Simplified Key Management for Digital Access Control of Information Objects

Kevin E. Foltz and William R Simpson

***Abstract*—Access control of information objects is complicated by the need to establish a common set of access requirements, bind those access requirements to the information object, and compute whether or not the criteria are met for allowing access. An information object can be an e-mail, a Word document, a spreadsheet, or a series of sensor readings. In the simplified case, objects that need to be controlled will be stored in an encrypted file. The file will be decrypted when access criteria are verified. With increasing requirements for records management and maintenance of more and more electronic objects, the number of controlled information objects is rising dramatically. In the past, key management has been extensive, with little efficiency available when encrypting large numbers of information assets. Often, grouping and segmenting objects by type is done to reduce the number of keys needed and hence reduce management of keys. This approach compromises a large number of content files when exploits manage to extract cryptographic keys. Yet maintaining distinct keys for each content object makes key management a serious issue. The proposed process uses a hybrid symmetric/asymmetric keying approach that provides a unique key for each information object while minimizing the key management requirements. This method reduces losses to individual information objects when keys are compromised, but with a greatly reduced key management process that relies on PKI processes.**

***Index Terms* — Access Control, Authorization, Content Protection, Digital Rights Management, Record Management**

I. INTRODUCTION

Content or information assets include documents, spreadsheets, web pages, presentations, and other complete or incomplete sets of information. All information assets are considered authoritative and are under rights management. Rights management is an integral part of the development of these contents. As much as is possible, the workings of the rights management system should be transparent to the user. This is as much for recordkeeping as for control.

Several concepts are reviewed in the next sections that apply to content delivery, before getting into the details of distributing assured content:

a. Digital Rights Management (DRM),
b. Document Ingest and Tagging,
c. Setting up Access, and
d. Key Generation and Management.

II. CONTENT DELIVERY AND DIGITAL RIGHTS MANAGEMENT

DRM technologies attempt to control use of digital media by preventing access, copying, or conversion to other formats by end users. Long before the arrival of digital or even electronic media, copyright holders, content producers, and other financially or artistically interested parties had an interest in controlling access and copying technologies. Examples include: player piano rolls early in the 20th century [1] and video tape recording [2]. The advent of digital media and analog/digital conversion technologies, especially those that are usable on mass-market general-purpose personal computers, has vastly increased the concerns of copyright-dependent individuals and organizations, especially within the music and movie industries, because these individuals and organizations are partly or wholly dependent on the revenue generated from such works.

The advent of personal computers as household appliances has made it convenient for consumers to convert media (which may or may not be copyrighted) originally in a physical/analog form or a broadcast form into a universal, digital form (this process is called ripping) for location- or time-shifting. This, combined with the Internet and popular file-sharing tools, has made unauthorized distribution of copies of copyrighted digital media (digital piracy) much easier. DRM technologies have enabled publishers to enforce access policies that discourage copyright infringements. DRM is most commonly used by the entertainment industry (e.g., film and recording). Many online music stores, such as Apple Inc.'s iTunes Store, as well as many e-book publishers, have implemented DRM [3]. In recent years, a number of television producers have implemented DRM on consumer electronic devices to control access to the freely broadcast content of their shows, in response to the rising popularity of time-shifting digital video recorder systems and other recording devices [4].

Common DRM techniques mentioned in the literature include:

- Embedding of tag(s) – usually encrypted (This technology is designed to control access, distribution and reproduction of accessed information) [5],
- Content encryption [6], and
- Scrambling of expressive material – another word for less formal encryption [7].

Additional DRM background material is presented in [8-16].

Manuscript received 12 February 2016; revised 14 March 2016. This work was supported in part by the U.S. Secretary of the Air Force and The Institute for Defense Analyses (IDA). The publication of this paper does not indicate endorsement by any organization in the Department of Defense or IDA, nor should the contents be construed as reflecting the official position of these organizations

Kevin E. Foltz is with the Institute for Defense Analyses.(email: kfoltz@ida.org)
William R. Simpson is with the Institute for Defense Analyses, 4850 Mark Center Drive, Alexandria, Virginia 22311 USA and is the corresponding author phone: 703-845-6637, FAX: 703-845-6848 (e-mail: rsimpson@ida.org)

Most DRM schemes use encrypted media, which either requires purpose-built hardware or run-time decryption (using hardware protected keys) through software to hear or see the content. This appears to ensure that only authorized users (those with the hardware) can access the content.

