

Reliable Transmission of Security-enabled Multimedia over Internet

David E. Moore and Farid Ahmed
Department of Electrical Engineering and Computer Science
The Catholic University of America, Washington, DC 20064

Abstract

In this paper we address the reliable transmission of security-enabled multimedia data over the internet which is becoming increasingly vulnerable to a variety of cyber-attacks. Due to their real-timeliness aspect, multimedia data in internet mostly uses User Datagram Protocol (UDP) as the transport media as opposed to the Transport Control Protocol (TCP). UDP is inherently an unreliable transport media that results in certain unacknowledged packet losses. Multimedia applications usually can tolerate some packet losses for its rendering at the receiver side. But, for the security-enhanced multimedia that we are talking about, reliability of reception of most of the packets within a certain tolerance time need to be guaranteed. This is where we come in with a new protocol that ensures packet-level reliability as well as stream-level authentication of multimedia.

Keywords: Security, multimedia, cyber-threats, watermarking, buffer management

1. INTRODUCTION

Multimedia data are being transmitted over the internet at an increasing rate. Applications like video conferencing, chat, internet telephony, tele-radiology, real-time remote control applications are gaining more momentum. At the same time, internet itself is becoming alarmingly vulnerable with the sophistication of cyber-attacks. A number of general cryptographic security protocols for different layers of internet architecture are already in force, while some others are in proposition [1].

When a multimedia data stream is sent across public networks the one of the primary concerns is to maintain the perceived continuity of the data. An occasional missing packet is not crucial when compared to the consumer's perception of continuity. Consequently, most multimedia protocols depend on connection-less datagram services. UDP is the primary example of this, with modification to lower layer protocols (RSVP [2]) and to higher layer protocols (RTP [3]) existing to ensure that the desired Quality of Service (QoS) is obtained [4]. TCP and other connection oriented protocols require more host to host interaction resulting in slower data throughput. This overhead prevents these connection-based protocols from providing a general purpose solution for multimedia delivery [4]. Approaches to securing multimedia streams have, up to this point, dealt with securing particular codecs [6], securing the edge protocols used during transmission [7], and restricting communication to privately held resources [8]. These solutions are restricted to a particular application and usually come at great costs.

In this work we specifically consider the reliability of security-enabled multimedia. Security is enabled by adding intelligent authentication information into the media by an information hiding technique. We in particular use a wavelet based digital watermarking technique for the hiding process. Authentication digital watermark has been shown recently to be very effective for secure multimedia communication [9]. 'Reliability' in our work refers to the lossless reception of the packets at the receiving end. Packet loss in the internet is addressed in two different generic way- retransmission and receiver-based correction. While TCP allows retransmission, multimedia streaming applications usually run on UDP. A number of packet loss recovery techniques are mentioned in [10].

The reliability that we prescribe comes from both the proposed secure streaming protocol and the embedded intelligence.

2. PROPOSED SECURITY PROTOCOL

There are two aspects of the proposed protocol. First the media is made security-enabled by adding self-authentication information using a transform domain watermarking technique. Secondly we propose a secure streaming protocol (SSP) for ensuring the continuity and reliability of the received media. Security is thus considered in two separate ways. One is done before the actual transmission of multimedia and the later is while the data is in transit across the internet.

2.1 Transform Domain Watermarking for Security-enabled multimedia

Figure 1 shows the process of security-enabled multimedia. First robust digital signatures are extracted from the media which can uniquely identify each packet and its logical sequence with its neighboring packets. This signature is then embedded with the packet using a digital watermarking technique. The resulting media packets will have self-authentication signatures embedded in them.

