Digital Key Management for Access Control of Electronic Records

William R Simpson, *Member IAENG* and Kevin E. Foltz

***Abstract*—Access control of electronic records involves establishing a common set of access requirements, binding those access requirements to the electronic records, and computing whether or not the criteria are met for allowing access. An electronic record can be an e-mail, a document, a spreadsheet, or a series of sensor readings. Records that need to be controlled are stored in an encrypted file, which is decrypted when access criteria are verified. The number of controlled electronic records is rising dramatically. Current methods of key management often group and segment objects by type to reduce the number of keys and the associated management overhead. This approach compromises a large number of content files when exploits manage to extract cryptographic keys. The proposed process uses a hybrid symmetric/asymmetric keying approach that provides a unique key for each electronic record while minimizing the key management requirements. This method reduces losses to individual electronic records when keys are compromised, but with a greatly reduced key management process that leverages PKI processes.**

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

I. INTRODUCTION

Electronic records are content or information assets, that include documents, spreadsheets, web pages, presentations, and other complete or incomplete sets of information. All electronic records generated within an enterprise are considered authoritative and are under rights management by the enterprise. Rights management is an integral part of the development of these records, but the workings of the rights management system should be transparent to the user. This helps the enterprise with recordkeeping and document control. This paper is based in part on a paper published by the WCE 2016 [1].

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.

Manuscript received 29 October 2016; revised 9 November 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)

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 [2] and video tape recording [3]. 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 [4]. 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 [5].

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) [6],
- Content encryption [7], and
- Scrambling of expressive material – another word for less formal encryption [8].

Additional DRM background material is presented in [9-17].

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. Furthermore, user verification provisions are frequently subject to attack, pirate decryption being among the most frequent. Content management within enterprises is of paramount importance for both the protection of assets from wiki-leaks type incidents and records management. Content management in the enterprise context is the restriction of access and movement of information within the enterprise and the release of the information outside of the 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. ENTERPRISE LEVEL SECURITY

A. Security Process Background

This work is part of a body of work for high-assurance enterprise computing using web services. The process has been developed over the last fifteen years and is termed Enterprise Level Security (ELS).

ELS is a capability designed to counter adversarial threats by protecting applications and data with a dynamic claims-based access control (CBAC) solution. ELS helps provide a high assurance environment in which information can be generated, exchanged, processed, and used. It is important to note that the ELS design is based on a set of high level tenets that are the overarching guidance for every decision made, from protocol selection to product configuration and use [18]. From there, a set of enterprise level requirements are formulated that conforms to the tenets and any high level guidance, policies and requirements.

B. Design Principles

The basic tenets, used at the outset of the ELS security model are the following:

0. The ***zeroth*** tenet is that the *malicious entities are present* and can look at network traffic and may attempt to modify that traffic by sending virus software to network assets. Current threat evaluation indicates that attacks are often successful at all levels; discovering these attacks and their consequences is problematic. In many cases attackers may compromise and infiltrate before a vulnerability can be mitigated by software changes (patches).

1. The ***first*** tenet is *simplicity*. Added features come at the cost of greater complexity, less understandability, greater difficulty in administration, higher cost, and/or lower adoption rates that may be unacceptable to the organization.

2. The ***second*** tenet, and closely related to the first, is *extensibility*. Any construct we put in place for an enclave should be extensible to the domain and the enterprise, and ultimately to cross-enterprise and coalition.

3. The ***third*** tenet is *information hiding*. Essentially, information hiding involves only revealing the minimum set of information to the requester and the outside world needed for making effective, authorized use of a capability.

4. The ***fourth*** tenet is *accountability*. In this context, accountability means being able to unambiguously identify and track what active entity in the enterprise performed any particular operation (e.g., accessed a file or IP address, invoked a service). Active entities include people, machines, and software process, all of which are named registered and credentialed. By accountability we mean attribution with supporting evidence.

5. This ***fifth*** tenet is *minimal detail* (to only add detail to the solution to the required level). This combines the principles of simplicity and information hiding, and preserves flexibility of implementation at lower levels.

6. The ***sixth*** is the emphasis on a *service driven* rather than a product-driven solution whenever possible. Services should be separated as stated in the separation of function tenant. This also allows simplification and information hiding.

7. The ***seventh*** tenet is that *lines of authority* should be preserved and information assurance decisions should be made by policy and/or agreement at the appropriate level. An example here is that data owners should implement sharing requirements even when the requirements come from “higher authority.”

8. The ***eighth*** tenet is *need-to-share* as overriding the need-to-know. Often effective health, defense, and finance rely upon and are ineffective without shared information. Shared does not mean released and the differences must be clear. However, judicious use of release authority and delegated access lead to a broader distribution of information. This leads to a more formalized delegation policy both within and outside of the enterprise.

9. The ***ninth*** tenet is *separation of function*. This makes for fewer interfaces, easier updates, maintenance of least privilege, reduced and easier identified vulnerabilities and aids in forensics.

10.The ***tenth*** tenet is *reliability*; security needs to work even if adversaries know how the process works. In setting up a large scale enterprise we need to publish exactly how things work. Personnel, computer operations people and vendors need to know how the system works and this should not create additional vulnerabilities.

11.The ***eleventh*** tenet is to *trust but verify* (and validate). Trust should be given out sparingly and even then trusted outputs need checking. Verification includes checking signature blocks, checking that the credential identities match (binding), checking the time stamps, checking to whom information is sent. Checking information received is identical to information sent, etc. Validation includes checking issuing authority, checking certificate validity, checking identity white lists and black lists.

12.The ***twelfth*** tenet is *minimum attack surface*; the fewer the interfaces and the less the functionality in the interfaces, the smaller the exposure to threats.

13.The ***thirteenth*** tenet is *handle exceptions* and errors. Exception handling involves three basic aspects. The first is logging. The second is alerting and all security related events should be alerted to the Enterprise Support Desk (ESD). The third is notification to the user.

14.The ***fourteenth*** tenet is to *use proven solutions*. A carefully developed program of pilots and proofs of concepts has been pursued before elements were integrated into ELS. It is our intention to follow that process even

when expediency dictates a quicker solution. Immediate implementation should always be accompanied by a roadmap for integration that includes this tenet.

15.The ***fifteenth*** tenet is *do not repeat old mistakes*. From a software point of view, this has many implications. First, never field a software solution with known vulnerabilities and exploits. There are several organizations that track the known vulnerabilities and exploits and an analysis against those indexes should be required of all software. Second, a flaw remediation system is required. After a vulnerability analysis, fixes may be required, after fielding, fixes will be required as new vulnerabilities and exploits are discovered. Third, from an operations standpoint take time to patch and repair, including outputs from the flaw remediation and improvements in Security Technical Implementation Guidelines.

Current paper-laden access control processes for an enterprise operation are plagued with ineffectiveness and inefficiencies. Given that in a number of enterprises tens of thousands of personnel transfer locations and duties annually, delays and security vulnerabilities are introduced daily into their operations. ELS mitigates security risks while eliminating much of the system administration required to manually grant and remove user/group permissions to specific applications/systems. Early calculations show that for government and defense 90-95% of recurring man-hours are saved and up to 3 weeks in delay for access request processing are eliminated by ELS-enabled applications [19]. While perimeter-based architecture assumes that threats are stopped at the front gates, ELS does not accept this precondition and is designed to mitigate many of the primary vulnerability points at the application using a distributed security architecture shown in Figure 1.