Additionally, purpose-built software for the content can enforce restrictions on saving or modifying content, and on dates of applicable use, etc. Purpose-built hardware and software additionally tries to protect a secret decryption key from the users of the system. While this in principle can work, it is extremely difficult to build the hardware to protect the secret key against a sufficiently determined adversary. Many such systems have failed in the field. Once the secret key is known, building a version of the hardware that performs no checks is often relatively straightforward. Additionally, user verification provisions are frequently subject to attack, pirate decryption being among the most frequent. Content management within defense enterprises is of paramount importance for both the protection of assets from wiki-leaks type incidents and records management. Content management in the defense enterprise context is the restriction of access and movement of information within the defense enterprise and the release of the information outside of the defense enterprise. A principle feature of all content management concepts is encryption of the material and decryption when conditions are met, leading to a very large key management problem.

III. ANATOMY OF MANAGED CONTENT

Authorized individuals are to be allowed access to managed content and unauthorized use or authorized misuse is to be prevented. Content includes documents, spreadsheets, web pages, presentations, and other complete or incomplete sets of information. These content items have applications that provide a user presentation (such as Word or Excel). The application must have an appliqué that recognizes managed content and contacts the content manager for resolution of access. Unmanaged content is simply imported into the display application. The user requests access to a piece of content (the user may discover the content location, be provided the content location by an outside source, or may browse to the content location) as shown in Figure 1. Normally the content type will be determined by the name extensions such as .doc or .ppt.

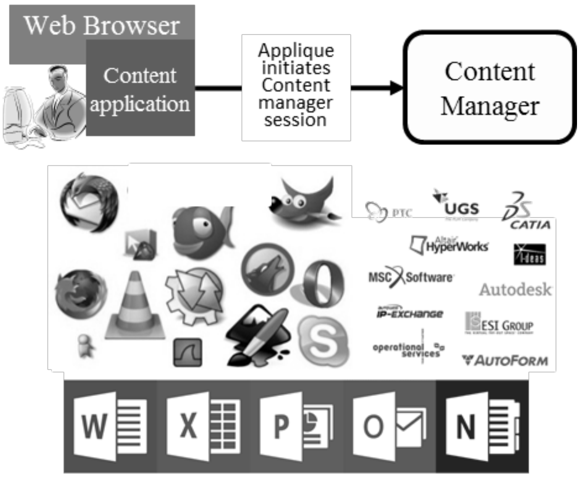

Fig 1 User Access Steps

At this point the content alone does not provide enough structure to achieve this approach.

A. *Electronic Records*

The concept of an electronic record is that each save is a new document. This provides an archive and trace for later analysis. At the time of the save, elements are added to the document so that the above scenario can be accomplished. These elements include (but may not be limited to):

a. Signatures of the individual saving the document,
b. The name and location of the content manager,
c. The access control elements (rules, tags, references),
d. The encrypted content,
e. The location and file names with extensions.

The actual form of this document is shown in Figure 2.

