PROTECTING ADAPTIVE MULTIMEDIA DELIVERY AND ADAPTATION USING PROXY BASED APPROACH

Ahmed Reda Kaced
Graduate School of Telecommunications, Computer Sciences and Networks Department
37-39 Rue Dareau 75014, Paris - France

Jean-Claude Moissinac
Graduate School of Telecommunications, Computer Sciences and Networks Department
37-39 Rue Dareau 75014, Paris - France

Keywords: Adaptive multimedia, encryption, digital signatures, content adaptation, multimedia security.

Abstract: By breaking the end-to-end nature of the communication, proxies render the task of providing end-to-end security much harder or even impossible in some cases. In this paper, we will address the questions of when and how end-to-end security, like confidentiality and authenticity can be preserved, in a multimedia content delivery platform, when having one or more adaptation proxies in the data path.

We describe SEMAFOR, a platform for protecting adaptive multimedia content delivery in heterogeneous environments. SEMAFOR aims to deliver an end-to-end authenticity of original content exchanged in a heterogeneous network while allowing content adaptation by intermediary proxies between the content transmitter and the final users. Adaptation and authentication management are done by the intermediary proxies, transparently to connected hosts, which totally make abstraction of these processes.

SEMAFOR provides AMCA a new content authentication based on multi-hop signature scheme using a Merkle Hash Tree, and XSST a secured transaction protocol that gives securely exchanges of transactions in SEMAFOR.

1 INTRODUCTION

Today's multimedia networks and multimedia communication platforms are becoming increasingly heterogeneous, mostly due to the growth of mobile computing. Users are accessing multimedia data with an increasing number of different types of devices like mobile phones, PDAs, Laptops and Desktop computers, with very different capabilities. this makes difficult to send the same content to all the end users.

Adaptation proxies can help to overcome the problem of heterogeneity. Generally, a proxy is an active intermediary placed between two communication endpoints, typically a client and a server. Such an intermediary can be used to adapt a communication session to the type of clients and networks involved either by doing content adaptation or network protocol enhancement. Figure 1 sketches the basic architecture for a proxy-based adaptation approach.

The main problem of using proxies is that the end-to-end nature of the communication is broken. This leads to some severe security problems. One of the main questions that arise is how content adaptation by intermediaries can be done when end-to-end security is required. For example, how can a proxy transcode a data stream if it is encrypted? How can end-to-end data integrity and authentication be provided if the proxy needs to alter the data in transit?

In order to address the end-to-end security of adaptive multimedia content delivery challenges, we proposed in (Kaced and Moissinac, 2006a) AMCA, a new content authentication scheme which allows proxy to adapt dynamically a multimedia document by deleting or adding parts, preserving the client capacity to verify the original signature. So, in this paper we give an overview of SEMAFOR, our multimedia content delivery platform which uses a proxy-based approach to perform adaptation operations for the exchanged content and implements AMCA to sign this later.

This paper is outlined as follows. In section 2 we give an overview of SEMAFOR's architecture, we give also a description of the two main contributions of SEMAFOR, AMCA, our content authentication scheme, and XSST the transaction security protocol. After that, section 3 presents briefly some of the related works to our present work, and we finish by a conclusion in section 4.

Reda Kaced A. and Moissinac J. (2006).
PROTECTING ADAPTIVE MULTIMEDIA DELIVERY AND ADAPTATION USING PROXY BASED APPROACH.
In *Proceedings of the International Conference on Security and Cryptography*, pages 87-90
DOI: 10.5220/0002104500870090

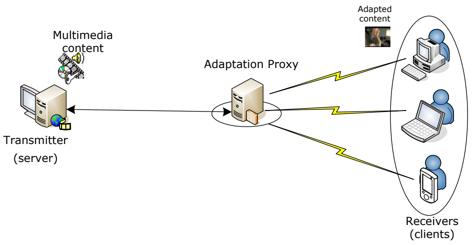

Figure 1: Basic architecture of a proxy-side adaptation.

2 BASIC FRAMEWORK

In the light of the above problems, we propose a SEMAFOR framework with a content authentication scheme called AMCA. Figure 2 shows the architecture of SEMAFOR, and describes the organization of the content delivery system.

The basic framework for our adaptive content delivery system consists of three main parts: client, proxy and server, each part consists of different modules.

