XML Security in the Next Generation Optical Disc Context

Gopakumar G. Nair[1], Ajeesh Gopalakrishnan[1], Sjouke Mauw[1], and Erik Moll[2]

[1] Eindhoven University of Technology (TU/e),
Eindhoven, The Netherlands
{G.gopakumar, A.gopalakrishnan, S.mauw}@tue.nl
[2] Philips Applied Technologies,
Eindhoven, The Netherlands
Erik.moll@philips.com

Abstract. The Extensible Markup Language (XML) is considered as the de facto standard for information processing and exchange on the Internet and in the enterprise services domain. It is widely regarded that XML has the potential of being an interoperable standard for interactive applications in the next generation connected Consumer Electronic devices. A key industry concern in using XML in CE devices is that how basic security requirements pertaining to the above said domain can be met. Notably, the standardization bodies of the Internet domain such as W3C and OASIS have defined specifications for cryptography-based security solutions using XML technology that is mainly aimed for web applications. This paper investigates and presents various scenarios where XML Security can be applied to markup based interactive applications in the context of a next generation Consumer Electronic Optical Disc Player. We conclude the paper by presenting a prototype establishing how these scenarios could be realized in practice.

1 Introduction

Until recently, the diverse and well-established domains of Personal Computers (PC), Web (Internet), Consumer Electronics (CE) and Broadcast domains have had their own autonomous realms of existence. Each of these domains spawned their own characteristic and individualistic ways of managing and doing things, with examples as diverse as the application specification to the very notion of interactivity. However, lately there has been considerable interest among these domain communities to share and adopt inter-domain best practices and knowledge. As an illustration, the content creators could create applications for one domain, which could be seamlessly integrated or be transferred to other domains. Such integration could provide new usage models in the CE optical disc domains [2]. As a fleshed out example, the content creators could author multi-domain interoperable applications which could be packaged in a disc and additional application extensions such as bonus materials, clips etc could be downloaded from a content server or a set top box in a home network, thus borrowing the ideas from Web and Broadcast domains. One of the possible candidates

W. Jonker and M. Petković (Eds.): SDM 2005, LNCS 3674, pp. 217 – 233, 2005.

for this cross-domain sharing is the XML and its related technologies, which entails the core theme of discourse in this paper.

XML is emerging as the de-facto standard for storing, managing, and communicating information on the Internet [1]. In addition, XML is the basis for markup applications and a wide range of XML based languages [7] for various web services. Several standardized interfaces, tools, techniques and their programming language bindings are available, both commercial and open source. This makes XML a serious contender for being considered as a standard for creating consumer interactive multimedia systems, the market where disc based systems mainly belong. A well-known example of such a standard is DVB-HTML [8], an XML based interactive application specification for Multimedia Home Platform [8] which has been existing for several years. With such pervasive and proven applications and usage scenarios of XML in a myriad of domains, it is without doubt a pick while considering the specification for Interactive Applications in next generation optical discs. Certainly, in combination with a procedural language, such as Java such a standard would open up new possibilities in bringing interactivity to such devices.

Next generation optical disc formats such as Blu-ray disc (BD) [2], High Definition (HD) DVD, and enhanced DVD (eDVD) [31] are reckoned as the natural successors of DVD as a medium for storage, playback and distribution of digital media [2].

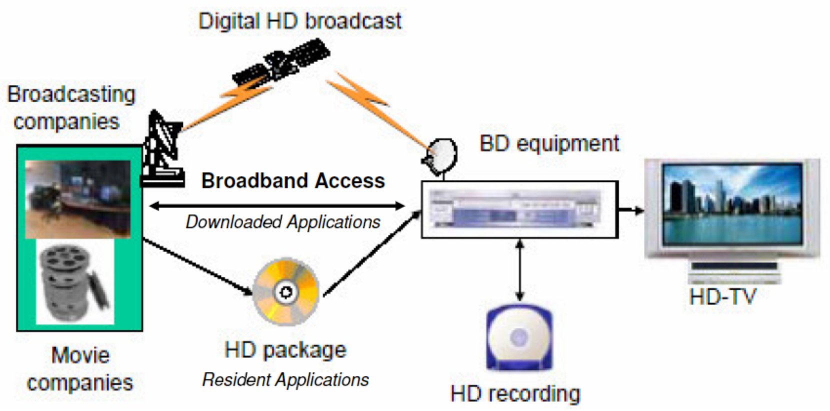

Fig. 1. End-to-End Usage Model (based on [1])

Figure 1 depicts the end-to-end usage model of the next generation optical discs based players in a consumer home. The movie companies distributes the High Definition (HD) content via optical discs as medium or via HD broadcast and the optical disc player equipment at the consumer home can playback the content on HD Televisions. As a consequence, the consumers get High Definition (HD) video experience, the content providers (movie companies) get the opportunity to store and distribute high quality videos and games and the independent content creators and vendors have the opportunity to provide value-added media based services. Additionally, due to the wide availability and growth of broadband connections, new Internet based usage models to download applications from content servers are foreseen for such devices due to the perceived characteristics of these devices to connect to Internet. In order to realize this, the next generation optical discs would need an interoperable interactive application specification with adequate considerations for security.