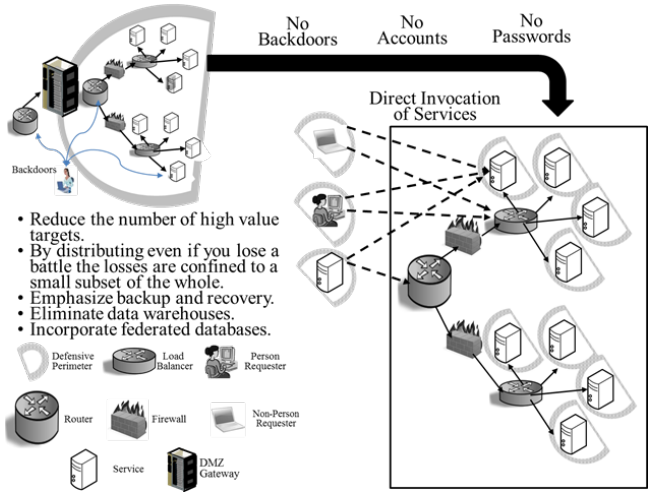


Figure 1 Distributed Security Architecture

C. *Security Principles*

The ELS design addresses five security principles that are derived from the basic tenets:

- Know the Players – this is done by enforcing bi-lateral end-to-end authentication;
- Maintain Confidentiality – this entails end-to-end unbroken encryption (no in-transit decryption/payload inspection);
- Separate Access and Privilege from Identity – this is done by an authorization credential;
- Maintain Integrity – know that you received exactly what was sent;
- Require Explicit Accountability – monitor and log transactions.

a. Know the Players

In ELS, the identity certificate is an X.509 Public Key Infrastructure (PKI) certificate [20]. This identity is required for all active entities, both person and non-person, e.g., services, as shown in Figure 2. PKI certificates are verified and validated. Ownership is verified by a holder-of-key check. Supplemental (in combination with PKI) authentication factors may be required from certain entities, such as identity confirming information or biometric data.

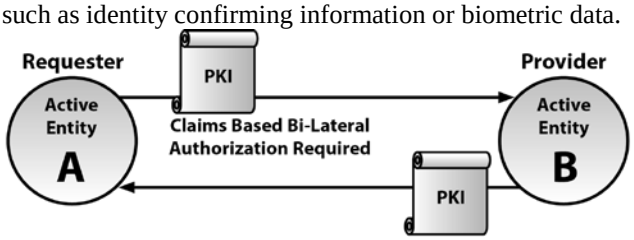


Figure 2 Bi-lateral Authentication

b. Maintain Confidentiality

Figure 3 shows that ELS establishes end-to-end Transport Layer Security (TLS) [21] encryption (and never gives away private keys that belong uniquely to the certificate holder).

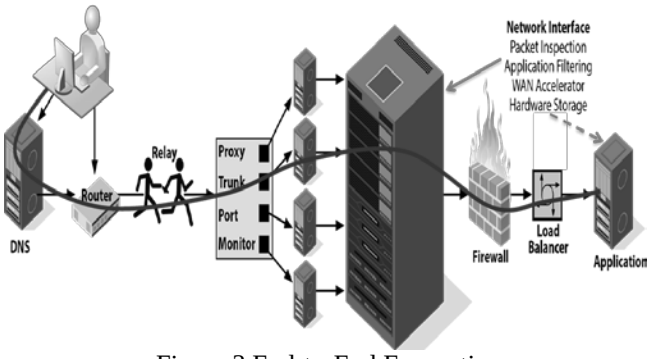


Figure 3 End-to-End Encryption

c. Separate Access and Privilege from Identity

ELS can accommodate changes in location, assignment and other attributes by separating the use of associated attributes from the identity. Whenever changes to attributes occur, claims are recomputed based on new associated attributes (see section III), allowing immediate access to required mission information. As shown in Figure 4, access control credentials utilize the Security Assertion Markup Language (SAML) (SAML authorization tokens differ from the more commonly used single-sign-on (SSO) tokens, and in ELS, are not used for authentication.) [22]. SAML tokens are created and signed by a Security Token Server (STS). The signatures are verified and validated before acceptance. The credentials of the signers also are verified and validated. The credential for access and privilege is bound to the requester by ensuring a match of the identity used in both authentication and authorization credentials.

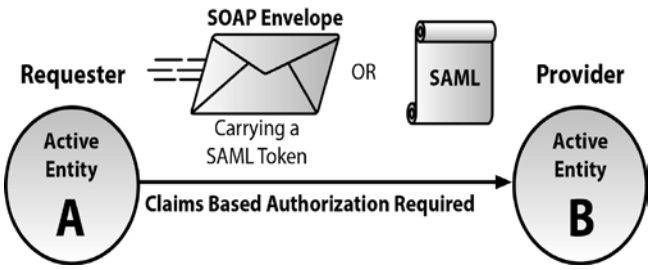


Figure 4 Claims-Based Authorization

d. Maintain Integrity

Integrity is implemented at the connection layer by end-to-end TLS message authentication codes (MACs), see Figure 5. Chained integrity, where trust is passed on transitively from one entity to another, is not used since it is not as strong as employing end-to-end integrity. At the application layer, packages (SAML tokens etc.) are signed, and signatures are verified and validated [23].

e. Require Explicit Accountability

All active entities with ELS are required to act on their own behalf (no proxies, or impersonation allowed). As shown in Figure 6, ELS monitors specified activities for accountability and forensics. The monitor files are formatted in a standard way and stored locally. For enterprise files a monitor sweep agent reads, translates, cleans, and submits to an enterprise relational database for recording log records periodically, or on-demand. Local files are cleaned periodically to reduce overall storage and to provide a centralized repository for help desk, forensics, and other activities. The details of this activity are provided in designated technical profiles and [24, 25].

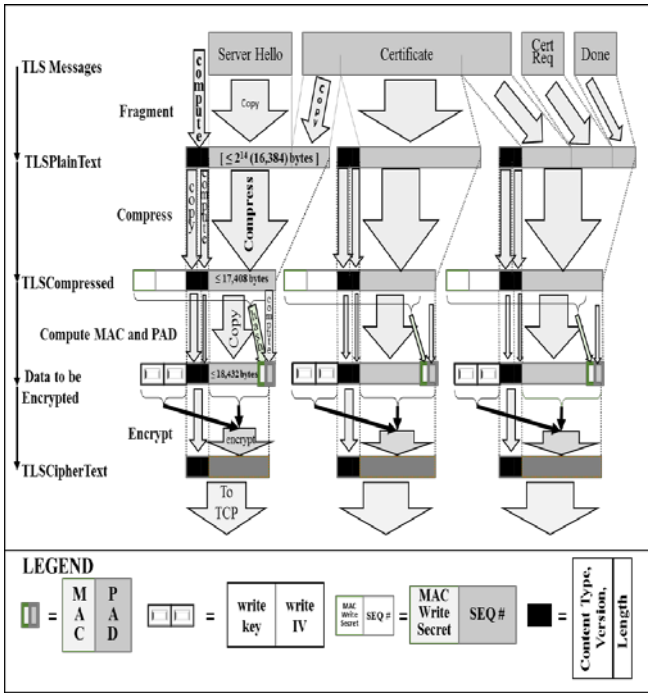


Figure 5 Integrity Measures

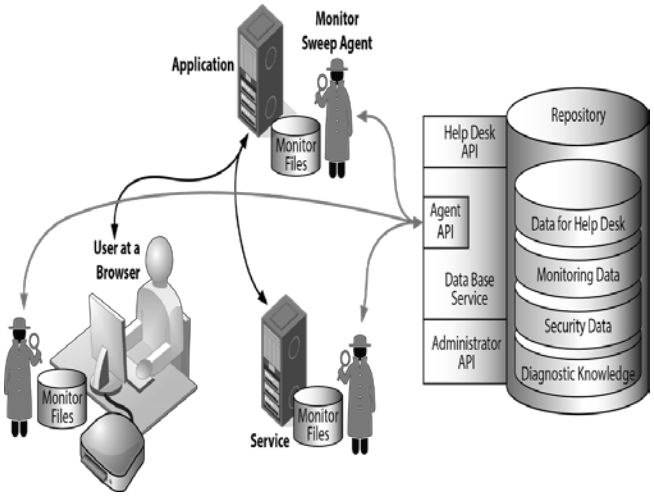


Figure 6 Accountability through Centralized Monitoring

IV. ANATOMY OF MANAGED CONTENT

A. Electronic Record Types

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, receive the content location from an outside source, or browse to the content location) as shown in Figure 7. Normally the content type will be determined by the name extensions such as .doc or .ppt.