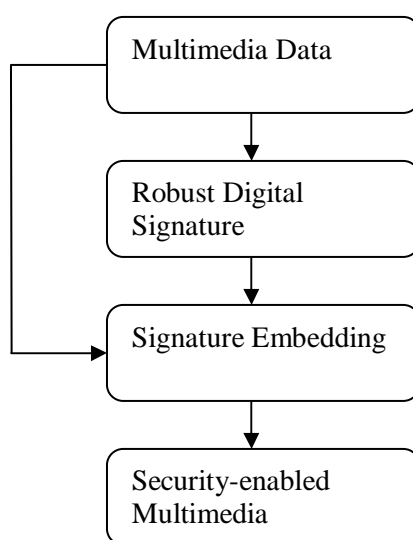


Fig. 1: The Process of Security-enabled Multimedia

In a previous work [11] we have shown how to extract features that can be used as image hash or signatures. In this work, we use that notion of signature, where part of the signature of a packet is dependent on the context, while the other part represents its relation with the neighboring packets. In order for the receiver to authenticate the packets, the signature must be robust to the watermarking process itself. That means features should be selected and processed such a way that the watermarking should not bring in any change in the signature. If a signature of a packet is authenticated, it gives two pieces of information. First, it is in its proper order in the sequence and second, the packet contents is not tampered with.

Figure 2 shows a block diagram of packet level signature embedding using transform domain watermarking process. Digital watermarking is the process of hiding imperceptible information into a host digital media. Recently, there has been a significant amount of research done in the field of watermarking [12]. These are broadly classified as i) Spatial domain vs. Transform domain watermark, ii) Robust vs. Semi-Fragile watermark, and iii) Blind vs. Non-blind watermark. In this research we propose to use transform domain and more specifically discrete cosine transform (DCT) domain watermark. This choice is motivated by the following facts. Firstly, this makes a predictable and analyzable effect of the robust signatures as mentioned above. Secondly we'll have better control over the feature selection process. Thirdly, the proposed research when applied to SSP as mentioned in the next section necessitates some sort of progressive watermarking which is better suited in transform domain. As for the robustness, we choose the semi-fragile paradigm so that the tampers in an image can be easily identified by observing the embedded watermarks and thus ensuring authentication. Finally we have to choose blind watermark detection as it is most practical and does not necessitate the original image.

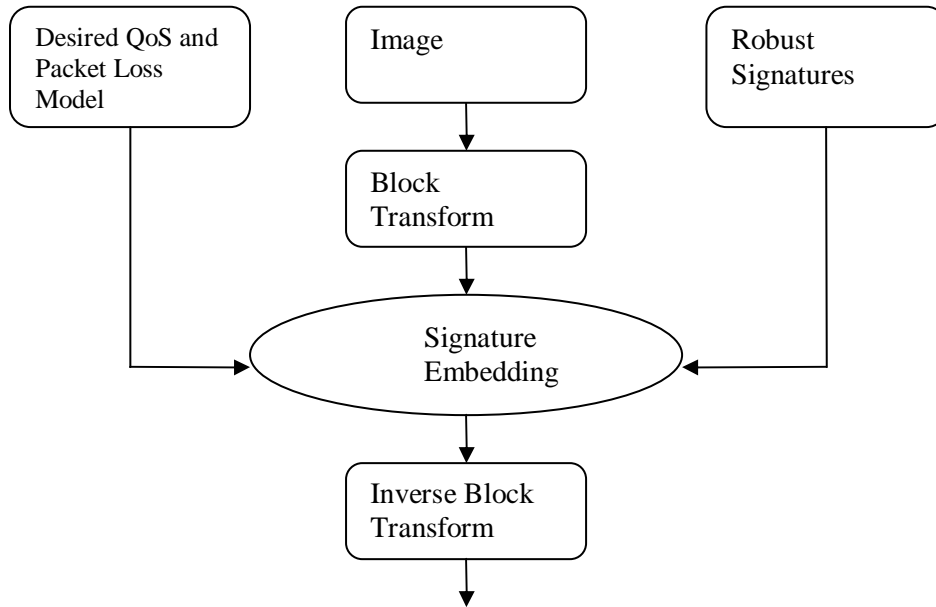


Fig. 2: Transform domain Signature embedding process

The desired QoS and packet loss model mentioned in Fig. 2 is used to select which block and/or which coefficients need to be marked as demanded by the required progressive watermarking.