In the context of next generation optical disc players, careful consideration should go into the interactive application security issues while considering the usual issues of copy protection of audio and video content. In this case, the applications could also be copyrighted and could be subjected to malicious usage. To give an example, consider a malicious application loaded from an external server that could corrupt the local storage of the player. As another example, the user could try to create his/her own application, load to the system and try to access content where he has no access rights. The security mechanisms that could prevent such issues must be non-invasive to the users, should be capable of being applied easily by the content creators and be necessarily future proof. Additionally, the opted security mechanism should be flexible, interoperable, and widely supported with appropriate tools and libraries in order to be accepted by the manufacturers.

W3C [22] and OASIS [18], the major standardization bodies within the Internet domain, have been working on creating XML based security standards for web-based applications. We foresee that these standards can be applied with the XML based interactive applications for the next generation optical disc systems. In this paper we discuss which are these standards, what problems they can solve in a disc player context and how we can establish the end-to-end security. We also see whether these mechanisms are realizable in an embedded system context.

This paper first portrays a typical markup based Content Hierarchy depicting the Interactive Applications in the next generation optical discs. Further to that, we characterize the security profile of a connected player by applying analytical Threat Modeling and ascertaining the detailed security requirements for this paper from the Threat model, which will be used for the rest of the discussions. Various XML Security standards are presented with their proposed solutions for the above said security requirements. Additionally, an end-to-end security scenario is presented and a proposal of how all the security standards could be brought together to guarantee an end-to-end security solution is exposited. Finally, to substantiate the proposal, a prototype implementation is detailed on a next generation optical disc reference platform.

2 A Markup Based Content Hierarchy

Optical discs are intended to store digital content, the term, used to describe any kind of collection of functional work, artwork, or other creative content, copyrighted or otherwise distributed in an optical disc. In this section, we introduce the content hierarchy in the next generation optical discs; in particular, the markup based application hierarchy, which can be used for representing Interactive applications. The Interactive Application refers to a part of the overall content that can be executed by the optical disc player.

At the top of the content hierarchy (see Figure 2) is the *Interactive Cluster*, which is the generic representation of packaged content, including Video, Audio and markup Application. The *Interactive Cluster* contains several *Tracks*, which form chapters for *Video/Audio Playlist* [23] and optionally *manifest* (application). The playlists contain meta-information about the play items and refer to *Clip Information*, which ultimately links to the Mpeg-2 Transport Stream file [24]. It is the *Application Manifest* that represents the Interactive Application in the hierarchy and captures its essence.

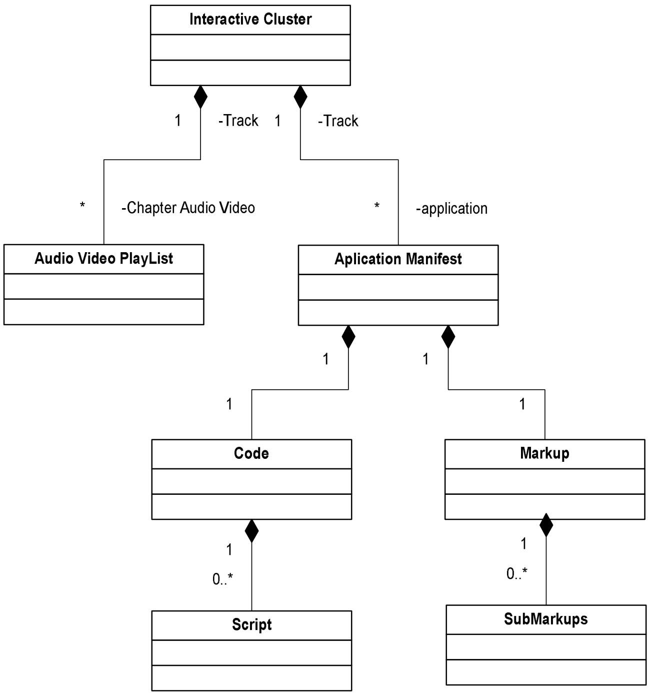

Fig. 2. An XML based content hierarchy

The manifest file consists of two distinct parts, namely the *Markup* and the *Code*. The Markup part captures the static composition of the application, which includes layout, timing model. The Code part provides flexibility by adding programmability and dynamics to the overall interactive experience. In turn the markup part could contain "*SubMarkups*" helping the separation of various characteristics of the application. For e.g. the layout can be captured in one SubMarkup and the timing issues in another. On the same lines, the code part can contain none or more scripts. As long as the overall structure and representation is respected, these subtle choices are entirely up to the discretion of the content author.

The choice of the markup for next generation optical discs may be from the following XML based languages, such as Synchronized Multimedia Integration Language (SMIL) [9], Scalable Vector Graphics (SVG) [26], Extended Hypertext Markup Language (XHTML) [27] and Extended Style Sheet Language (XSL) [28]. Additionally, ECMAScript [10] could be considered for the programmable part of the manifest.

3 Identifying Security in an Optical Disc Player Context

As pointed out in Section 1, next generation optical discs can be used to distribute HD video [3] content, along with interactive applications packaged in the disc and additional resources may be downloadable from an external location. The disc players can represent a myriad of devices ranging from Consumer Electronics devices to PC drives and mobile systems such as portable game stations. To get a complete picture of security requirements in such varied usage contexts, a complete threat characterization, and analysis is necessary. As a result, a Threat Modeling approach based on STRIDE [15] has been applied in order to make a methodical analysis of the security threats for optical disc based systems– especially with regard to the accession of interactive applications.