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

B. Electronic Record Elements

The concept of an electronic record is that each save creates 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.

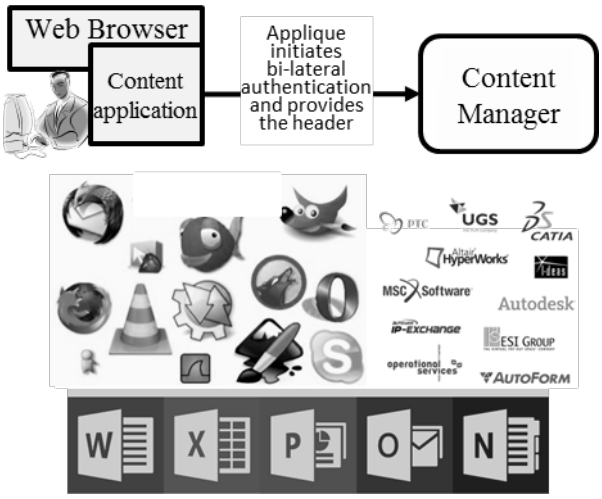


Fig 7 User Access Steps

The actual form of this document is shown in Figure 8.

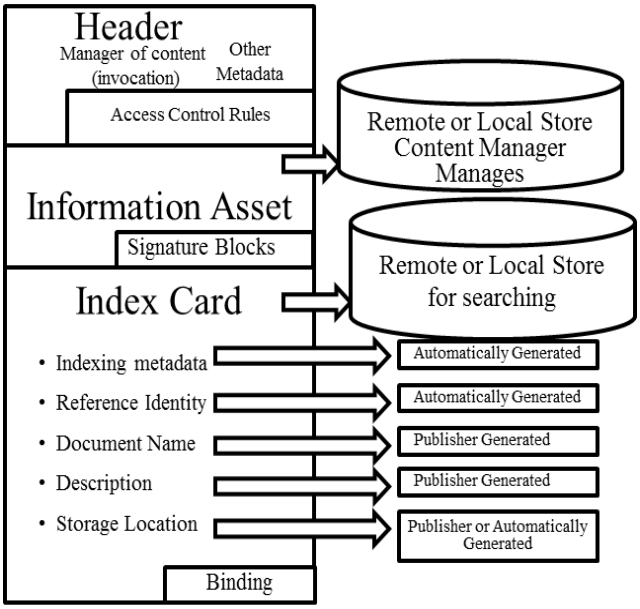


Fig 8 Managed Content Elements

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).

C. Unmanaged Content

Unmanaged content is defined as content without restriction on access, and this content may be saved directly as unencrypted. The appliqué 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

A. Access Rules

Often access can be encapsulated as simple tags that represent the computation of a Boolean function of simple attributes. For example:

$$\textbf{Access} = f_{Bool}\textbf{(office, age, service date, etc.)} = \textbf{\textit{IsPresent}(tag3).}$$

This could be arbitrarily complex (for example, 15 attributes having 100 possible values, etc.), which results in a large number of tags to manage.

Transmitting access control rules can reduce complexity:

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

B. Rule Evaluation

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.

C. 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.

VI. RECORD AND KEY MANAGEMENT

A. Creating a Content Record

Asymmetric encryption is used with PKI credentials. Information encrypted with the public key can be decrypted only with the private key. This form of asymmetric encryption provides important functionality, but it is impractical for encrypting large data sets.

Large scale encryption uses a combination of asymmetric and symmetric encryption 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 9.

1. The content manager receives the content to be saved from the content application appliqué 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 previous values 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 the content symmetric key (K_{doc}), the second is the header symmetric key (K_{hdr}).
4. The content manger encrypts the content with signature using the content encryption key (K_{doc}).
5. The content manger appends the content encryption key (K_{doc}) to the access control rules.
6. The content manager then encrypts this combination with the header symmetric 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 other server public keys and added to the header for key archive and maintenance. Any of these entities can decrypt the header using their private key 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), the metadata for the header, and the encrypted access data. At this point the content manager digitally signs the encrypted part of the header for integrity of the header information. This header is appended to the encrypted content. Metadata includes the identity of the content manager for use by the content application appliqué, 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.

B. 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.

The process and resulting record is shown in Figures 9 through 11. Figure 9 shows the sequence of steps and their dependencies. It processes inputs and keys through concatenation, signatures, and encryptions to arrive at a content record.

Figures 10 and 11 illustrate the resulting record and its structure.