2.2 Secure Streaming Protocol

The proposed secure streaming protocol (SSP) is used to ensure that all packets arrive at the final destination reliably and within the window of time prescribed for the real-time rendering of the multimedia. Coupled with the self-authentication information embedded in the security enabling phase, it is therefore aimed at achieving security at both the packet level and the stream level. The data will travel across a datagram service, such as UDP, to keep the network requirements low. Any failed security checks or missing data packets will be resent. Retransmitted data is accomplished via a connection-based protocol, such as TCP.

The details of SSP along with required buffer management, retransmission manager, and security manager now follows.

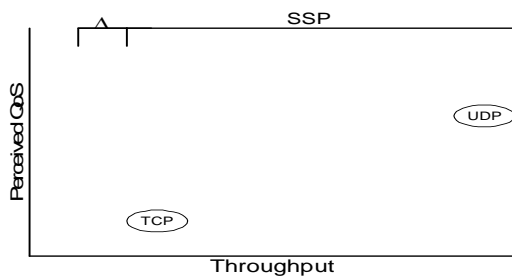


Fig. 3: Relative QoS offered by the proposed SSP

Figure 3 diagrams the relative throughput of two transport protocols, TCP and UDP, along the horizontal axis. The vertical axis shows the relative impact on the perceived QoS when considering the relative throughput achieved by particular protocols. Perceived QoS is loosely defined here to include factors such as multimedia shutter, jitter and continuity of stream. The throughput of SSP is expected to have a lower bound similar to that of a UDP connection and an upper bound close to TCP plus a protocol defined constant called Δ . Discussion of the constant Δ will be deferred to later in this paper.

Consequently the perceived QoS of multimedia transmissions using SSP would also improve. It is worth mentioning that this throughput is achieved without compromising the reliability of data delivery. Data sent across SSP are guaranteed to reach the destination, similar to what TCP offers.

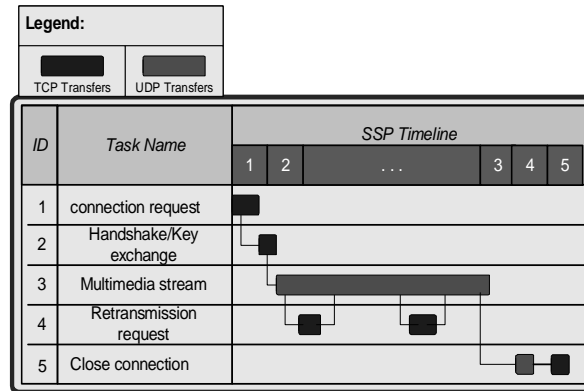


Fig. 4: SSP task sequence

Fig. 4 shows the task sequences of the proposed protocol. A connection request is sent to the server from the client via TCP. If data encryption is desired, the symmetric key exchange is done during this client/server handshake. Although a specific handshake algorithm is not defined in this paper, a handshake protocol similar to the one used by the Transport Layer Security (TLS) protocol should suffice [5]. Once the session key has been exchanged, the client and server determine an appropriate Static Transmission Unit (STU) for this connection. The Internet Protocol recommends a payload of 576 bytes [13]. Higher values may be appropriate if the protocol's implementers are certain that the datagram will not be fragmented during transmission. Once an appropriate STU has been exchanged, transmission of the multimedia content via the UDP protocol commences. Every datagram sent from the server to the client will have a payload size equal to the STU. If, at any point during the transmission, the client-side determines that there is missing or corrupted data, the client requests the missing data to be retransmitted back to the client from the server. Once all the content has been viewed, the client is assured that all the data has been transmitted successfully from the server. When the client wishes to close the connection, the connection completion notification is sent to the server via TCP. The client and server then free the UDP and TCP resources associated with this connection.

