Efficiency

A problem with truncating packets in a system that uses motion-compensated prediction is the drift that results when truncated or eliminated data is not available to the receiver. This drift problem may be addressed in a number of ways. First, drift may be completely eliminated by allowing the prediction to only depend on base-level video data that is required to be at all decoders. The drawback of this approach is the lost compression efficiency that results from not including the rest of the video data in the prediction loop. The other extreme is to maximize compression efficiency by using all the coded video data in the prediction loop and allowing drift in all the partial reconstructions.

The middle ground is to allow a small amount of drift to exist in the system. For example, in a system that codes with three or more RD reconstruction levels, one could allow the prediction to depend on the video data contained up to the second cutoff point. If a transcoder truncates packets at the first cutoff point, then some drift will occur. If a transcoder truncates packets at the second cutoff point, then RD optimality is achieved. If the transcoder truncates data at third or higher cutoff points or if no transcoding is performed, then no drift occurs, however reduced compression efficiency will result because only partial video data is used in the prediction loop. Empirical evidence shows that the compression efficiency gained by using the higher quality video in the prediction loop often outweighs the reduced performance caused by drift errors in the lower quality video. However, the choice of cutoff level to use in the prediction loop is quite heuristic and is left to the system designer.

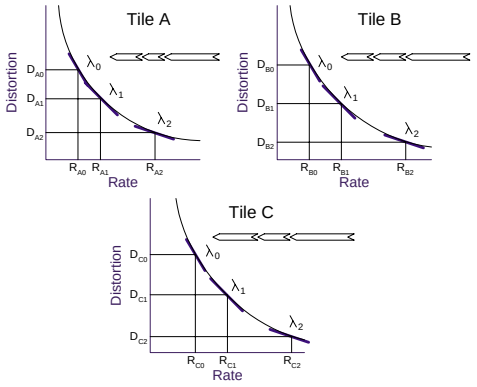


Fig. 6. RD optimality of Secure Scalable Streaming.

6. SUMMARY

Important features for video streaming over wireless networks include scalable video coding, efficient transcoding, and security. SSS provides scalability, efficiency, and security by encoding video into secure scalable packets through the use of jointly designed scalable video coding, packetization, and progressive encryption techniques. SSS transcoders can then transcode these secure scalable packets by simply truncating or eliminating packets, and without decrypting the coded video. A key feature of SSS is that it enables low-complexity and high-quality transcoding to be performed at intermediate, possibly untrusted, network nodes without compromising the security of the end-to-end wireless streaming system.

7. REFERENCES

[1] H. Sun, W.K. Kwok, and J.W. Zdepski, "Architectures for MPEG compressed bitstream scaling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 2, April 1996.

[2] S.J. Wee, J.G. Apostolopoulos, and N. Feamster, "Field-to-frame transcoding with spatial and temporal downsampling," in *IEEE International Conf. on Image Processing*, Kobe, Japan, October 1999.

[3] W. Tan and A. Zakhor, "Real-time internet video using error resilient scalable compression and TCP-friendly transport protocol," *IEEE Transactions on Multimedia*, pp. 172–186, June 1999.

[4] S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 6, August 1997.

[5] Bruce Schneier, *Applied Cryptography*, John Wiley & Sons, Inc., 2 edition, 1995.