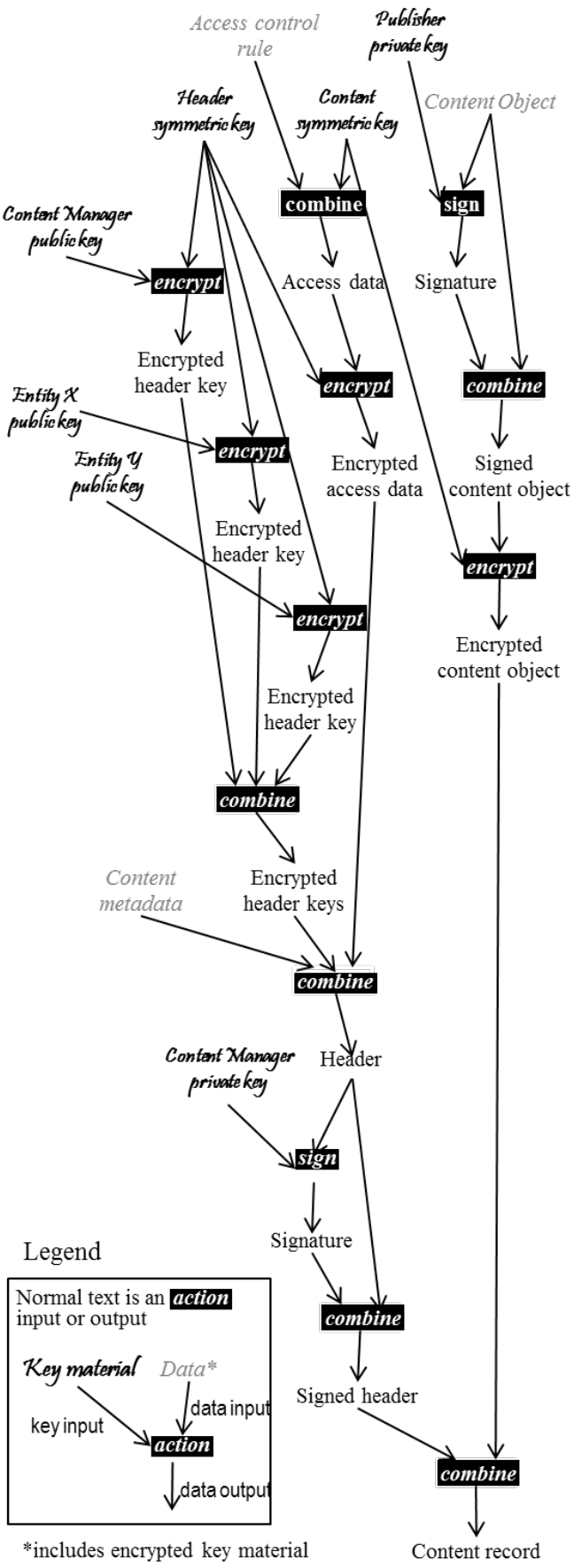


Fig 9 Content Record Creation Process

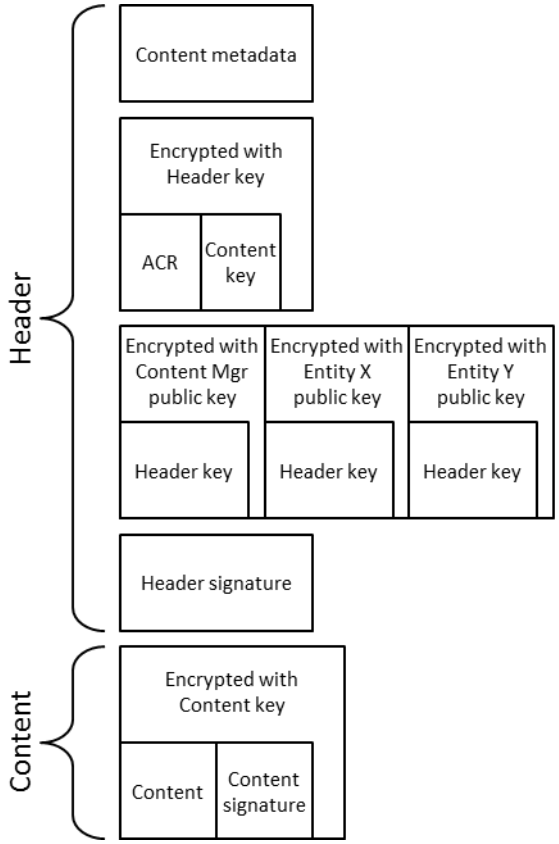


Fig 10 Content Record Structure

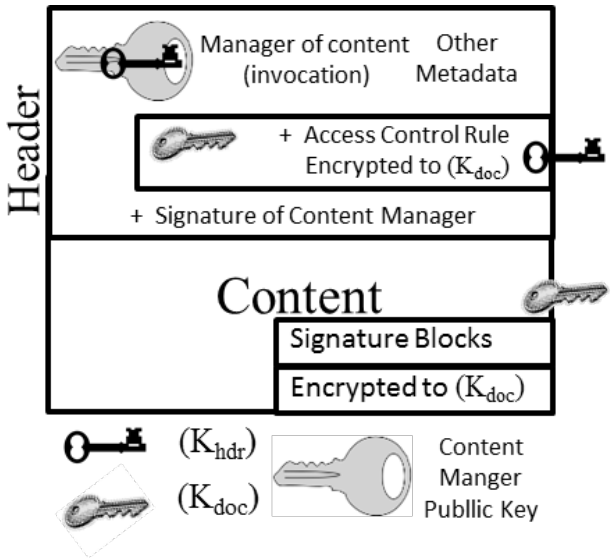


Fig 11 Proposed Encryption Process

C. *Retrieving a Content Record*

Documents are either encrypted or unencrypted. Unencrypted assets are available to all users. They do not contain access controls, and no additional processing is needed to compute whether a user is allowed to access such assets. These are accessed and viewed much like normal files on a standard desktop. No further action by the appliqué or content manager is required for these assets.

For encrypted assets, native applications cannot process the content. They must be modified to use an appliqué that pre-processes the content for the application. The appliqué is invoked for encrypted content directly by the user, or based on the file extension or format. It is responsible for validating the security features of the record, decrypting the

content and providing it to the application for viewing. The appliqué performs the decryption using a decryption key provided by the content manager. The processing of the appliqué and content manager are shown in Figure 12.

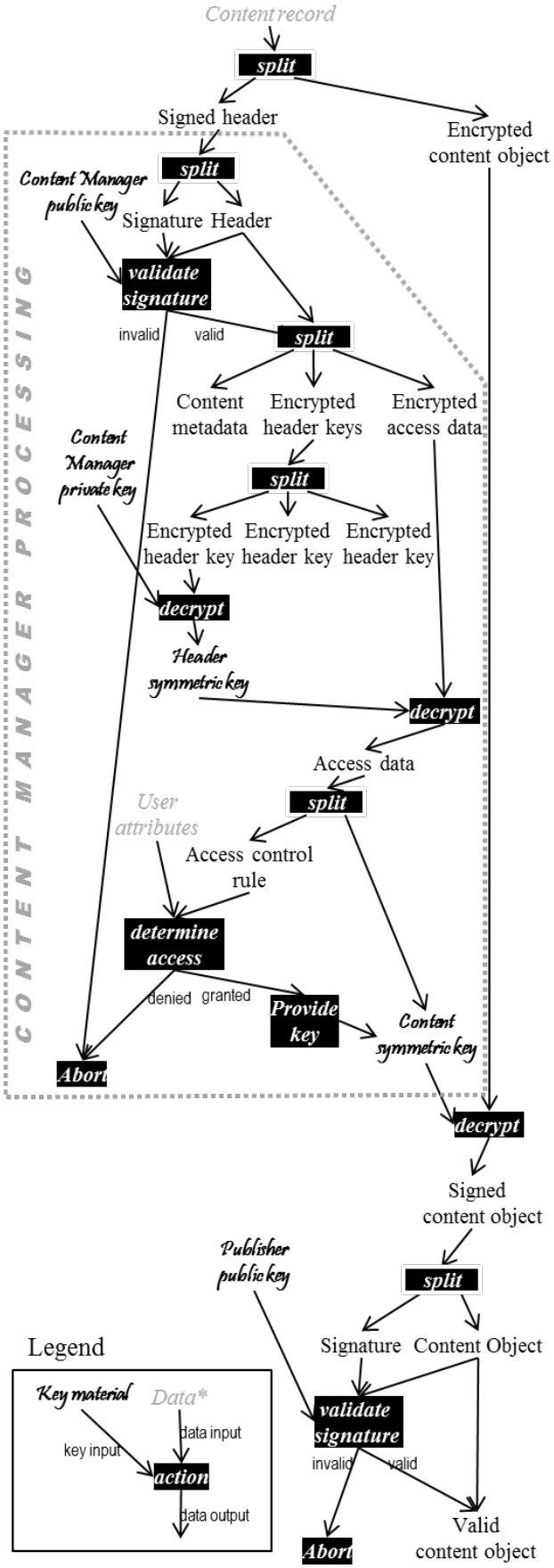


Figure 12 Content Object Extraction Process

The appliqué, upon recognizing a content record, splits the record into the signed header and the encrypted content object. It sends the signed header to the content manager and retains the encrypted content object for decryption using the key that the content manager will return.

The content manager splits the signed header into the header and signature. It uses the public key of the signer to validate the signature and verify that the header was created by a trusted content manager. The content manager public key may be the key of the content manager doing the header processing, or it may be the public key of another content manager in the enterprise that originally created the header signature. The enterprise provides a list of trusted content manager public keys for use by all content managers within the enterprise.

Upon validation, the header is split into content metadata, encrypted header keys, and encrypted access data. The content metadata is used to describe the content, and is useful for searches and other functions, but it is not used by the content manager. The encrypted header keys are split, and the key that is intended for the content manager is decrypted using the content manager's private key. This provides the header symmetric key, and using this key the content manager extracts the access data. This is split into access control rules for the encrypted content object and the content symmetric key, which can be used to decrypt the encrypted content object.

The content manager uses a connection to the enterprise attribute services to determine whether the requester has appropriate attributes to satisfy the access control rule. If the attributes satisfy the rule, access is granted, and the content symmetric key is returned to the appliqué.

Any anomaly within the content manager, such as an invalid signature on the header, a failed decryption of the header key or access data, or insufficient attributes of the requester to meet the access control rule, triggers a standard error procedure. This includes logging the error, providing a standard error response to the appliqué, and discarding and deleting any keys, content, or other material extracted from the record.

