ARMS : Adaptive Rich Media Secure Streaming

Lisa Amini Raymond Rose Chitra Venkatramani
Olivier Verscheure Peter Westerink
IBM T.J. Watson Research Center
P.O.Box 704
Yorktown Heights, NY 10598
{aminil, rer, chitrav, peterw, ov1}@us.ibm.com

Pascal Frossard
Signal Processing Institute
EPFL
Lausanne 1015,Switzerland
pascal.frossard@epfl.ch

ABSTRACT

In this demonstration we present the ARMS system which enables secure and adaptive rich media streaming to a large-scale, heterogeneous client population. The ARMS system dynamically adapts streams to available bandwidth, client capabilities, packet loss, and administratively imposed policies - all while maintaining full content security. The ARMS system is completely standards compliant and to our knowledge is the first such end-to-end MPEG-4-based system.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Distributed Systems*

General Terms

Design, Security, Standardization

Keywords

Adaptive, Encrypted, MPEG-4, video server,streaming, Scalability

1. INTRODUCTION

Many enterprises use streaming video to convey news clips or corporate communications to their employees or clients. However, since the networks are based on packet-switching technology which is designed for data communication, achieving efficient distribution of streaming video and multimedia to a wide heterogeneous user population poses many technical challenges. Besides the standard video-over-IP issues, enterprises have additional requirements due to the need to control a shared infrastructure where business media comes first. In addition to challenges in terms of video coding and networking, one of the key requirements for enterprise streaming is clearly posed in terms of security. The video distribution has to be efficient and to adapt to the clients requirements, while at the same time offering a high degree of security through proper authentication, authorization and encryption techniques.

MM'03, November 2–8, 2003, Berkeley, California, USA..
ACM 1-58113-722-2/03/0011.

In this demonstration we present the ARMS system which is an end-to-end system for streaming multimedia in an enterprise. The ARMS system dynamically adapts streams to available bandwidth, client capabilities, packet loss, and administratively imposed policies - all while maintaining full content security. Specifically, ARMS addresses the following requirements of an enterprise media distribution solution – (i) Adaptive, Secure and Standards-based streaming, (ii) Archive and VoD Streaming, (iii) Scalable Broadcast using Application-level or IP multicast, to a large-scale heterogeneous client population. A key feature of ARMS technology is open-standards compliance - ARMS employs MPEG-4 encoding and HTTP or RTSP/RTP protocols, and can be demonstrated with ISMA-interoperable MPEG-4 players, including the IBM Java MPEG-4 player. To our knowledge, this is the first such end-to-end MPEG-4 based system.

2. SYSTEM ARCHITECTURE

The main components of the ARMS streaming architecture are illustrated in Figure 1. The components consist of the broadcaster which is based on a multiple encoding scheme, the VideoStore to store the multiply encoded content, the server which uses a simple and efficient stream-switching technique for adaptation, and finally the playback clients. The figure illustrates a simple configuration with one instance of each of the main components. In large scale deployments, the streaming servers can be networked for distribution and there can be multiple Broadcasters and VideoStores. The working of each of the components with the key standards they support is described below.

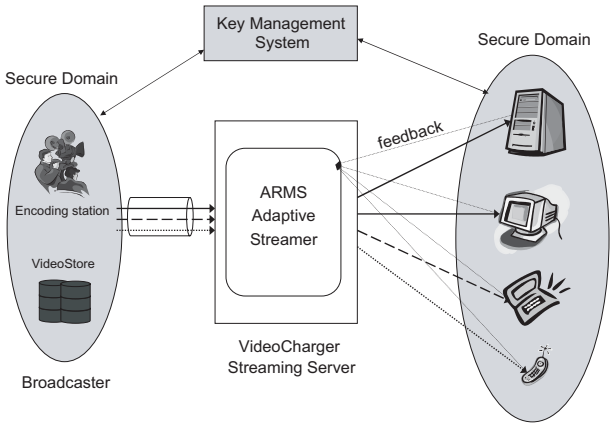

Figure 1: Architecture of a End-to-end Secure Streaming System.