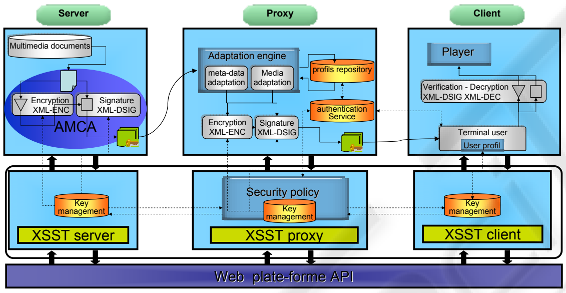

Figure 2: SEMAFOR architecture.

The Server is the sender of the multimedia content. It consists of two main modules, *AMCA* to sign the flow and *XSST Server* to send it. In a content delivery process, it's main task is to sign the flow using AMCA, generating a data structure which is sent to the client using the XSST-Server module.

The Client represents the end-user which receives the adapted document. It have the necessary tools needed to verify the authenticity of the received content.

The Proxy is an active intermediary placed between the server and the client. It is used to adapt the exchanged content to the type of clients and to secure the communication session by doing content adaptation and encryption. Modules implemented at the proxy-side are adaptation engine, that synchronizes adaptation operations, and XSST-Proxy module which secures communications with the other sides.

When the client requests a document from the server, the proxy makes request to the server on behalf of the client. The server signs the document to send using AMCA, encrypts it using XSST-Server, and sends it to the client(saying that it allows some proxies to adapt its content). The proxy, intercepts the reply from the server, decides on and performs the adaptation after having decrypted the document (or synchronizes adaptation on several proxies). Then it sends the transformed content on to the client. Finally, the client decrypts the received data using XSST-Client, verifies the signature then plays the multimedia document.

In the following, we describe AMCA and XSST modules that we use in SEMAFOR framework.

2.1 AMCA

Giving a secure delivery process of an adaptive multimedia starts by securing the content itself. So, to ensure an end-to-end authentication of the exchanged flow, we proposed AMCA (*Adaptive Multimedia Content Authentication*), a signature scheme based on a binary tree representation of the multimedia flow. Where each media object of the content is represented by a leaf of a binary tree, while new leaves (which we call *Freeleaves*) are added to ensure the insertion of adapted media.

The binary tree resulted is associated to a Merkle Hash Tree (MHT) (Merkle, 1990), for the purpose of allowing adaptation operation in the signed content.

A Merkle Hash Tree is a complete binary tree equipped with a candidate one-way function $hash$ and an assignment ψ , which maps the set of nodes to the set of media objects: $n \rightarrow \psi\{n\} \in \{m_1, m_2, \ldots, m_k\}$. And for each two child nodes, n_{left} and n_{right}, of any interior node, n_{parent}, the assignment ψ is required to satisfy

$$\psi\left(n_{parent}\right) = hash\left(\psi\left(n_{left}\right) \parallel \psi\left(n_{right}\right)\right)$$

We describe in the following the signature and verification schemes used in AMCA, and we talk after about the *freeleaves* that allow the dynamic insertion of media elements in the binary tree.

2.1.1 Signature

Let $\mathcal{T}$ denote the server and let $\mathcal{R}$ denote the client who verifies the signature. The data pass through a proxy $\mathcal{P}$. We assume the existence of an open or closed public-key infrastructure where $\mathcal{T}$ has key pair (K_p, K_s). Here K_s is $\mathcal{T}$'s private key for computing a traditional signature on a message, and K_p is the public verification key. $Sign(K_s, M)$ denotes an algorithm that outputs a signature φ on message M

under signing key K_s and $Verif(K_p, M, \varphi)$ denotes the verification algorithm. $\mathcal{P}_i$ need not know K_p or K_s. $\mathcal{T}$ and $\mathcal{R}$ may also use a symmetric key (which proxies need not know).

Let *M* denote the initial content that is composed by a meta-data description (m_1) and $n-1$ media objects($m_2, m_3, \ldots, m_n$). Where convenient, we assume *n* is a power of 2. In our scheme, adapting a media object means remove the corresponding leaf in the binary tree representation of the content and replace it by a new leaf.The intermediary may also choose to add a new media object or remove one.