When the appliqué receives the content symmetric key, it decrypts the encrypted content object to get the signed content object. This is split into signature and content object, and the signature is validated using the public key of the publisher.

If the signature is valid, the appliqué provides the content to the application for processing. If the signature is invalid, the appliqué aborts the process, logs the error, and returns an error message to the application. In some cases it may be desirable to have the appliqué return the document to the application with a warning when a signature is not valid. However, the content manager should not be allowed to ignore invalid signatures, because such an issue may reflect a larger concern within the enterprise.

The original request by the user for content may come from selection in a content store, execution of a link provided by a colleague, or a search result. The content manager interaction is shown in Figure 13.

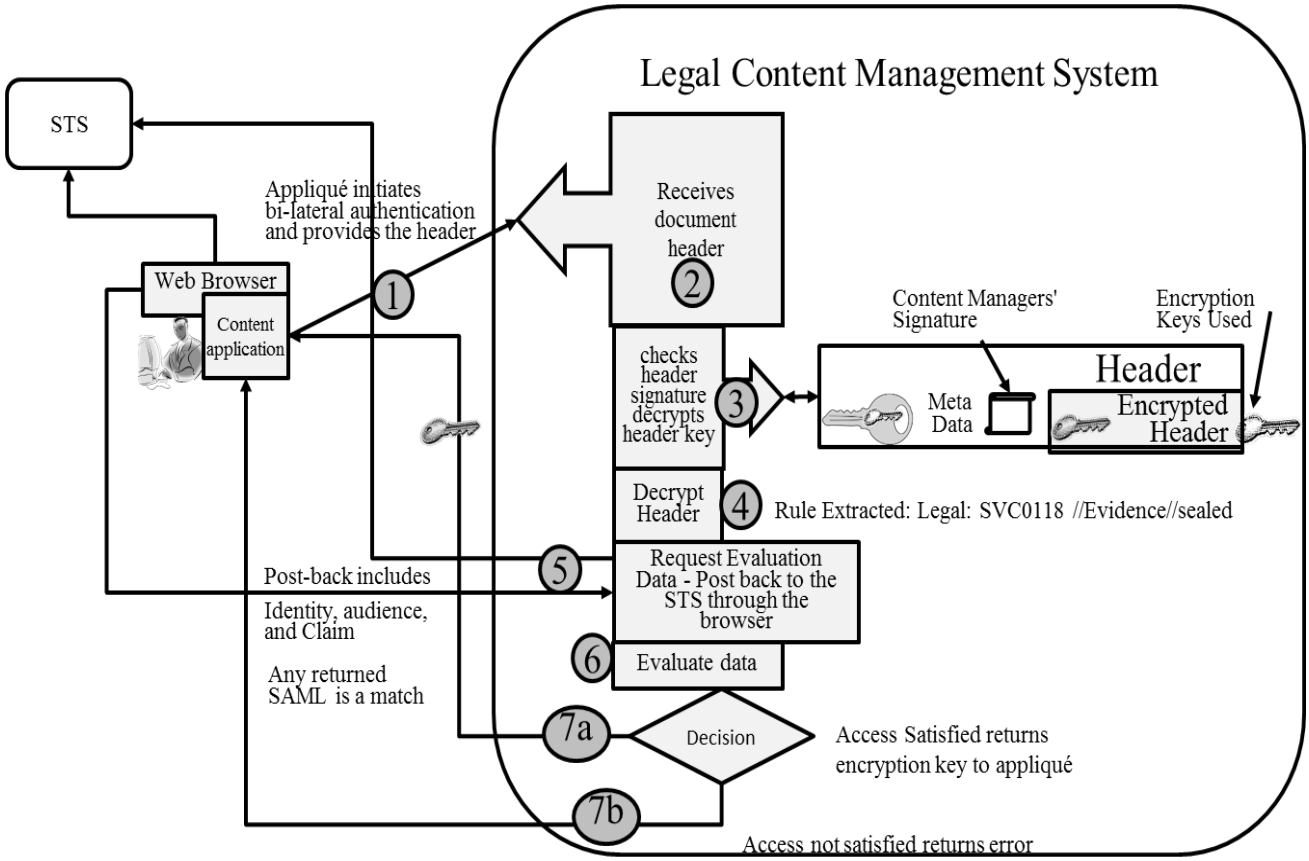


Fig 13 User at a Browser Requesting a Document

1. The appliqué authenticates with the user's credentials to the content management system and provides the signed header to the content management system with a request to decrypt.
2. The content management system authenticates the requester and receives and parses the header.
3. The content management system validates the header signature and decrypts the header key using its private key. The appliqué of the requester should choose a content manager among those for whom an encrypted header key exists in the header.
4. The content management system uses the header key to decrypt the header and parse its contents.
5. The content management system returns an intermediate response to the appliqué that sends it to an STS that provides necessary attributes and access claims. The STS provides the needed claims and attributes and redirects the appliqué back to the content management system. The appliqué sends the necessary claims and attributes to the content management system to enable access according to the access control rule.
6. The content management system evaluates the claims and attributes against the access rules.
7. The content management system makes a decision to provide or deny access, and either returns the content encryption key to the appliqué or returns an error.

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 electronic record, 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 [26-40].

REFERENCES

[1] Kevin Foltz and William R. Simpson, "Simplified Key Management for Digital Access Control of Information Objects", Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2016, WCE 2016, 29 June - 1 July, 2016, London, U.K., pp. 413-418.
[2] Umeh, Jude, "The World Beyond Digital Rights Management," British Computer Society, 2007, IBSN 978-I-902505-87-9.
[3] Rimmer, Mathew, "Digital Copyright and the Consumer Revolution", Edward Elgar Publishing Limited, 2007, IBSN 978 –I -84542-948-5.
[4] Anonymous, "How Apple is changing DRM", The Guardian, May 2008.
[5] Berlind, David, "TiVo sits at nexus of DRM conundrum", ZDNet News and Blogs, Setemeber 2006.
[6] Wijering, Jeroen, "W3C Web TV: Adaptive Streaming & Content Protection", Long Tail Community Blog, Feb 2011.

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

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

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

[10] 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.

[11] Tom Bramwell, "Ubisoft DRM was 'attacked' at weekend," 2010. http://www.eurogamer.net/articles/ubisoft-drm-was-attacked-at-weekend

[12] 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

[13] Earnest Cavalli, "Steam Update 'Makes DRM Obsolete'," 2009. http://www.wired.com/gamelife/2009/03/steam-update-ma/

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

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

[16] 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/

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

[18] William R. Simpson and Kevin Foltz, Proceedings of The 20th World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI, *"Enterprise Level Security - Basic Security Model"*, Volume I, WMSCI 2016, Orlando, Florida, 8-11 March 2016, pp. 56-61.