For the server, the secure stream entails a connection manager to the client and a retransmission manager. The connection manager is responsible for the UDP connection to the client. This process treats all multimedia content as a sequence of packet payloads, which when summed, is equivalent to the total multimedia file. Equations (1) and (2) show these relationships. Note that MM is the multimedia file, $Frg()$ is the fragmentation function and λ is the total number of packets to be sent. The fragmentation function $Frg()$ need not be as simple as defined here. Other possibilities exist, such as embedding watermarks on the packet's payload. Work is currently being done to investigate the progressive watermarking options using techniques that selectively watermark packets. Also note that this is an instantaneous view of the data to be transmitted to the client. All the data may not be available and may change as a function of time - video conferencing is a good example of this. Therefore, in these applications, λ will increase as a function of time and encoding rate. These relationships can easily be fully derived.

$$\lambda = frg(M)$$

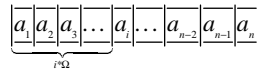
$$frg(M) = \frac{|M|}{\Omega} + \begin{cases} 1 & \text{if } |M| \bmod \Omega < 0 \\ 0 & \text{else} \end{cases} \quad (1)$$

$$\sum_{n=1}^{n=\lambda} a_n = |M| \quad (2)$$

$$|a_n| = \Omega$$

Responsibilities of the connection manager include creating a digest of each datagram, optionally encrypting the datagram and giving each datagram a sequence number. The connection manager also has a reference to the multimedia data source, an indication of how large the packets are that sent to the client, and the session key.

The retransmission manager is a TCP connection to the client and is responsible for resending a datagram's payload over TCP. Since the size of each packet is predetermined, the server can locate any previously sent datagram within the data source for retransmission. If a request is received to have a_n retransmitted, the retransmission manager knows that $a_n = \{a\}_\lambda$ and can find the appropriate payload using observation that $i * \Omega = \text{offset } (M)$. The following illustration demonstrates this.



On the client, a secure stream consists of the following: the session key, a security buffer, a retransmit manager, and a buffer manager. The security buffers acts as private, safe space for both the buffer and retransmit managers to deal with the multimedia content. The security buffer is of a predetermined size, that is $|SB| = \Xi$. The value Ξ can be defined as number of packets or as a segment of playback time t . Is the latter case Ξ is dependant on the following formulation:

$$f(\Xi) = t * \text{bitrate}(M)$$

The security buffer is treated at a window of time moving along the stream of data. Its purpose is to verify that all data is in place. Since the packets are given a sequence number, there is a logical order to how the packets should be arranged within the security buffer. Furthermore, since Ξ is an independent of any datagram coming in, the size of the security buffer does not change.

The retransmission manager handles requests for information packets to be resent over the TCP connection. This task maintains the initial connection request made to the server over TCP. The retransmission manager is also responsible for putting information received from the server back into its proper place in the data stream.

Figure 5 illustrates the relationships of the different tasks on the client.

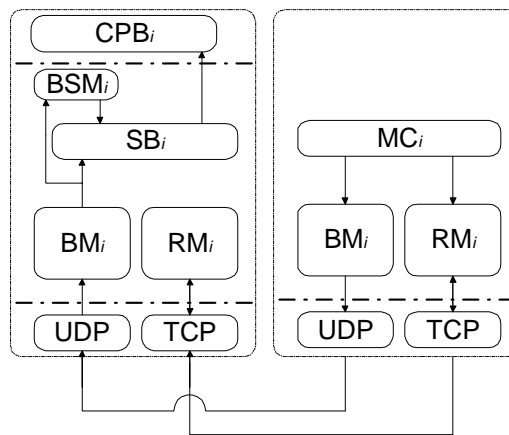


Fig. 5: The client model of SSP

The heart of this proposal is the buffer management task. Since the assumption is that the communication protocol is connection-less, there is no assurance that the client will receive data sent to in from the server. The buffer management task allows developers to assume that a secured virtual connection resides between the client and server above the network

layer. This task processes in-band data sent along the connection-less networking protocol and determines where to put the packet's payload. This is accomplished by introducing a three tier structure of network buffers as opposed to the normal two tier structure. Normally a datagram goes directly from the UDP buffer to the content players buffer (CPB) where the content player consumes the data at will. SSP uses the security buffer (SB) as a middle tier between the network stack's buffer and the application's buffer as shown in Fig. 6. A transfer rate is associated with data movement between buffers. Normally the transfer rate α is present between the network stack and the application's consumption buffer. This transfer rate is dependant primarily on the throughput of the network and partially on the operating system. Since the operating system contribution to the transfer rate α is minimal compare to the network throughput, it will be ignored. The addition of a third buffer creates a second transfer rate β as information moves from the network to the applications. The new transfer rate is dependant both on α and the controllable parameter Δ -defined as the rate in which data is moved from the SB to the CPB.