Let *h()* denote a collision resistant hash function that takes as input a leaf *l* and a (fixed and publicly known) *v*-bit initialization vector (*IV*), and produces a *v*-bit output.

To process flow authentication, the framework consists of three main steps: document preparation and signature by transmitter, document adaptation by proxies and document verification by end users. We describe them as follows:

Preparation The meta-data document is first parsed to generate an adaptive format according to the server policy. To build an MHT from the original content, each media is associated to a leaf, and for each adaptive media object we join a *Freeleaf* as sibling. The Merkle tree associated with *M* is a balanced binary tree in which each node *v* is assigned a value ψ*(v)*.

To sign *M*, the transmitter computes the root value Ω of the MHT associated with *M*. The signature is φ=$Sign(K_s$,Ω).

Adaptation On receiving the protected document, a proxy can apply adaptation operations to fit the document into some different format. as we said above, the adaptation operation for a media in our scheme is performed by deleting this media in the MHT and inserting the adapted format as new leaf. Deletion and insertion by the intermediaries are ensured like follows:

Deletion is supported by supplying some extra verification data so that the verifier can still compute the root of the MHT, as we now describe. First, let Mi denote the transformed data after the removal of blocks.

Insertion of an adapted media by the intermediary is done using *freeleaves*. We use conventional public-key signatures, $\mathcal{S}$ places $\mathcal{P}$'s public key, or instructions on where to retrieve it, in the *freeleaf*. $\mathcal{S}$ then creates a MHT digest and signs as described above. $\mathcal{P}$, in turn, attaches its content and signs it separately. $\mathcal{R}$ checks the validity of both signatures.

Verification The verification procedure is actually reversing the above process. Suppose a receiver retrieves a content as well as its authentication data from some proxy. It then analyze it. With recovered authentication data, the receiver can verify the stream integrity and signatures with proper aggregate signature verification scheme.

The verification process includes unpacking, decoding and verifying. At least k out of *n* initial media objects should be received in order to recover the authentication data. Suppose *k* media objects $m'_1, m'_2, \ldots, m'_k$ are received successfully.

1. For m$'_j$(1≤ $j \le k$), computes ψ(m'_j)=*h(IV,* m'_j*)*.
2. Let $\{\varpi'_1, \varpi'_2, \ldots, \varpi'_k\}$ the set of results. If any pair correspond to siblings, replace the pair with their hash (which corresponds to their MHT parent). Repeat step 2 until only one value remains. call it Ω'.
3. Finally run $Verif(K_p, \Omega', \varphi)$.

We can see that Ω = Ω', from which it is easy to see why the above algorithm works. If one has all the initial media objects, then the above procedure is the standard algorithm for computing the MHT root. Now, observe that whenever $\mathcal{R}$ receives some hashes $\varpi_1, \varpi_2 \ldots, \varpi_n$ these come from $\mathcal{P}$ running the same algorithm on the subset of missing frames. Therefore, $\mathcal{P}$and $\mathcal{R}$ have together run the algorithm on all *n* blocks which yields the Merkle root value.

2.2 XSST

In order to ensure a fully secured service for adaptive multimedia content delivery in SEMAFOR, we have to ensure secure transactions between all parts of the platform. In other words, all communications have to be authenticated and encrypted.

For this purpose, we proposed XSST *(Xml Secure SEMAFOR Transaction)* providing the user with guarantee on the confidentiality, integrity, authentication and non-repudiation. XSST consists of three major components: XSST Client, XSST Proxy and XSST Server. It will firstly establish a secure tunnel for a real transaction before this transaction takes place. This secure tunnel is established by using a public-key based key exchange between the XSST proxy and XSST server in order to derive a unique secret key that can then be used to ensure data integrity and confidentiality throughout the session. XSST allows both Proxy and Server sign the messages, this signature can be generated directly by the proxy and server, or by other applications which the XSST interacts with. More informations about XSST can be found in (Kaced and Moissinac, 2006b).

Figure 3 illustrates the operations between the multimedia server $\mathcal{S}$ and the proxy $\mathcal{P}$, and the operations between the proxy $\mathcal{P}$ and the client $\mathcal{C}$ for requesting and adapting the multimedia data.