3.1 Threat Modeling for Next Generation Optical Disc Player

The Threat Model [12] provides us with a comprehensive list of threats to the application security and the various mitigation strategies that can be applied. We intend not to present the full results from the model [12], since the model per se is out of scope of this paper. Nevertheless, using the model we select a subset of the requirements and investigate how XML security mechanisms could be used to mitigate certain risks. In particular, the requirements of Authentication, Application Integrity, Content Secrecy, and Access control management ([4]) were under scrutiny.

Some of interesting inferences resulting from the investigative study of the model [12] about threats and their widely adapted mitigation strategies [4] are presented below in the context of next generation optical disc and players.

- *Authentication & Integrity*: The markup applications or resources loaded from the disc need to be authenticated by the player in order to guarantee that only trusted applications are executed. Additionally, the applications or their parts or downloaded resources [3] may need to pass integrity checks [4] to detect (malicious) tampering before being used by the disc player.
- *Encryption*: Applying encryption techniques [4] can allow content authors to avoid wiretapping (man-in-the-van attack) and application sources/resources protection. This is important in the context of markup and script applications because they are essentially verbose.
- *Key Management:* Key Management includes all the services pertaining to key handling, registration, revocation and updates of cryptographic public keys, which are used in authentication and encryption mechanisms. In particular, appropriate Key Management procedures [4] must be in place to circumvent the causes of illegal creation, exchange, repudiation, replacement, protection, storage, and usage of keys used in the scenarios of optical disc application authentication and encryption.
- *Access Control:* The access control mechanisms [4] allow the next generation optical disc player to give access rights to the markup based on certain predetermined policies. As a result, it can provide or restrict access to certain resources, as requested by the application author or under certain conditions.

In the subsequent sections, we see various XML security standards and examine their usage in satisfying the requirements (identified in Section 3.1) in an optical disc usage context.

4 Overview of the Applicable XML Based Security Mechanisms

In this section, an overview of the standardized XML security mechanisms, proposed by W3C and OASIS, is presented.

The issues of Authentication and Integrity identified in Section 3.1 can be mitigated by Digital Signatures, which can be used to verify the integrity of the Interactive Application or associated content assets. To this end, XML Digital Signature [5] proposes a specific syntax to represent a Digital Signature [4] over arbitrary digital content. Furthermore, the XML Digital Signature is in itself a well-formed XML document and carries all the information needed to process the signature, including the verification information. The XML Digital Signature specification also recommends a mechanism for signature creation and verification of XML based markup. It is useful for signing and verifying entire or portion of the markup, which may be of binary content and/or include multiple documents.

Another point identified for elaboration in Section 3.1 was the issue of encryption. This issue has been treated well by W3C, in particular with the standardization of the XML Encryption Syntax and Processing [6], which can be applied in the disc-player context to satisfy specific needs of application content protection and secrecy. XML Encryption can handle both XML and non-XML (e.g. binary) data, which makes it flexible to be used along with the interactive applications. A typical usage scenario foreseen in the above-mentioned systems is to encrypt markup applications residing in the disc along with the resources such as images and data. The player will decrypt the application and resources on execution of these markup applications. The content or referenced resources could be encrypted as well.

References [6] and [16] suggest that XML encryption can be done at various levels. The content could be encrypted and stored in parts or as a whole. This allows flexibility and better performance. A Player, for instance, can encrypt and store the high scores of a game in a local storage while keeping the general application markup unencrypted. When the game is being executed, the player needs to decrypt only the scores, which can be done in parallel to the execution of the markup.

Another advantage of XML encryption in ensuring confidentiality is that mechanisms such as Secure Sockets Layer (SSL)/Transport Layer Security (TLS) protocols only provide confidentiality while the information is in transit and not while it is stored at a server, but XML Encryption takes it one step further by maintaining the confidentiality of information, both while in transit as well as when stored. Notably, the secrecy is not dependent on the state or a particular session of the communicating parties.

The issue of Key Management could be dealt with using the XML Key Management Specification (XKMS) [33] from W3C. The XKMS helps manage the sharing of the public key realizing the possibility of signature verification and encrypting for recipients. The usage of XML based message formats for key management eliminates the need to support other specialized public key registration and management protocols for markup based interactive applications in the next generation optical discs.

In order to counter the Access control issue from Section 3.1, the XACML [19] Specification proposed by OASIS [18] provides access control mechanism for applications, based on assertions. This may allow content creators to add policies to request the disc player devices to provide certain rights to an application.

The mechanism defined by MHP [8] suggests the usage of XML based "*permission request*" files. In this case, the content provider can add the permission request file along with the markup as an attachment. This will be interpreted by the platform and will provide access rights to the application (e.g. rights to use return channel or rights to dial to a particular server). Based on the adopted policy, the platform can allow or reject the rights to the resources.

Having looked at the XML based security possibilities, we now broaden the discussion with an overview of comparison of XML based security mechanisms with other potential content download security mechanisms like OMA DRM (Open Mobile Alliance - Digital Rights Management) [34]. Reference [37] provides an interesting comparison between OMA DCF (DRM Content Format) extensions (see [35] [36]) and XML based security mechanisms on overhead and performance for data broadcast in mobile networks. The reference [37] suggests that XML based security incurs 2.5 to 5.1 times more overhead as compared to OMA DCF and performance wise the text based XML takes a back seat when compared to binary-based OMA DCF. Nevertheless, our scenario test runs using the developed prototype (see Section 8) convinced us that in the context of a consumer electronic device like optical disc player, this performance reduction while using XML based security would be within the allowable performance requirements. Additionally, the indicated overhead would also not be a significant issue, owing to the fact that the broadband Internet bandwidth (used by next generation disc players) is not as much of a concern when compared to the mobile over-the-air bandwidth, which the reference [37] refers to.