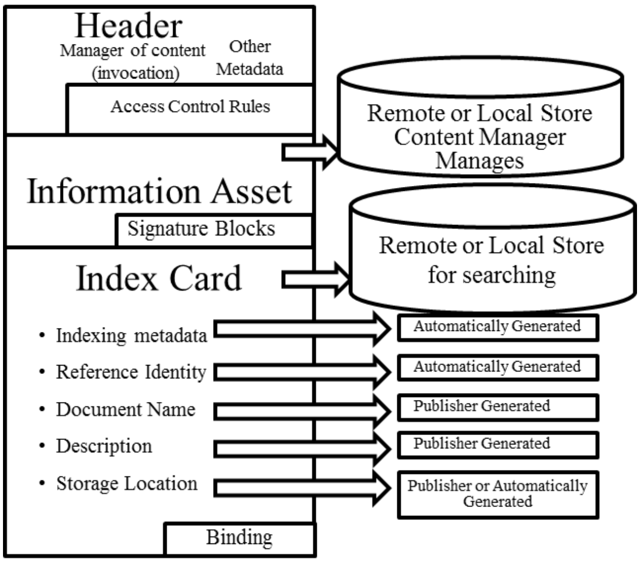

Fig 2 Managed Content Elements

The revised content now has three major elements, which comprise a header, a content block, and an indexing element for searching. That last element may be stored anywhere convenient for searching as long as it is bound to the document. The remaining two elements must be encrypted to avoid misuse. They must also provide an indicator to the appliqué that processing is needed. This can be as simple as a name extension (e.g., .docctrl, or .pptctrl)

IV. Unmanaged Content

Unmanaged content is defined as content without restriction on access, and this content may be saved directly as unencrypted. The applique will save this content in this way for all content that the developer of the content has designated as open access. It is still recommended that the content be signed by the creator for integrity purposes. Indexing data may or may not be generated for these items. The unencrypted data has a normal name extension and is directly processed by the content display application. The storage and usage by content applications is today’s norm. The content manager is not involved in this process. Signature checking and verification is the responsibility of the content application.

V. SETTING UP ACCESS

Often access can be simple tags. For example:

Access = office, age, service date, etc. = tag3.

This could be arbitrarily complex – (for example, 15 attributes having 100 possible values, etc.).

Transmitting access control rules can reduce complexity:

Access = BOOL [(Office = xxx . AND. Rank >= O2) .OR. …] .OR . DN= author}

Transmitting cross-reference activities may also reduce the complexity as shown in the next example:

Access = Possession of an auditor claim to XYZ Financial System.

The rules may contain permissions such as read only, or full editing (cut and paste, etc.). Saving always creates a new record. The rule could have a default based on the author's credentials (modifiable by author). Attributes in a rule must be available to the content manager. A drawback is that each rule must be evaluated at retrieval time (single identity, against a rule or two). There could be many content claims engines. Each content management system must be configured for access to an attribute or a claims store so that the access requirements can be evaluated.

The content display applications must invoke evaluation (word, acrobat, computer-aided design (CAD), etc.) through their appliqué. The content manager is responsible for evaluating access. The most difficult part is management of the encryption keys for many documents and their variations (each save is a new document). Schemas for grouping content elements by class, category, or location and reusing keys within groups create a problem of losses. The compromise of using a single key may subject many documents to unauthorized use.

VI. KEY GENERATION

Key Generation is the responsibility of the content manager. Two keys for each document are needed. The first key is for encryption of the content element. This is the key that will be returned to the appliqué on the user's machine to allow decryption and display of the content element. This key will be returned when the access control requirements are verified as met. The second key is for the access control portion of the header. Storing this information in the clear creates an unintended information leak when nefarious entities are present in the system. Generating the keys is no problem, but protecting, maintaining, and managing them for potentially thousands of documents is problematic. Such problems are normally solved by generating a secure database with cross-references between keys and content being protected. Since recordkeeping requires every save to be a new document, this quickly becomes a numbers and assets game. Standard methods for reducing the number of keys being managed are discussed in the previous section. The next section proposes a novel approach, which may be peculiar to our security approach.

VII. SIMPLIFIED KEY MANAGEMENT

To understand the simplified key management approach, we must first review some basics of our security approach.