- **Broadcaster** : The broadcaster encodes and broadcasts live video. Once a scheduled event begins, the ARMS broadcaster begins passing raw audio/video input through a bank of MPEG-4 encoders to produce encoded streams in multiple resolutions. These are then passed through an encryption module which keeps all the necessary streaming headers in the clear and only encrypts the content in secure containers as per the ISMA encryption standard [1], with our extensions for adaptivity (See [2] for details). Finally the data is packetized for transport in the RTP format. The particular payload format used depends on the media type and whether the data is encrypted or not. The broadcaster may optionally also write content to disk so that it may later be played back in a video-on-demand (VOD) scenario.

  The broadcaster is highly configurable and the configuration parameters are passed in a format that closely follows the MPEG-4 high-level textual format"XMT-O", and is extended in places for ARMS features.

- **VideoStore** : This component uses standard ISMA MPEG-4 hinting to store the different stream resolutions under different hint-tracks. An intelligent content-preparation tool analyses the content, and adds meta information to the hint-tracks so that the streaming server can react effectively to fluctuations in bandwidth and packet-loss characteristics. Once the content is packaged, it may be encrypted for secure streaming. Here again, the hinting information as well as the metadata are kept in the clear while the media content is encrypted. Such content is MPEG-4 compliant and can be streamed by any compliant streaming server. However, only servers with the ARMS capability will be able to switch among tracks and adapt to changing client conditions.

- **Streaming Server** : The Streaming server is setup for broadcast using the SDP file (IETF RFC 2327) generated by the Broadcaster, describing the session. The encryption is transparent to the server and in the live case, it receives RTP packets from the broadcaster over one of many different transports – (i) multiplexed in one TCP channel in the RTSP-interleave format, (ii) over multiple independent UDP unicast channels or (iii) over multiple independent UDP multicast channels. In each case, a unique channel number identifies packets belonging to a particular encoding.

  The server has a monitoring component which measures the available bandwidth to each of its clients. When the RTP/UDP protocol is used to stream content, the client sends status messages in the form of Real-time Control Protocol (RTCP) receiver reports. The indicated packet-loss ratio in these reports is used to compute the TCP-friendly rate to stream at. When HTTP or TCP is used to stream content, feedback is in the form of TCP congestion indicators, observed in terms of application buffer status. If the buffer exceeds a certain threshold, the server switches the client to a lower resolution. If there is sufficient bandwidth however, it probes the connection to the client periodically with duplicate packets to determine if it can transition the client to a higher bandwidth stream. Besides this automatic adaptation mechanism, the monitor component may also receives policy-based directives through the server API to control the bandwidth to any client. In this case, the server forwards the most suitable channel to the client.

  The server also aggregates client streaming statistics and feeds back to an upstream server and ultimately to the Broadcaster. Based on the gathered statistics, the Broadcaster recomputes its encoding parameters and may adjust the encoding rates and the number of different encoded versions.

- **Client** : The client receives and renders the streams while also providing reception feedback to the server. The client can be configured to receive MPEG-4 data in RTP packets using any one of many transports such as – interleaved-RTSP over TCP, interleaved-RTSP over HTTP or UDP. The Videocharger server is capable of streaming MPEG-4 to any standards compliant client such as Apple QuickTime 6, Philips player, Cisco player and the IBM player. Among these, at the time of this writing, only the IBM player implements the ISMA decryption standard.

3. STRUCTURE OF THE DEMO

In this demonstration, we shall illustrate the working of the system in the following ways :

1. Adaptive streaming of a multiply-encoded, secured and unsecured, live-stream : This demonstrates the capability of the server to adapt a stream to changing network conditions automatically. The content may be encrypted or not.

2. Secure and adaptive video-on-demand streaming : In this case, we demonstrate that the video can be paused, seeked and played back adaptively while still in the secure domain.

3. Selective encryption of substreams : The broadcaster can be directed to encrypt different sub-streams such as audio and video or different tracks in the presentation with different keys. Supplying the appropriate key will decrypt the corresponding stream at the client.

4. Policy-based adaptation : Using this, we demonstrate the capability to control the bandwidth usage of a single client or a group of clients using an administrative interface. This is an important requirement for enterprises where business data may have to take precedence over streaming media, in a shared infrastructure. Policy directives may also be used to control the aggregate bandwidth usage of the server.

4. REFERENCES

[1] Internet Streaming Media Alliance Implementation Specification, Version 1.0. *Internet Streaming Media Alliance*, Aug 2001.

[2] C. Venkatramani, P. Westerink, O. Verscheure, and P. Frossard. Securing Media for Adaptive Streaming. *ACM Conference on Multimedia*, Nov. 2003.