Additionally, the DVD (Digital Versatile Disc) Content Scrambling System (CSS) used on DVDs to encrypt media data thereby restricting the decoding to only licensed DVD players is less likely to be extended to downloaded application security scenarios in the next generation optical disc in lieu of differences in the nature of content data and usage scenarios. Particularly, the CSS is meant for the protection of digital content and lacks flexibility and scalability when extended to interactive applications. Moreover, CSS has been tampered.

In Section 5, we see various ways in which the XML Digital Signature is applied to interactive applications. In Section 6, we see how XML encryption and decryption is applied. Even though Key Management as a requirement was highlighted earlier, an example of XKMS application in the context of interactive application in optical discs has been left out of the discourse in this paper. Section 7 describes the order of integration of these two, along with additional mechanisms to provide order such that end-to-end security is ensured when applications are created and then later executed.

5 Applying XML Digital Signature in the End-to-End Usage

5.1 Global Scenario

Figure 3 examines the global scenario for usage of Digital Signature in the context of signing, transmission, and verification of Interactive Applications. In the previous sections, we introduced the notion of the player accessing the applications over the Internet in addition to accessing the applications on the pre-authored disc. Disc based applications are inherently trusted since they were authored into the disc by the content providers - provided the disc is authenticated [29]. The real security issue [12] lies with the interactive applications downloaded over the Internet and the Signing/Verification scenario identified in Figure 3 would facilitate in mitigation. Though the realization of the XML Digital Signature [5] in this section is addressed using examples from the over the Internet downloaded Blu-ray applications, the Digital Signature creation and verification can be extended to disc-resident disc-based Blu-ray applications too.

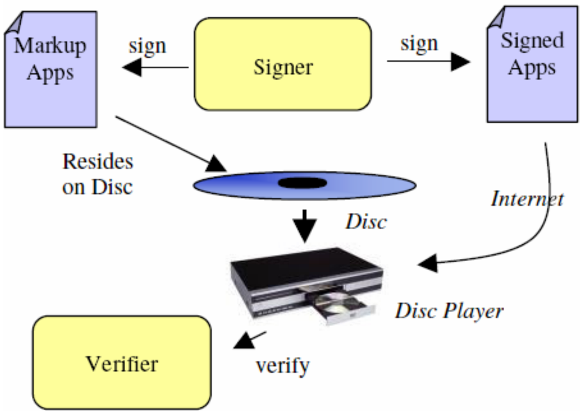

Fig. 3. Global Signing/Verification Scenario in Bluray

As seen in Figure 3, both at the content creators end and at the application authors' end, the applications can be digitally signed. When the player accesses any application from the Internet (e.g. Servers), it tries to authenticate the content by verifying the Digital Signature of the markup application. If the verification succeeds, the application is executed. In the case of signature verification failure, the application is barred from being executed. This implies that the player needs to have a Verifier component, which can carry out the signature (XML Signature) verification. Furthermore, the flexibility with this approach lies in the fact that this signing/verification mechanism can be applied in a variety of ways and levels.

5.2 Identified Signing/Verification Levels for Applications

To account for and prove the application of XML Digital Signature mechanism in next generation optical disc format, we look at the various sub-scenarios where the XML Signing and Verification can be achieved in the context of the interactive Applications as introduced in the "Content Hierarchy" section (see section 2). Even though these sub-scenarios can be extended with more detailed scenarios, here we only propose the general concept supported by examples here.

5.3 Signing/Verification at Interactive Cluster Level

We envision that the XML Digital Signature [5] can be applied at the level of Interactive Cluster (see Figure 2). Since the Interactive Cluster is Markup based, the XML Digital Signature can be used to sign/verify the Interactive Cluster in its entirety or can be used to sign/verify at Track (see Figure 2) Level. It is entirely up to the discretion of the Signer if (s)he wishes to sign the non-markup audio/video Content, which is nevertheless possible using XML Digital Signature. Since the main discourse is inclined towards Interactive Application authentication, a realization of selective Signing/Verification of application Track is hence commendable.

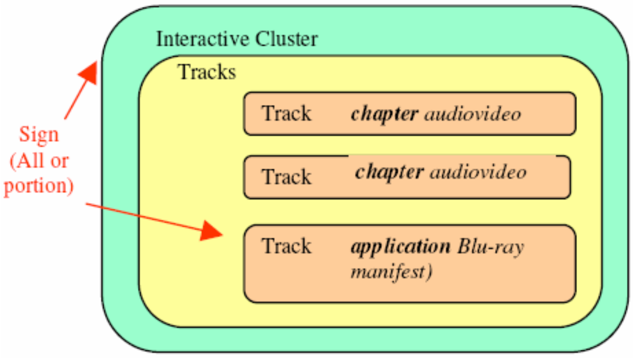

Fig. 4. Signing/Verification Scenarios in the Interactive Cluster Level

5.4 Signing/Verification at the Manifest Level

Taking the Signing and Verification one level deeper to the Manifest (see Figure 2) that forms part of the application Track, we notice that the control of authentication becomes much fine-grained or more granular. In this case, the choices available to the