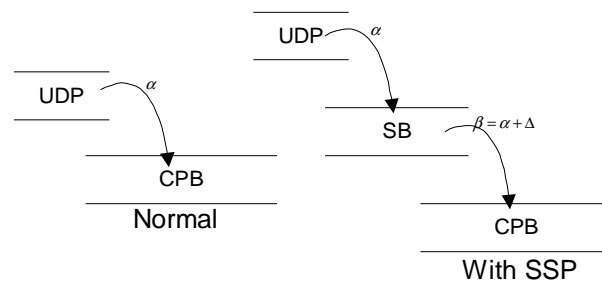


Fig. 6: Buffer manager processing a datagram

Before a data packet is moved from the UDP buffer to the security buffer, buffer manager does some preprocessing. If channel between the client and server is encrypted, the buffer manager will have to decrypt each packet in order to determine how to best handle any given packet. After this optional step, two discrete checks are performed: a signature check and a sequence number check as shown in Fig. 7. In the event of a digest check failure, a request is sent to the retransmission manager to have the server resend this failed packet. The sequence check is done by a security buffer monitor process to make sure that incoming packets fall within the size of the security buffer or move the packets to a cache if fall outside the range defined by Ω .

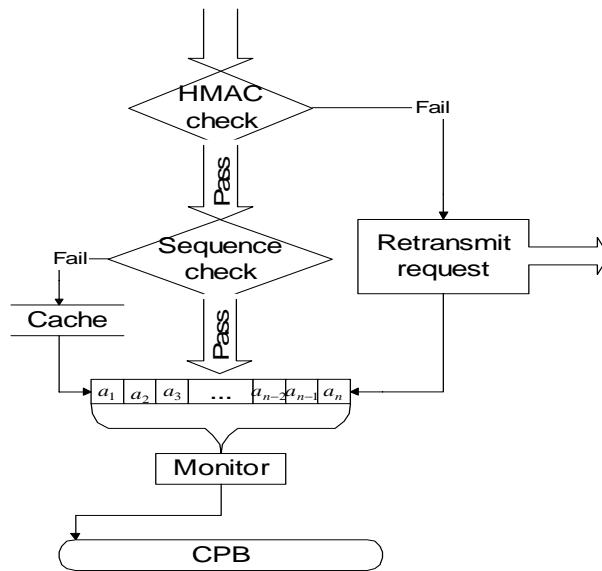


Fig. 7: Retransmit Manager

The security buffer monitor merits extra attention, since it monitors all that is going on within the buffer. If a packet is out of sequence two scenarios exist: the datagram's sequence number falls within the security buffers window of time, Ω , or the sequence number is outside the window. If the datagram's sequence number is falls within the window, the buffer manager searches the security buffer for the correct location and inserts the payload in the buffer. Packets that need to be placed in a position outside of the security buffer are stored in a cache separated from the security buffer. There is not implied order within the cache. A datagram stored in the cache are simply viewed as a member the set of extra data by the monitor process. This is treated as a special case. When the datagram's sequence number is before the sequence counter, the packet is initially stored in the security buffer cache. A packet leaves the cache when the sequence number associated with the packet falls within the bounds of the time window of the security buffer.

Data movement between SB and CPB is bound by the following equation:

$$\Delta = s + \delta + \xi \quad (3)$$

where s is Packet Retransmit Scheduler (PRS) delay, δ is frequencies in which packet move from the security buffer to the content player's buffer and ξ is defined as the ratio of packet loss. Packet loss is defined as any situations which require a packet to be retransmitted, i.e. missing packet or a failed signature check. The PRS is an algorithm used to determine when a missing packet needs to be requested for retransmission. Initially the PRS will be defined simply as a Lowest Sequence First (LSF) algorithm. That is to say the scheduler will move down the security buffer in linear fashion asking for missing packets to be retransmitted. The coefficient s is time value used to define how fast the scheduler moves down the security buffer. The second parameter to Δ has a range of $0 > \delta > |\text{SB}|$. When $\delta = 1$, a packet is place in the CPB as soon as it appears in the beginning of the SB. Conversely, when $\delta = |\text{SB}| - 1$, the security buffer moves to the application buffer as a single block. It should be noted that when both s and δ are 0, then Δ is 0 and β is only dependant on the ratio of missing data packets. If the ratio of packet loss, ξ , is 0, then SSP performs as though it were a UDP connection. On the other hand, with $\Delta = 0$ and $\xi = 1$, then all data will need to be resent over the TCP connection to the server and SSP performs as though it were a TCP connection.

3. CONCLUSION

A proof of this concept is in development. Following this barebones application, a plug-in for the popular multimedia players (i.e. Apple's QuickTime media player) will be developed. As soon as a test bed is developed, data will be gathered to see how the performance of SSP is affected by various values for the parameters used by SSP. An investigation into

optimizing the algorithm's parameters will also be done, primarily the Kalman Filter [14] used by the buffer size manager. Modifications to this algorithm could include a look into how user-defined confidence levels to the data being transmitted will affect the performance of this algorithm's tunable parameters. The effect of progressive digital watermarks on the authentication performance of this algorithm could be investigated.

Different algorithms can be used by the fragmentation function on the server side and the packet retransmission scheduler on the client side. A different fragmentation function could be used to allow intelligent watermarks to be placed in set packet intervals in order to improve the customer experience. Additionally, an improvement in the PRS could allow greater throughput and reduce the instances of redundant retransmits.

REFERENCES

- [1] Cryptography and Network Security- Principles and Practices, William Stallings, Third ed. 2003, Prentice Hall, ISBN 0-13-091429-0
- [2] Braden, R., Ed., Zhang, L., Estrin, D., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) Version 1 Functional Specification", RFC 2205, September 1997
- [3] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications." Work in Progress, Mar. 1995.
- [4] Neuman de Souza, Jose (ed.), Boutaba, Raouf (ed.), "Managing Qos in Multimedia Networks and Services" 2000, Kluwer Academic Publishers; ISBN: 0792379624
- [5] Dierks, T. and C. Allen, "The TLS Protocol, Version 1.0", RFC 2246, January 1999.
- [6] Dittmann, J., Wohlmacher, P., and Nahrstedt, K., "Using Cryptographic and Watermarking Algorithms", IEEE Multimedia and Security, Oct. 2001, pp 54-65
- [7] Berger, L., O'Mallery, T., "RSVP Extensions for IPSEC Data Flows," RFC 2207, September 1997
- [8] Preneel, Bart Ed., "Secure Information Networks: Communications and Multimedia Security", 1999, Kluwer Academic Publishers; ISBN: 0792386000
- [9] Liehua Xie and Gonzalo R. Arce, "A Class of Authentication Digital Watermarks for Secure Multimedia Communication," IEEE Trans. Image Proc., vol. 10, no. 11, 2001.
- [10] C Perkins, O Hodson, V Hardman, "A Survey of packet loss recovery Techniques for Streaming Audio," IEEE Network, Sep 1998, pp. 40-48.
- [11] F. Ahmed and E. Abebe, "A Hybrid Approach to secure Storage and Transmission of Medical Informatics," Proceedings of International Conference on Computers and Information Technology (ACM-sponsored), Dec 2002.
- [12] Ingemar Cox, Jeffrey Bloom, Matthew Miller, "Digital Watermarking: Principles & Practice," 2001, Morgan Kauffman Publishers, ISBN 1-55860-714-5.
- [13] Postel, J. (ed.), "DOD Standard Internet Protocol," Defense Advanced Research Projects Agency, Information Processing Techniques Office, RFC 760, IEN 128, January 1980.
- [14] Baochun Li, Dongyan Xu, Klara Nahrstedt, "Optimal State Prediction for Feedback- Based QoS Adaptations", *Proc. of Seventh IEEE/IFIP International Workshop on Quality of Service (IWQoS 99)*, London, UK, May, 1999.