A. *Security Processes*

The approach is a distributed end-to-end process called Enterprise Level Security (ELS). It consists of four main security principles:

a. Know the players – This is done by enforcing bi-lateral end-to-end strong authentication. In ELS the identity certificate is Public Key Infrastructure (PKI). Private keys are held in a secure tamper-proof hardware store and are available only to the owner of the certificate. There is no distribution of private keys. PKI certificates are verified and validated. Ownership is verified by a holder-of-key check. Questionable identities are subjected to additional identity measures.

b. Maintain Confidentiality – This entails end-to-end unbroken encryption. In ELS this is end-to-end TLS encryption (It is important to never give away private keys, which belong uniquely to the certificate holder) at the end point.

c. Enforce Access Control – This is done via an authorization credential. In ELS, the certificate is Security Assertion markup Language (SAML). SAMLs are signed, and the signatures are verified and validated. The credentials of the signers are verified and validated.

d. Maintain Integrity – Know that you received exactly what was sent; know that the content has not been tampered with—in ELS this is done in the message authentication codes and electronic signatures, and by holding to end-to-end unbroken communications.

B. *Relying on PKI*

Encryption using PKI is unique in that the public key is available to all, but the private key is available to only the certificate holder. Information encrypted with the public key can be decrypted only with the private key. However, asymmetric encryption is impractical on a large scale.

C. *Combining Symmetric and Asymmetric Encryption*

The approach undertaken uses a combination of the two encryption methods and eliminates the key management issue. The symmetric key for the header is wrapped in the asymmetric public key of the content manager and the header contains the symmetric key for decrypting the content. This arrangement allows only the content manager (or other privileged entities) to use its private key for content decryption. The following steps describe the creation of a content record, which includes the content and header as described below and is shown graphically in Figure 3.

1. The content manager receives the content to be saved from the content application applique at the time the save is initiated by the user in the content application.

The content application is responsible for adding the signature of the content creator for integrity.

2. The content manager may initiate a dialogue with the user to develop the details of the access requirement. If the saved content is an edit of a previous content object, then the default may be provided as a starting point. As indicated earlier, the lack of access requirements will trigger a save of the digitally signed content as unmanaged content without a header and without encryption.

3. The content manger validates the signed content and generates two symmetric keys. We recommend Advanced Encryption Standard (AES 256) for the symmetric encryptions. The first is for content encryption (K_{doc}), the second is for header encryption (K_{hdr}).

4. The content manger encrypts the content with signature using the document encryption key (K_{doc}).

5. The content manger appends the document encryption key (K_{doc}) to the access control rules.

6. The content manager then encrypts this combination in step 3 with the header encryption key (K_{hdr}).

7. The content manager then encrypts the header key (K_{hdr}) with its own public key. We recommend RSA 2048 for asymmetric encryption. Additional copies may be wrapped in the administrator public key or secondary server public key and added to the header for key archive and maintenance. Any of the certificate holders can decrypt the header using their private keys to obtain the symmetric key for the decryption of the content.

8. The content manager then builds out the rest of the header by placing the wrapped key(s) of step 5, the metadata for the header, and the encrypted access rules. At this point the content manager digitally signs the encrypted part of the header for integrity of the header information. This header is added to the encrypted content. Metadata includes the identity of the content manager for use by the content application applique, as well as metadata tags required by that content manager. These content manger tags may be unique to the content manager and the library that it maintains.