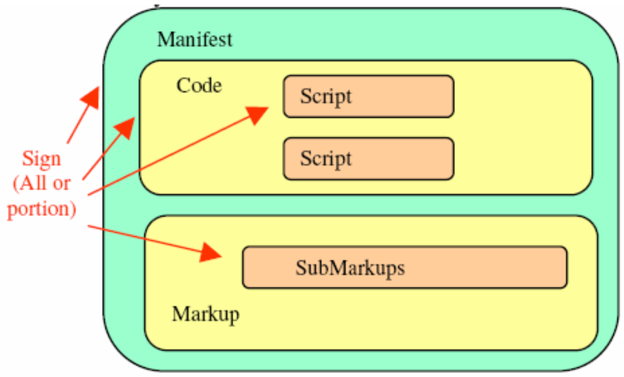

Fig. 5. Signing/Verification Scenario at the–Manifest Level

signer are quite large. (S)He can selectively sign only the Code or the Markup part (see Figure 2). Within the Code or Markup part itself, (s)he can choose to sign/verify only one of scripts or submarkups. The capability of the script to dynamically manipulate the Interactive Application makes it much more suited for authentication using XML Digital Signature [5]. Nevertheless, a maliciously tampered markup can be also detrimental to the Security of the Disc Player and the content.

From the above two example sub-scenarios we have seen that a number of Markup Items can qualify to be the target for XML Digital Signature. We refer to these as "Markup Targets".

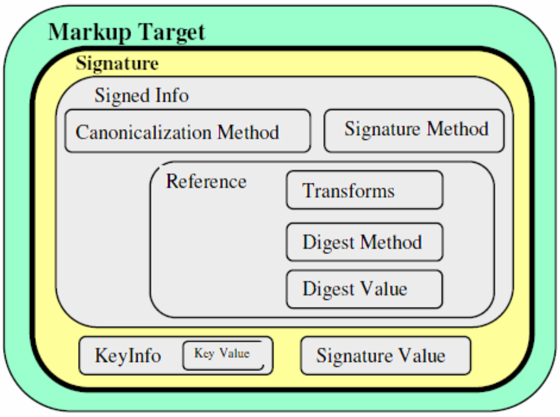

Fig. 6. Result of XML Signing on markup Targets

Figure 6 indicates the result of signing a Markup Target. The result of the signing process is the *Signature Markup* section that is demarcated in the figure by thick lines. This signature can be *enveloped* or *enveloping* [5] based on whether the markup target is parent or child to the "Signature". The signature can also be detached [5] if the target has no parent-child relationship to "Signature" element. The choice resides entirely with the content signer. The fact that XML based markups allows syntactic variations while remaining semantically equivalent and the nature of hash functions to be sensitive to syntax variations, calls for the application of *canonicalization* (XML-C14N)[*32*] to the signature to remove syntactic differences from semantically equivalent XML documents.

Reference [4] gives the concepts of Digital Signatures and the cryptographic algorithms which can be used for signatures. Reference [16] gives description of the markup tags for digital signature.

An additional concept used within the context of Digital Signature is the Certificate Handling [8], which uses a digital signature for public key [8] bindings.

5.5 Certificate Based Authentication

XML Digital Signature supports the insertion of digital certificates along with the signatures and provides syntax for the certificates present within the markups or referred to by the markups, which will be useful for the players to verify the authenticity of the keys. Reference [8] suggests a mechanism for the verification of certificates leading to a trusted root certificate within the player. It should be noted that the XML Digital Signature [5] could be used for such verification.

6 Applying XML Encryption to Markups

Having introduced the notion of XML based Encryption/ Decryption in the context of Interactive Applications in Section 4, we now take a look at how they can be applied in practice. As mentioned earlier, the XML Encryption could be used to encrypt both

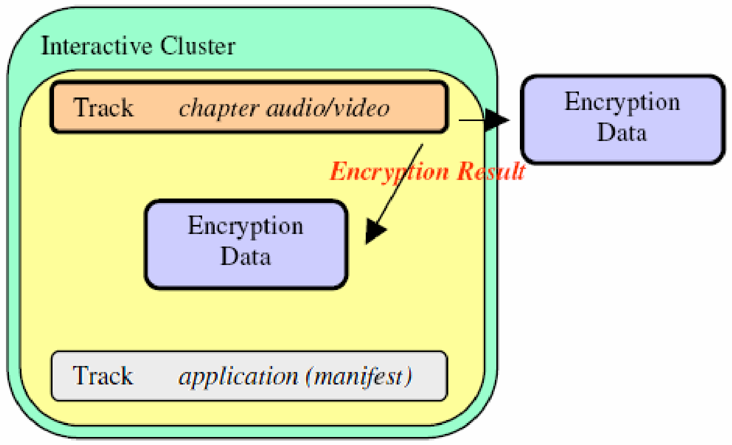

Fig. 7. Result of XML Encryption on Track Target

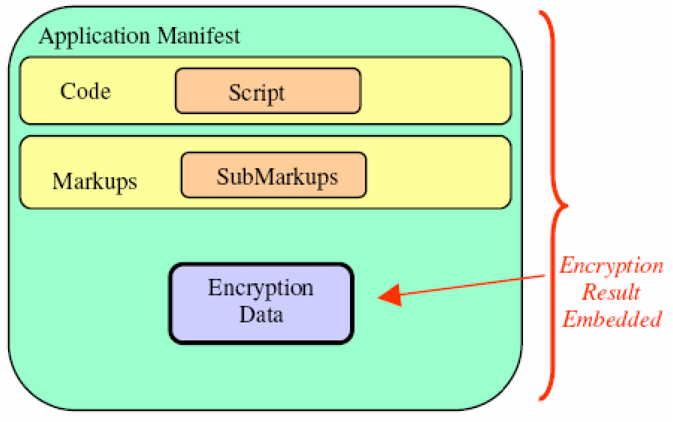

Fig. 8. Result of XML Encryption on Manifest Target

XML markup based or non-XML based content. This brings us two different scenarios for encryption of Interactive Applications, namely, the Encryption of the *Track Target* and the Encryption of *Manifest Target*.