[19] Email from Rudy Rihani, Project Manager, Accenture Corporation, dated March 6, 2016, Subject: " manpower savings with ELS"

[20] X.509 Standards

a) DoDI 8520.2, Public Key Infrastructure (PKI) and Public Key (PK) Enabling, 24 May 2011

b) JTF-GNO CTO 06-02, Tasks for Phase I of PKI Implementation, 17 January 2006

c) X.509 Certificate Policy for the United States Department of Defense, Version 9.0, 9 February 2005

d) FPKI-Prof Federal PKI X.509 Certificate and CRL Extensions Profile, Version 6, 12 October 2005

e) RFC Internet X.509 Public Key Infrastructure: Certification Path Building, 2005

f) Public Key Cryptography Standard, PKCS #1 v2.2: RSA Cryptography Standard, RSA Laboratories, Oct 27, 2012

g) PKCS#12 format PKCS #12 v1.0: Personal Information Exchange Syntax Standard, RSA Laboratories, June 1999; http://www.rsa.com/rsalabs/node.asp?id=2138 PKCS 12 Technical Corrigendum 1, RSA laboratories, Feb 2000

[21] TLS family Internet Engineering Task Force (IETF) Standards

a) RFC 2830 Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security, 2000-05

b) RFC 3749 Transport Layer Security Protocol Compression Methods, 2004-05

c) RFC 4279 Pre-Shared Key Ciphersuites for Transport Layer Security (TLS), 2005-12

d) RFC 5246 The Transport Layer Security (TLS) Protocol Version 1.2, 2008-08

e) RFC 5289 TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM), 2008-08

f) RFC 5929 Channel Bindings for TLS, 2010-07

g) RFC6358 Additional Master Secret Inputs TLS, 2012-01

h) RFC 7251 AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS, 2014-06

i) RFC 7301 Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension, 2014-07

j) RFC 7457 Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS), 2015-02

[22] Organization for the Advancement of Structured Information Standards (OASIS) open set of Standards

a) N. Ragouzis et al., Security Assertion Markup Language (SAML) V2.0 Technical Overview, OASIS Committee Draft, March 2008

b) P. Mishra et al. Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, March 2005.

c) S. Cantor et al. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, March 2005

[23] William List and Rob Melville, IFIP Working Group 11.5, Integrity In Information, Computers and Security, Volume 13, Issue 4, pp. 295–301, Elsevier, doi:10.1016/0167-4048(94)90018-3, 1994.

[24] William R. Simpson and Coimbatore Chandersekaran, CCCT2010, pp. 84–89, "An Agent Based Monitoring System for Web Services," Orlando, FL, Apr 2011.

[25] William R. Simpson and Coimbatore Chandersekaran, 1st International Conference on Design, User Experience, and Usability, part of the 14th International Conference on Human-Computer Interaction (HCII 2011), "A Multi-Tiered Approach to Enterprise Support Services," 10 pp. Orlando, FL, July 2011. Also published in: A. Marcus (Ed.): Design, User Experience, and Usability, Pt I, HCII 2011, LNCS 6769, pp. 388–397, © Springer-Verlag Berlin Heidelberg 2011.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] Simpson, William R., CRC Press, "Enterprise Level Security – Securing Information Systems in an Uncertain World", by Auerbach Publications, ISBN 9781498764452, May 2016, 397 pp.

APPENDIX:

Legal Content Access Controls Walk Through

A. Introduction

The example in this appendix of legal records generation and control is an adaptation of the general approach to access control of electronic records. The header may contain access control by tags, rules, or a combination of these. This example may contain elements in the access control section that are repeated in the metadata.

This walk through assumes legal cases in a judicial system comprised of pro bono defense attorneys and assigned prosecutors in a judicial pool where records are separately stored by each but the data system is available to all with restricted access and privilege. The legal system is fairly complex in that cases are maintained at the defense and prosecution levels separately and they may eventually become a judicial case. The relationship between the case assignments are shown in Table 1.

The separation is entirely logical in that data may be mixed in the repository. Cases may start as any of the three. A potential defendant may contact an attorney about an incident and the defense case will be opened and assigned a defense case number (Dxxx). A plaintiff may contact a Victim's Counsel about an incident and a prosecution case may be opened as (SVCyyy). An attorney of Record may be assigned if a legal issue is indicated. An incident may trigger a prosecution assignment where the attorney of record is assigned and he will open a case (SVCaaa) with or without a plaintiff to start. The attorney of record on an SVC case may petition the court to open a case where a case may be assigned as (AMJAMSzzz) where AMJAMS is Automated Mandatory Justice Analysis & Management System. The court clerk will assign a judge, and assign or record an attorney of record for the defense, and an attorney of record for the prosecution (the latter attorneys may already exist). If the defense case and prosecution case did not already exist, the will be opened by the attorneys of record. No matter how the case starts or what components it has, strict access controls must be maintained including attorney-client privilege even when outside consultants are used. Each case file may have a number of information categories as follows:

- Correspondence
- Notes
- Evidence (with or without sealed or unsealed tags)
- Filed Motions (with or without sealed or unsealed tags)
- Paperwork (any other document)

Defense (DEF) and prosecution (SVC) material are sealed because of attorney-client privilege. The access rules may be complex, for example, in accordance with ELS tenets, these rules for access are determined by the data owner. In this case, the chief justice of the court with jurisdiction or his designee is the data owner.

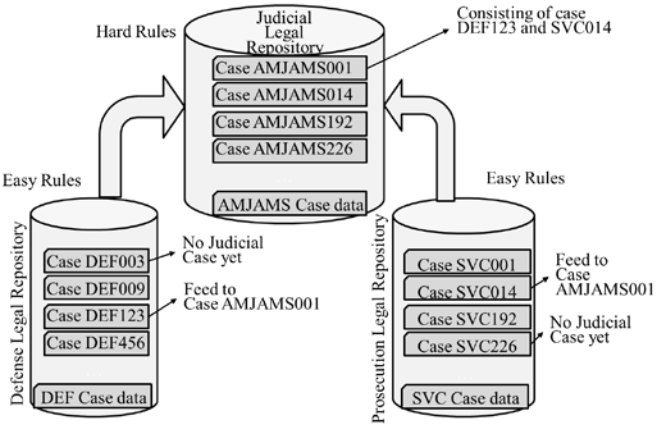


Figure 14 Case Relationships

B. Document Ingest and Tagging of Legal Documents

In accordance with the basic electronic records concept, each laptop or workstation that will access the legal content will be provided content display software that has an appliqué installed for processing controlled documentation.

Table 1 Example of Legal permissions

Judicial cases	Prosecution cases	Defense Cases
For Correspondence (To-From) IF{ID=Any Attorney OR ID = [Client OR Paralegal] } Then access granted return decryption key. ELSE return Null For Notes IF{ID = Any Attorney} Then access granted return decryption key. ELSE return Null For Evidence -Prosecution IF{ID=[Any Attorney OR Paralegal OR Accused]) Then access granted return decryption key. ELSE return Null For Evidence – Defense IF {ID=[Def Attorney OR Def Paralegal OR Def Consulting Attorney OR Accused]} Then access granted return decryption key. ELSE return Null For Paperwork * IF{ID=[Any Attorney OR Paralegal[} Then access granted return decryption key. ELSE return Null For Filed Motions That are Sealed: IF{[ID=Pro/Def/SVC Attorney OR Pro/Def/SVC Paralegal OR Judge]} Then access granted return decryption key. ELSE return Null That are Unsealed: Public (no control) access granted return decryption key. For Judicial Decision No control access granted return decryption key.	For Correspondence (To-From) IF{ID= Pros Attorney OR ID = [Pros Paralegal] } Then access granted return decryption key. ELSE return Null For Notes IF{ID = Pros Attorney} Then access granted return decryption key. ELSE return Null For Paperwork * IF{ID=[Pros Attorney OR Pros Paralegal[} Then access granted return decryption key. ELSE return Null	For Correspondence (To-From) IF{ID= Defense Attorney OR ID = [Client OR Defense Paralegal] } Then access granted return decryption key. ELSE return Null For Notes IF{ID = Defense Attorney} Then access granted return decryption key. ELSE return Null For Other– Defense IF {ID=[Def Attorney OR Def Paralegal OR Def Consulting Attorney OR Accused]} Then access granted return decryption key. ELSE return Null For Paperwork * IF{ID=[DefAttorney OR DEF Paralegal[} Then access granted return decryption key. ELSE return Null.