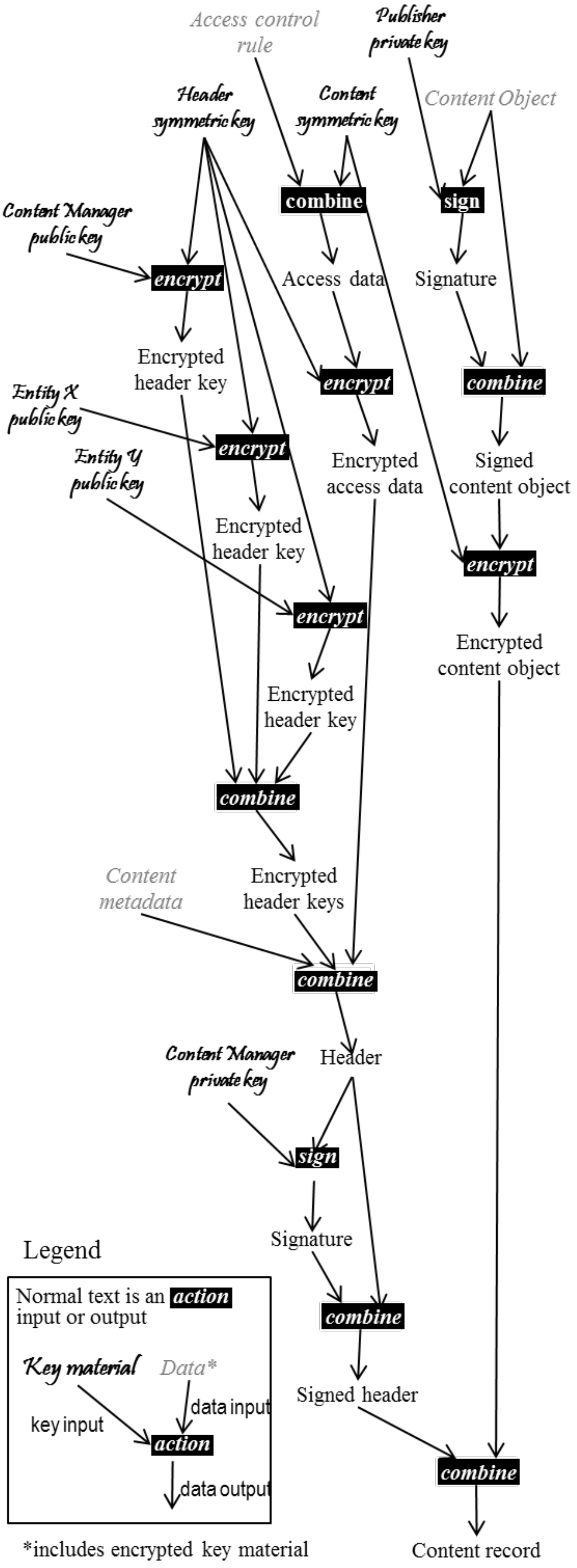

Fig 3 Content Record Creation Process

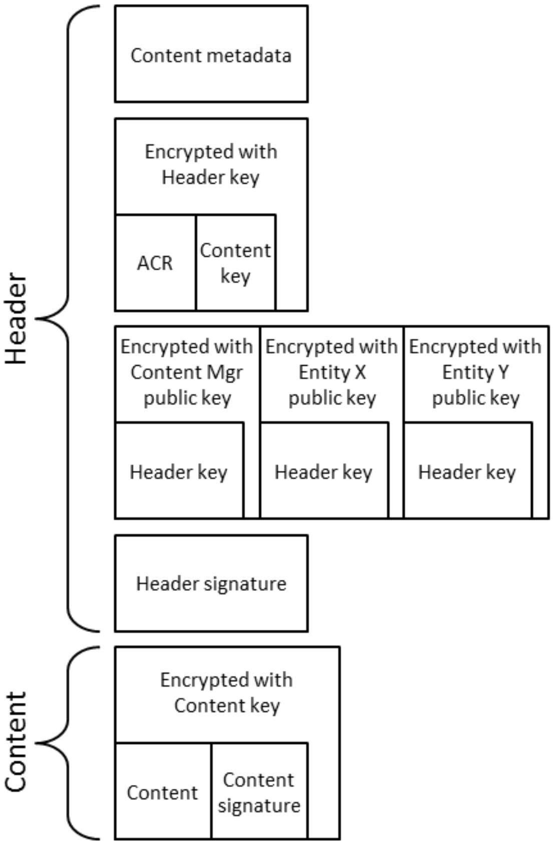
Header
Content metadata
Encrypted with Header key
ACR
Content key
Encrypted with Content Mgr public key
Encrypted with Entity X public key
Encrypted with Entity Y public key
Header key
Header key
Header key
Header signature
Content
Encrypted with Content key
Content
Content signature

Fig 4 Content Record Structure

The process and resulting record is shown in Figures 3 through 5.