Figure 7 illustrates the result of signing non-markup content, i.e. a chapter's audio/video track (see Figure 2) that is an Audio/Video Play list [23]. The result of encryption of non-markup content in this case is an "Encryption Data" [6], which is either created and embedded in the Interactive Cluster or jettisoned as a separate Markup.

However the scenario is different when markup content is encrypted. Figure 8 shows such a scenario where the manifest (see Figure 2) is encrypted.

In this case, the signing of manifest, an XML based markup, results in the Encryption Data being embedded in the manifest itself. For more details on XML Encryption and various elements within the Encryption Data refer to [5].

7 Providing End to End Security

In this section we explore the possibility of integrating the previously identified security mechanisms in order to provide end-to-end security. In particular, we consider how the above-mentioned XML security mechanisms works together when a typical application is created, packaged or transmitted and later executed in a CE optical disc player. Figure 9 shows an example of how these mechanisms work together when an application is created, packaged and transmitted via Internet. Such an order can be used when applications are created and packaged in a disc and later launched by a player.

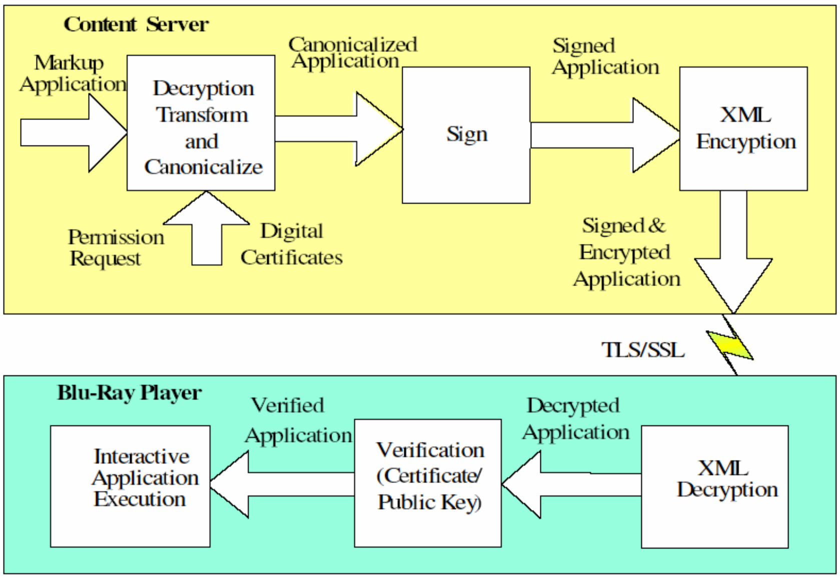

Fig. 9. Encryption and decryption process- end to end

To ensure the order between content encryption and signing, W3C specifies the Decryption Transform [21], which provides the signers with mechanisms specifying the order of signing and encryption. The resulting application contains sufficient information in the form of additional markup that enables the player to identify how the application needs to be decrypted and verified. The content creators also can add XML based permission request file, which requests permissions from the player to allow access to certain Player resources (e.g. access to graphics plane or writing to local storage). Furthermore, the XKMS [33] based Key Management could be used to convey key registrations and information requests to any “trusted source (trust server)” and to convey responses back from the server. Note that SSL/TLS mechanisms could be used for mutual authentication and secrecy between server and the player when applications are transmitted over the network.

8 Prototype

This section aims to substantiate the proposals mentioned in the previous sections with a prototype on a reference platform as a proof of concept. We chose Blu-ray as our optical disc format and aimed at prototyping the concepts mentioned above on a Blu-ray based optical disc platform.

For the choice of the markup target, we chose *Application Manifest* (see Figure 10), which represents the Interactive application. This choice is guided by the possibility to demonstrate the flexibility of the application of XML Digital Signature in a Blu-ray disc player.

8.1 Realizing the Reference Blu-Ray Interactive Application

The first and foremost need in the prototyping stage is to start with a reference Interactive Application with Blu-ray as a target system, along with appropriate choice of the markup target, the scripts and the sub-markups.

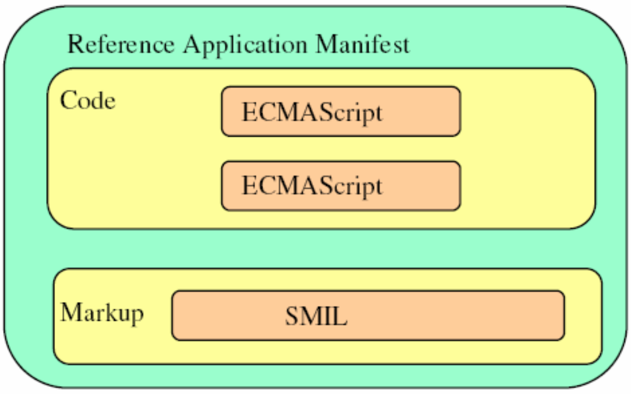

Fig. 10. Choices of Reference Blu-ray Markup Target, Script and SubMarkups

For the prototype we choose to represent the script with ECMAScript [10], a scripting programming language created to capture the common core language elements of both Javascript[11] and JScript[25]. For the timing and layout markup, we chose SMIL [9], a W3C [22] recommendation for describing multimedia presentations based on XML. It defines timing markup, layout markup, animations, visual transitions, and media embedding, among other things.