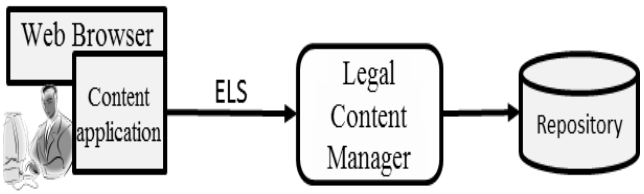


Figure 15 Ingest and Creation of a Controlled Document

A document may arrive by email or other electronic means, or it may be created by one of the principals. The first step is to open or create the document in the appropriate application and then save the document, which triggers the appliqué (the appliqué will ask for the appropriate content manager) and provides a connection to (in this case) the Legal Content Manager. This connection is via normal Enterprise Level security (ELS) including bi-lateral PKI authentication and a Security Assertion Markup Language (SAML) token with the claim of user for the Legal Content Manager and claims for content access (discussed later). The access will be governed by three tags that are created at the time of a save (any changes and saving of a document results in a new document, and documents must be named appropriately by a configuration control process). The three tags are:

1. 'Legal:' This tag is needed because there may be other domains for document controls (e.g., 'Logistics:', 'Specifications:', 'Manuals:' etc.)
2. Case Number: Must be one of DEF, SVC, or AMJAMS followed by a numerical assignment. Examples are: DEF0045, SVC0232, AMJAMS 3214.
3. Information Category: For Legal, this must be one of the categories assigned: Correspondence, Notes, Evidence, Filed Motions , Paperwork.
4. Note -Sealed or unsealed. Required for knowing general availability, but in DEF and SVC cases these are sealed. In AMJAMS cases these are at the discretion of the Judicial Authority for the case. Legal documents are either sealed or unsealed. Unsealed are generally available and will be stored unencrypted.

The process of embedding the tags is provided in Figure 16. The form of the document as stored is shown in Figure 17.

The save dialogue process with the appliqué is shown in Figure 18.

C. Setting Up Access to Legal Documents

A number of steps within the enterprise are necessary to make the access restrictions enforceable. These are listed below and will be covered individually:

a. Authoritative Content Store for cases and personnel assignments
   - Data stored in Enterprise Attribute Store (EAS).

b. Ability to add content access control rules to registry
   - Data Owner is legal system representative.

c. Process Rules and create legal content claims

d. Delegation process for personnel out side of the system

a) Authoritative Content Store

The enterprise must establish an authoritative content store for legal assignments. This store may come from the judicial system or other sources but it must meet the requirements of an authoritative content store. These include authority, assignment, currency, maintenance, etc. The normal ingest of attribute data applies as shown in Figure 19.

Note in the figure the complication of assignments of one individual to another that the claims engine needs to work out. The aggregation and Mediation service must work out conflicts, such as an individual being assigned to both prosecution and defense.

b) Content Access Control Rules Registration

The Enterprise Attribute Ecosystem (EAE) already has facilities for the data owner to create and store in the registry access control rules as shown in the next figure.

These are the rules that will be used by the claims engine in creating content claims for individuals. The auto registration service will be required to accommodate a number of domains including legal. Access to auto-registration is normally the department head, but this access and privilege may be delegated through the normal delegation process. The establishment and complexity of these rules are discussed earlier and a more complete set of rules is shown in Figure 20.

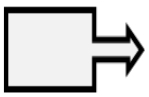
c) Execution of Content Access Rule and Individual Assignments in the Attribute Store

Changes to Attributes and assignments of entities in the enterprise attribute store are processed periodically or on demand. Nominally this would be during non-peak hours after changes have been ingested into the enterprise attribute store. That process is shown in Figure 20.

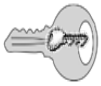
Rule processing is shown in Figure 21. Note that the Processing rules create two classes of claims; the first is the access to the Legal Content Manager, as a user. This is necessary for the saving and retrieval of documents. As a reminder, any modification and subsequent save creates a new document. In the metadata for the content claim, the Claim Name is set equal to the Claim. This requirement will become apparent later. The second set of claims is for the documents by case and category. Note that the form of this claim is identical to the form of the access control header which will be used to advantage on retrieval.

The claims engine must work out the rules of assignment to an individual, which may require iteration. In the introduction, we noted that a paralegal may be assigned to another individual and the rules will state that he can access some of the content that his supervisor has access to.

1. Input from author save function to Legal Content Manager:
   a. Information Asset
   b. Header with domain, access controls tags, meta data
   c. Entire Content Signed by Content Produce.
   d. Unsealed data are stored in unencrypted form.



Legal: DEF0231 //Correspondence
OR
Legal: AMJAMS0112//Motion
…

2. Publisher must have access to created header (Legal Content Manager must check).
3. Publisher Signature is check for integrity.
4. Domain Content Manager generates document key). 
5. Document Content Manager encrypts the information asset using the document key and signs the information asset for integrity.
6. Domain Content Manager generates Golden Key  for this document.
7. Document key is added to header Golden Key is used to encrypt header.
8. Golden Key is wrapped in public key of the domain content manager. 
9. Wrapped Golden key added to the header (restricting use)
10. Additional copies may be wrapped in administrator public key or secondary server public key for archive and maintenance. These can be added to header as necessary.

External Distribution Administrator Copy Archive Copy

11. Any of the certificate holders can decrypt the header.
12. Golden key may be changed on any document (by decrypt and re-encrypt of header – hash does not change).
13. No other key maintenance required
    1. Golden key encrypts all headers for documents maintained by the content manager.
    2. Document keys are generated as needed
    3. Loss of a document key affects only that document
14. Encrypted header and information asset are stored in domain repository.

Figure 16 Access Control Header Generation

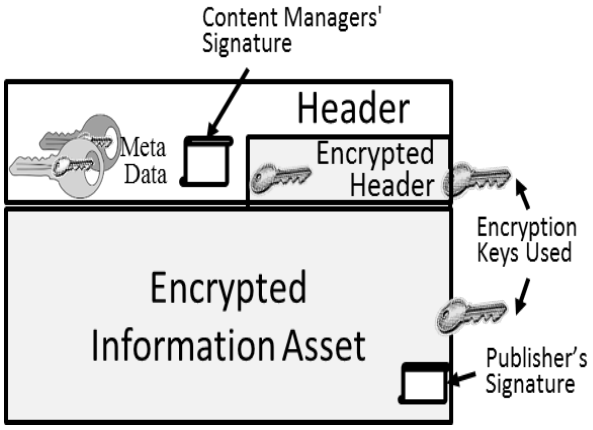


Figure 17 Document Form

d) Delegation Process

In the introduction, special consultants were described as individuals outside of the enterprise (they may or may not be attorneys, but must keep attorney client privilege). These individuals may be granted access by using a special delegation process. This requires several steps that include:

1.The issuance of an identity credential that will be recognized by the legal content manager (normally by adding the certificate authority to the trust store),

2.Issuance of software with the appliqué or provision of access to computer equipment that has this software installed,

3.The creation of a delegated or extended claim as shown in Figure 22. The delegator provides claims to allow delegation through a SAML issued by a local STS.