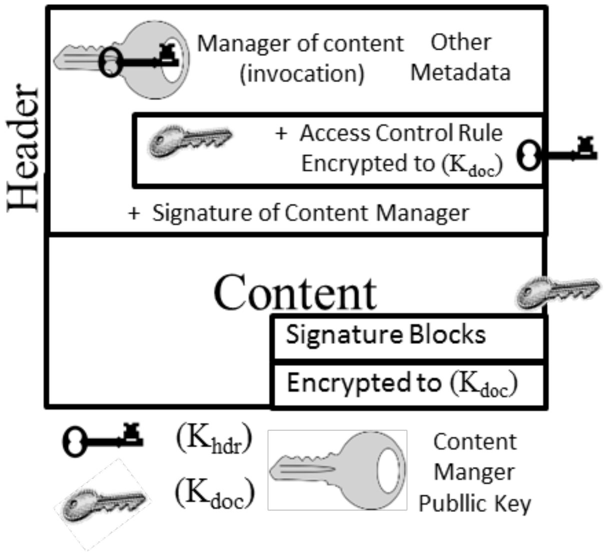
Header
Manager of content (invocation)
Other Metadata
+ Access Control Rule Encrypted to (K_doc)
+ Signature of Content Manager
Content
Signature Blocks
Encrypted to (K_doc)
(K_hdr)
(K_doc)
Content Manger Publlic Key

Fig 5 Proposed Encryption Process

C. *Key Management*

Key management has several elements as described below:

a. **Key Generation –** This is the responsibility of the content manger. The content manager must have certified software for generation of high-entropy encryption keys.
b. **Key Exchange –** There is no key exchange.
c. **Key Use –** This is a responsibility of the content manger. The content manager must have certified software for use of encryption/decryption algorithms. The keys may be changed from time to time or when an event occurs by simply decrypting and then re-encrypting and repackaging.
d. **Key Protection –** This is the responsibility of the content manger. The keys only need to be protected during document preparation and can be destroyed after the document is stored.
e. **Key Storage –** There is no key storage. Each document stores its own keys.
f. **Key Destruction –** This is the responsibility of the content manger. The content manager must have certified software for key destruction.

The process results in each document having a unique symmetric encryption key, limiting losses to one document when an exploit discovers a key. The overall security is heavily dependent on the PKI and the protection of the private key of the server.

VIII. ACCESS OF DOCUMENTS

Enforcing access control is through the discretionary access control process. If the information asset is unencrypted, access will be provided (this means no access restrictions). If the information asset is encrypted, access is restricted and the information asset is provided an extension that takes the request to open the file to the appliqué for enforcing access, as shown in Figure 4.

The request may come from selection in a content store or by execution of a link provide by a colleague or a search from metadata. The initial post to the content manager contains the header of the document. The content manager decrypts the header and sends a request for information to a configured attribute store that contains the users Distinguished Name, and evaluates the returned data against the access requirements. The content decryption is a seven-step process as shown in Figure 4.

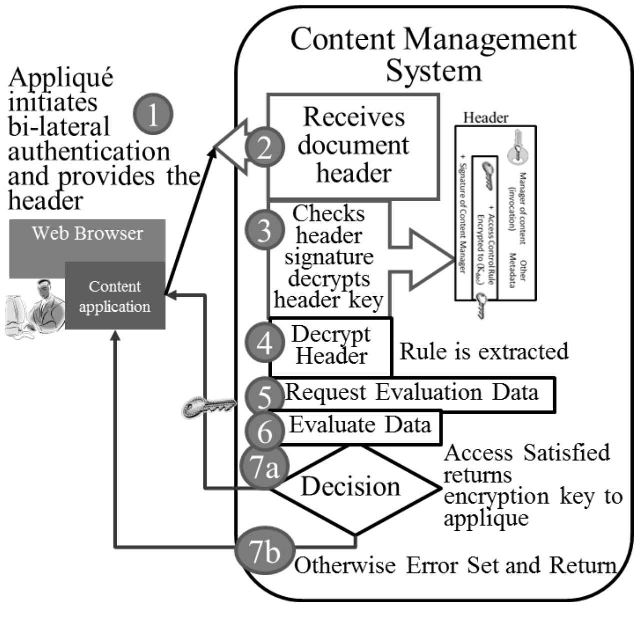
Content Management System
Appliqué initiates bi-lateral authentication and provides the header
1
Web Browser
Content application
2
Receives document header
Header
3
Checks header signature decrypts header key
4
Decrypt Header
Rule is extracted
5
Request Evaluation Data
6
Evaluate Data
7a
Decision
Access Satisfied returns encryption key to applique
7b
Otherwise Error Set and Return