8.2 XML Security Library Implementations

At the time of prototyping available XML Security Libraries were from Apache Security Project [13] and IBM Alphaworks [14]. We selected the Apache XML Security libraries since it provides flexible licensing options for prototyping. At the time of the prototyping, two XML security implementations were available in Java and C++ from Apache. Each of these library flavors stood out as potentials for our cause. Our choice of Java was guided by the need of quick prototyping and stability of the available Java based Libraries over C++ based libraries. Apache XML security uses Java Cryptography Extension (JCE) and this was included in the prototype, along with the bundled Sun cryptography provider [31].

8.3 Reference Platform

We chose Linux based Blu-ray platform and created a prototype as discussed in this section.

Figure 11 shows the layered view of the software architecture.

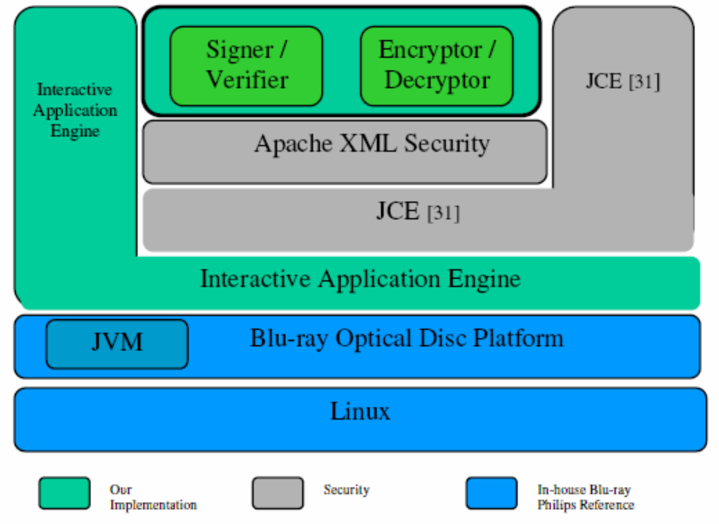
Interactive Application Engine
Signer / Verifier
Encryptor / Decryptor
JCE [31]
Apache XML Security
JCE [31]
Interactive Application Engine
JVM
Blu-ray Optical Disc Platform
Linux
Our Implementation
Security
In-house Blu-ray Philips Reference

Fig. 11. Software Architecture for feasibility – Layered View

The Interactive Application Engine is the main component, which has access to the Interactive Cluster (see Figure 2) and is responsible for getting the application contents decrypted, if encrypted, and verified, if signed. In addition to the Verifier and Decryptor, a Signer and an Encryptor component were created to fulfill end-to-end requirements, which enabled the signing and encryption of Application Content.

The resulting prototype substantiated the pragmatic dimension of the proposal for using XML Security in the next generation optical discs. This prototype also demonstrated that XML based security and Interactive Application Engine (see Figure 11) can exist independent of the type the Disc format, be it Blu-ray disc [2], High Definition-DVD and enhanced DVD (eDVD) [31]. In addition, one of the main highlights of the proposal was the overall development time of the prototype, which took no more than 4 man weeks to complete. This demonstrates the simplicity and flexibility in the case of implementation of the proposal.

9 Conclusions and Future Work

We have seen that XML security offers a standard and interoperable mechanism that can be used by content providers to accommodate necessary security requirements for next generation optical discs. The content authors may use the flexibility of partially signing or encrypting the applications. For player platforms, this flexibility translates into better performance. The standard looks mature and implementations are available in Java and C++. The prototype enabled us to conclude the feasibility of proposal in an embedded platform, although we did not derive any performance constraints. We conclude that the usage of XML Security as a mechanism for markup based interactive applications can alleviate the security concerns pertaining to application security for content providers and CE manufacturers.

In lieu of future work, to expand the scope of XML Security mechanisms we envision that XRML [20], an XML based rights management language proposed by OASIS [18], to express digital rights for the usage of markup-based applications and resources, can be investigated for digital rights management in the next generation disc player context. Additionally, we intend to extend the current prototype with XML based Key Management [33].

Additionally, the current prototype could be extended to other underlying platforms, with respect to optical disc formats, Operating Systems and Hardware Platforms to account for the interoperability. Next, a scalable Interactive Application Engine library could be developed enabling ease of deployment. Finally, a performance model with comprehensive performance study measurements could be done for identifying and tuning the system resources needed for interpretation of the markup applications, and the associated XML based security.