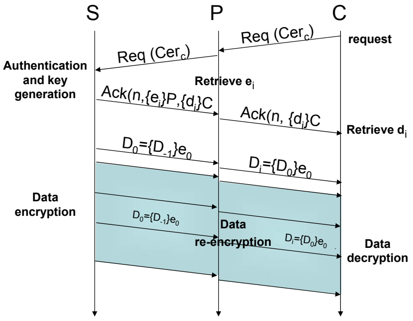

Figure 3: Operations between the source server $\mathcal{S}$ and the proxy $\mathcal{P}$, and the client $\mathcal{C}$.

3 RELATED WORKS

Various research works in proxy-based multimedia adaptation approach have been made in this area. Solutions proposed by (Johanson, 2001), Layaida (Layaida et al., 2004), use a proxy based adaptation for continuous data flow transcoding, other projects use a multi-proxy adaptation, *Appat platform* proposed by Lapeyre (Lapayre and Renard, 2005), *ubiQoS* presented by Bellavista (Bellavista et al., 2004). Unfortunately, these existing adaptation platforms do not address security and authentication of exchanged flows.

However, there are a small set of papers emphasizing security issues in a proxy-based approach. Recently, Gentry proposed in (Gentry et al., 2005) two new provably secure schemes that ensure secure streaming media authentication with adaptive proxies. (Li et al., 2004) proposed a secure MPEG-4 stream authentication scheme that allows a proxy flexibly manipulate streaming packets while they can still be verified by the final receiver. Suzuki in (Suzuki et al., 2004) proposed a multimedia content delivery system that protects the end-to-end authenticity of adaptive multimedia. However, they had the same problems; they only considered generic multimedia content simply as message blocks instead of real tree representation format like MPEG-21 or SMIL. Moreover most of them did not address the multi-proxy adaptation authentication problem.

4 CONCLUSION

A framework for securing an adaptive content delivery in heterogeneous network environments has been presented. To achieve this goal, we used the conventional extension of the client-server model to a client-proxy-server model.

The different aspects of system technologies include content authentication modules for signing the exchanged flow and a transaction encryption protocol for securing communication in the platform.

The content authentication scheme uses a multi-time signature scheme which is based on MHT technique. The concept of FreeLeaves was introduced to allow adaptation operations by the intermediary proxies. Our experiences suggest that a MHT-based signature scheme is both a feasible and practical solution to this problem.

While we have only touched upon some of the issues necessary for deploying Merkle hash tree based authentication schemes for adaptive multimedia content in this short paper, an extended article is available (Kaced and Moissinac, 2006b) that provides more details, including: a deeper look at Merkle tree maintenance, algorithms for the verification procedures, and experiments and experiences with an actual implementation.

REFERENCES

Bellavista, P., Stefanelli, C., and Tortonesi, M. (2004). The ubiqoS middleware for audio streaming to bluetooth devices. In *MobiQuitous*. IEEE Computer Society.

Gentry, C., Hevia, A., Jain, R., Kawahara, T., and Ramzan, Z. (2005). End-to-end security in the presence of intelligent data adapting proxies: the case of authenticating.

Johanson, M. (2001). An rtp to http video gateway. pages 499 – 503.

Kaced, A. R. and Moissinac, J. C. (2006a). Multimedia content authentication for proxy-side adaptation. In *Proceedings of the IEEE International Conference on Digital Telecommunications, ICDT 06*.

Kaced, A. R. and Moissinac, J. C. (2006b). Semafor: a framework for authentication of adaptive multimedia content and delivery for heterogeneous networks. In *Proceedings of ICISP 06*.

Lapayre, J.-C. and Renard, F. (2005). Appat: A new platform to perform global adaptation. In *the 1st IEEE Int. Conf. DFMA'2005*, France.

Layaida, O., Attalah, S. B., and Hagimont, D. (2004). Adaptive media streaming using self-reconfigurable proxies. In *7th IEEE International Conference, HSNMC 2004*. Springer.

Li, T., Wu, Y., Ma, D., Zhu, H., and Deng, R. H. (2004). Flexible verification of MPEG-4 stream in peer-to-peer CDN. In *ICICS*.

Merkle, R. (1990). A certified digital signature. In *CRYPTO: Proceedings of Crypto 1990*.

Suzuki, T., Ramzan, Z., Fujimoto, H., Gentry, C., Nakayama, T., and Jain, R. (2004). A system for end-to-end authentication of adaptive multimedia content.