Fig 6 User at a Browser Requesting a Document

ISBN: 978-988-19253-0-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

IX. SUMMARY

We have reviewed the basic approaches to content access control in computing environments. We have also described an approach that relies on high-assurance architectures and the protection elements they provide through PKI. The distribution of private keys is a fundamental violation of a high-assurance model. The high-assurance process, called ELS, allows us to rely on the PKI elements of the system and greatly reduces the key management requirements normally associated with controlling access to content. ELS also permits the unique encryption of each information object, limiting losses to exploits without the growth of key management requirements that normally accompanies such a prolific cryptographic key activity. This work is part of a body of work for high-assurance enterprise computing using web services. Elements of this work are described in [17-30].

REFERENCES

[1] Umeh, Jude, “The World Beyond Digital Rights Management,” British Computer Society, 2007, IBSN 978-I-902505-87-9.

[2] Rimmer, Mathew, “Digital Copyright and the Consumer Revolution”, Edward Elgar Publishing Limited, 2007, IBSN 978 –I -84542-948-5.

[3] Anonymous, “How Apple is changing DRM”, The Guardian, May 2008.

[4] Berlind, David, “TiVo sits at nexus of DRM conundrum”, ZDNet News and Blogs, Setemeber 2006.

[5] Wijering, Jeroen, “W3C Web TV: Adaptive Streaming & Content Protection”, Long Tail Community Blog, Feb 2011.

[6] Kundar, D., and Karthik, K., “Video Fingerprinting and Encryption Principles for Digital Rights Management”, Proceedings of the IEEE, Vol. 92, No. 6, June 2004.

[7] Safavi, R., and Yung, M. (eds), “Digital Rights Management Technologies, Issues, Challenges and Systems”, 1st International Conference, Sydney, Australia, November 2005.

[8] Jean-Marc Boucqueau, “Digital Rights Management,” 3rd IEEE International Workshop on Digital Rights Management Impact on Consumer Communications, January 11 2007.

[9] Oestreicher-Singer, Gal and Arun Sundararajan, “Are Digital Rights Valuable? Theory and Evidence from the eBook Industry,” Proceedings of the International Conference on Information Systems, 2004.

[10] Tom Bramwell, “Ubisoft DRM was ‘attacked’ at weekend,” 2010. http://www.eurogamer.net/articles/ubisoft-drm-was-attacked-at-weekend

[11] Kartik Mudgal, “35 Million Active Gamers on Steam; Valve hints at an Improved Source Engine,” 2011. http://gamingbolt.com/35-million-active-gamers-on-steam-valve-hints-at-an-improved-source-engine

[12] Earnest Cavalli, “Steam Update ‘Makes DRM Obsolete’,” 2009. http://www.wired.com/gamelife/2009/03/steam-update-ma/

[13] Advanced Access Content System (AACS): “Introduction and Common Cryptographic Elements’, Book, 2011. http://www.aacsla.com/specifications/AACS_Spec_Common_Final_0952.pdf

[14] Carey Lening, Copyright Protection of Digital Television: The “Broadcast Flag,” 2005. http://fpc.state.gov/documents/organization/45183.pdf

[15] Dean Takahashi, “With online sales growing, video game market to hit $81B by 2016 (exclusive),” 2011. http://venturebeat.com/2011/09/07/with-online-sales-growing-video-game-market-to-hit-81b-by-2016-exclusive/

[16] Gerard M Stegmaier; Pike and Fischer, Inc.; United States. “The Digital Millennium Copyright Act. 2005 Supplement,” 2005.