References

1. Tim Bray et al., Extensible Markup Language (XML) 1.0 (Third Edition), World Wide Web Consortium (W3C) Recommendation. www.w3.org/TR/REC-xml/
2. Blu-ray Disc Association (BDA), Blu-ray Disc –Application Specification, BD-J Baseline Application Model Definition for BD-ROM – March 2005. www.bluraydisc.com

3. Blu-ray Disc Association (BDA), White Paper: Blu-ray Disc Format - General, August 2004. www.bluraydisc.com
4. Bruce Schneier, Applied Cryptography, Wiley, Second Edition, 1995, ISBN: 0471117099.
5. Mark Bartel et al., XMLDigSig - XML-Signature Syntax and Processing, World Wide Web Consortium (W3C) Recommendation 12 February 2002. www.w3.org/TR/2002/REC-xmldsig-core-20020212/
6. Takeshi Imamura et al., XML Encryption Syntax and Processing, World Wide Web Consortium (W3C) Recommendation 10 December 2002. www.w3.org/TR/2002/REC-xmlenc-core-20021210/
7. Uche Ogbuji, A survey of XML standards. www-106.ibm.com/developerworks/xml/library/x-stand1.html
8. European Telecommunications Standards Institute (ETSI), Digital Video Broadcasting (DVB), Multimedia Home Platform 1.2.1, ETSI TS 102 812 V1.2.1 (2003-06).
9. Jeff Ayars et al., Synchronized Multimedia Integration Language (SMIL 2.0), World Wide Web Consortium (W3C) Recommendation, 07 January 2005. www.w3.org/TR/2005/REC-SMIL2-20050107/
10. European Computing Manufacturing Association (ECMA), ECMAScript Language Specification, Standard ECMA-262 ISO/IEC 16262, 3rd Edition - December 1999.
11. Mozilla, JavaScript 2.0 Specifications. www.mozilla.org/js/language/js20/
12. G Gopakumar, A Gopalakrishnan, Threat Model based on STRIDE, OOTI Project Report 2005, TU/e.
13. Apache Security. xml.apache.org/security/
14. IBM Alphaworks, XML Security Suite. ww.alphaworks.ibm.com/tech/xmlsecuritysuite
15. Frank Swiderski et al., Threat Modeling, Microsoft Press 2004 ISBN: 0-7536-1991-3
16. Blake Dournaee, XML Security, RSA Press, McGraw-Hill/Osborne 2002, ISBN: 0-07-219399-9.
17. Balal Siddiqui, Exploring XML Security. www-106.ibm.com/developerworks/xml/library/x-encrypt/
18. Organization for the Advancement of Structured Information Standards (OASIS). www.oasis-open.org
19. Tim Moses et al., Extensible Access Control Markup Language Specification (XACML) version 2.0, Organization for the Advancement of Structured Information Standards (OASIS) Committee draft 04, 6 Dec 2004. docs.oasis-open.org/xacml/access_control-xacml-2_0-core-spec-cd-04.pdf
20. Simon Godik, Tim Moses et al., Xtensible Digital Rights Markup Language (XRML), OASIS Standard, 18 February 2003. www.oasis-open.org/committees/xacml/repository/
21. Decryption Transform for XML Signature, World Wide Web Consortium (W3C) Recommendation, 10 December 2002. www.w3c.org/TR/xmlenc-decrypt
22. World Wide Web Consortium (W3C). www.w3c.org
23. Blu-ray Disc Association (BDA), Audio Visual Basic Specifications version 0.89 July 2004. www.bluraydisc.com
24. ISO/IEC 13818-2 1996 Information technology, ISO/IEC 13818-2 1996 Information technology—Generic coding of moving pictures and associated audio information—Part 2: Video (MPEG-2 Video)
25. Microsoft Corporation, Jscript Language Reference 5.5, MSDN library.
26. Ola Andersson et al., Scalable Vector Graphics (SVG) 1.1 Specification, World Wide Web Consortium (W3C),Recommendation 14 January 2003. www.w3.org/TR/2003/REC-SVG11-20030114/

27. Steven Pemberton et al., XHTML1.0 The Extensible HyperText Markup Language (Second Edition), World Wide Web Consortium (W3C)Recommendation revised 1 August 2002. www.w3.org/TR/2002/REC-xhtml1-20020801
28. Sharon Adler et al., Extensible Stylesheet Language (XSL) Version 1.0, World Wide Web Consortium (W3C) Recommendation, 15 October 2001.
29. Intel et al., Advanced Access Content System (AACS), Technical Draft, July 14 2004 www.aacsla.com
30. DVD Forum, Enhanced DVD Specification version 0.9, DVD Forum news, Vol 19, October 2003, Office of the secretary, DVD Forum.
31. Sun Microsystems, JavaTM Cryptography Extension 1.2.2, API Specification & Reference. java.sun.com/products/jce/
32. John Boyer, Canonical XML Version 1.0, World Wide Web Consortium (W3C) Recommendation 15 March 2001. http://www.w3.org/TR/2001/REC-xml-c14n-20010315
33. Phillip Hallam-Baker et al., XKMS – XML Key Management Specification, World Wide Web Consortium (W3C) Recommendation 2 May 2005. http://www.w3.org/TR/2005/PR-xkms2-20050502/
34. OMA DRM 2.0, OMA-DRM-DRM-V2_0-20040716-C www.openmobilealliance.org
35. OMA DRM Content Format, OMA-DRM-DCF-v2_0-20040715-C, www.openmobilealliance.org.
36. Nokia, S3-040781 Extensions to OMA DRM V2.0 DCF for MBMS Download Protection,S3#35, Oct 2004, 3GPP.
37. Nokia, Overhead and Performance Comparison of OMA DRM V2.0 DCF and XML for MBMS Download Protection, 3GPP TSG SA WG3 Security S3#36,