[17] William R. Simpson, Coimbatore Chandersekaran and Andrew Trice, “A Persona-Based Framework for Flexible Delegation and Least Privilege,” Electronic Digest of the 2008 System and Software Technology Conference, Las Vegas, Nevada, May 2008.

[18] William R. Simpson, Coimbatore Chandersekaran and Andrew Trice, “Cross-Domain Solutions in an Era of Information Sharing,” The 1st International Multi-Conference on Engineering and Technological Innovation: IMET2008, Volume I, Orlando, FL, June 2008, pp. 313–318.

[19] Coimbatore Chandersekaran and William R. Simpson, “The Case for Bi-lateral End-to-End Strong Authentication,” World Wide Web Consortium (W3C) Workshop on Security Models for Device APIs, 4 pp., London, England, December 2008.

[20] William R. Simpson and Coimbatore Chandersekaran, “Information Sharing and Federation,” The 2nd International Multi-Conf. on Engineering and Technological Innovation: IMETI2009, Volume I, Orlando, FL, July 2009, pp. 300–305.

[21] Coimbatore Chandersekaran and William R. Simpson, “A SAML Framework for Delegation, Attribution and Least Privilege,” The 3rd International Multi-Conf. on Engineering and Technological Innovation: IMETI2010, Volume 2, pp. 303–308, Orlando, FL, July 2010.

[22] William R. Simpson and Coimbatore Chandersekaran, “Use Case Based Access Control,” The 3rd International Multi-Conference on Engineering and Technological Innovation: IMETI2010, Volume 2, pp. 297–302, Orlando, FL, July 2010.

[23] Coimbatore Chandersekaran and William R. Simpson, “A Model for Delegation Based on Authentication and Authorization,” The First International Conference on Computer Science and Information Technology (CCSIT-2011), Springer Verlag Berlin-Heildleberg, Lecture Notes in Computer Science, 20 pp.

[24] William R. Simpson and Coimbatore Chandersekaran, “An Agent Based Monitoring System for Web Services,” The 16th International Command and Control Research and Technology Symposium: CCT2011, Volume II, Orlando, FL, April 2011, pp. 84–89.

[25] William R. Simpson and Coimbatore Chandersekaran, “An Agent-Based Web-Services Monitoring System,” International Journal of Computer Technology and Application (IJCTA), Vol. 2, No. 9, September 2011, pp. 675–685.

[26] William R. Simpson, Coimbatore Chandersekaran and Ryan Wagner, “High Assurance Challenges for Cloud Computing,” Lecture Notes in Engineering and Computer Science: Proceedings World Congress on Engineering and Computer Science 2011, WCECS 2011, San Francisco, USA, 19–21 October 2011, pp. 61–66.

[27] Coimbatore Chandersekaran and William R. Simpson, “Claims-Based Enterprise-Wide Access Control,” Lecture Notes in Engineering and Computer Science: Proceedings World Congress on Engineering 2012, WCE 2012, London, U. K., 4-6 July 2012, pp. 524–529.

[28] William R. Simpson and Coimbatore Chandersekaran, “Assured Content Delivery in the Enterprise,” Lecture Notes in Engineering and Computer Science: Proceedings World Congress on Engineering 2012, WCE 2012, London, U. K., 4–6 July 2012, pp. 555–560.

[29] William R. Simpson and Coimbatore Chandersekaran, “Enterprise High Assurance Scale-up,” Lecture Notes in Engineering and Computer Science: Proceedings World Congress on Engineering and Computer Science 2012, WCECS 2012, San Francisco, USA, 24-26 October 2012, pp. 54–59.

[30] Coimbatore Chandersekaran and William R. Simpson, “A Uniform Claims-Based Access Control for the Enterprise,” International Journal of Scientific Computing, Vol. 6, No. 2, December 2012, ISSN: 0973-578X, pp. 1–23.