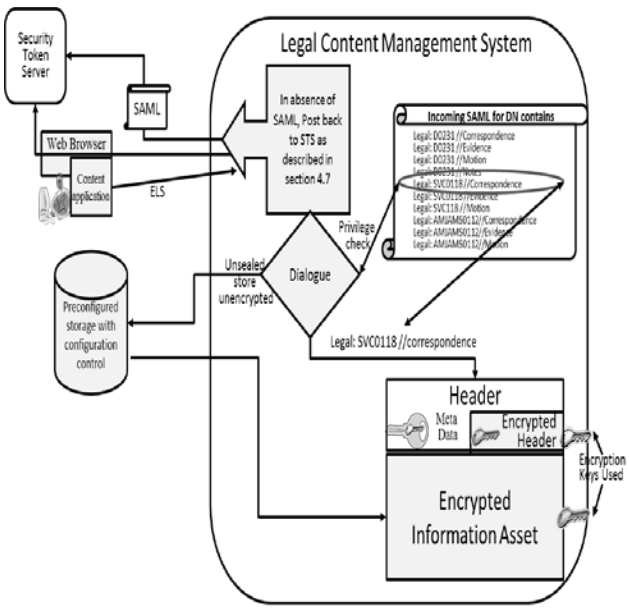


Figure 18 Saving a Document

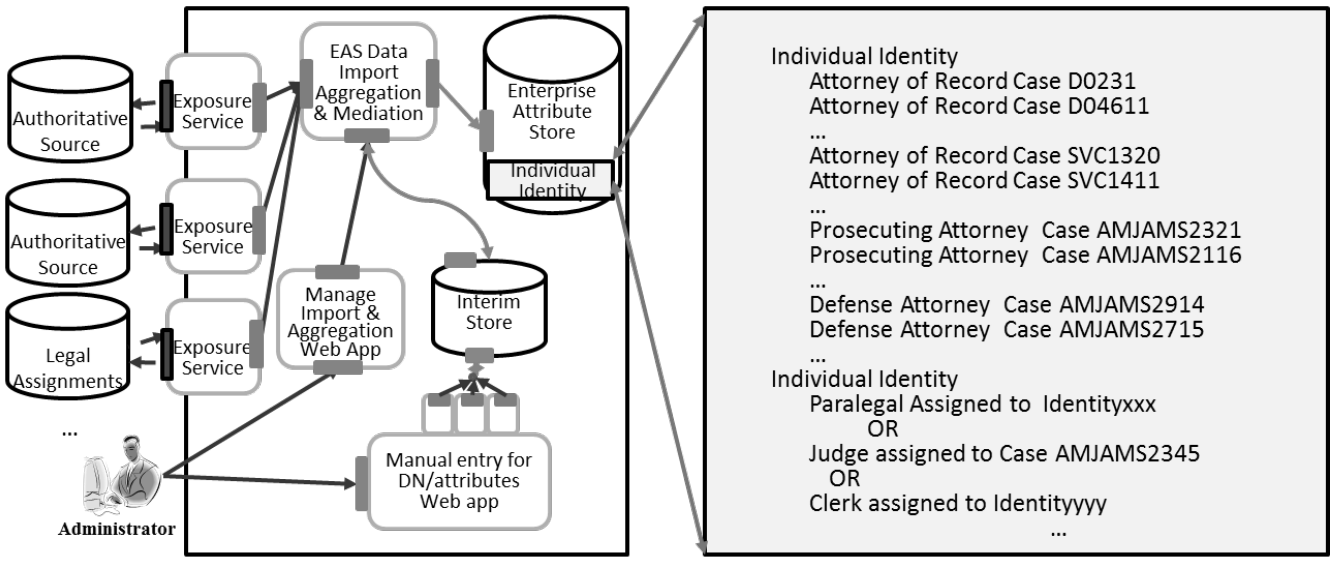


Figure 19 Authoritative Content Store for Legal assignments

D. Retrieving Legal Content

Now that each party has been provisioned, the pieces are in place for content retrieval. 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 which for legal corresponds to unsealed). If the information asset is encrypted this means 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 23.

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 Legal Content Manager contains the link to the document but not the requester's SAML. The content manager, upon recognizing the link retrieves the document, decrypts the header and sends a request for a SAML back though the browser that contains the user's identity, the content manager's name, and the specific claim (which for content is equal to the claim name). The request will either return that claim or set an error flag which then returns a message to the user. The content decryption is a seven step process as shown in Figure 24.

In the figure, the following steps are followed:

1. The appliqué initiates bi-lateral authentication and send header to the legal content management system.
2. The content manager receives the header and begins processing.
3. The content manager initially checks the header signature and verifies the integrity.
4. The content manager decrypts the wrapped header key which was wrapped in the content manager's public key.
5. The content manager decrypts the header and extracts the document key and the access control tokens.
6. The content manager posts back through the browser a request to the local STS that contains the content manager's identity and the claim name (equal to the access control string).
7. If the EAE generates any SAML it has found a match, so the document encryption key is provided to the appliqué, which decrypts the document for the requester.

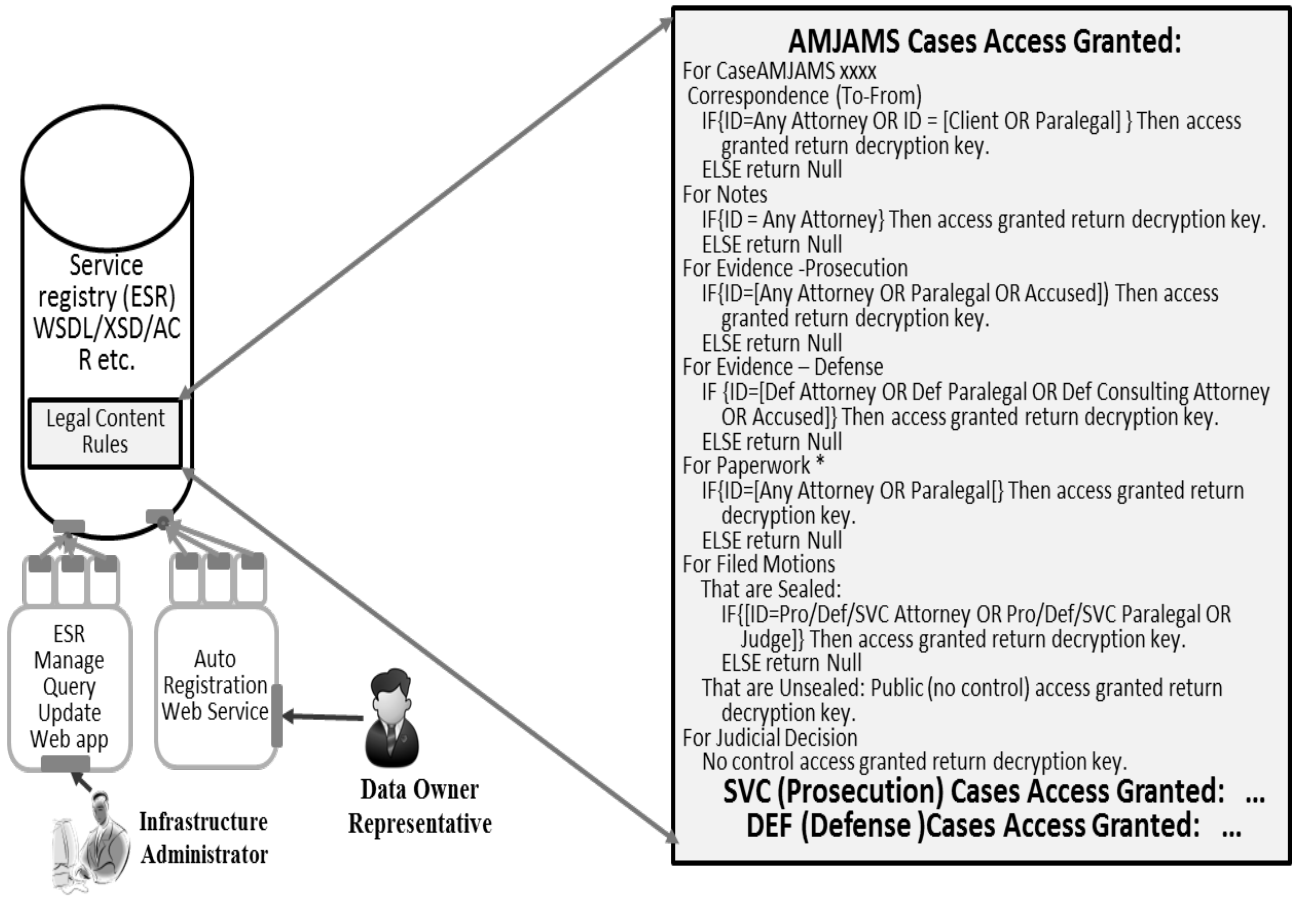


Figure 20 Data Owner Registration of Document Access Control Rules

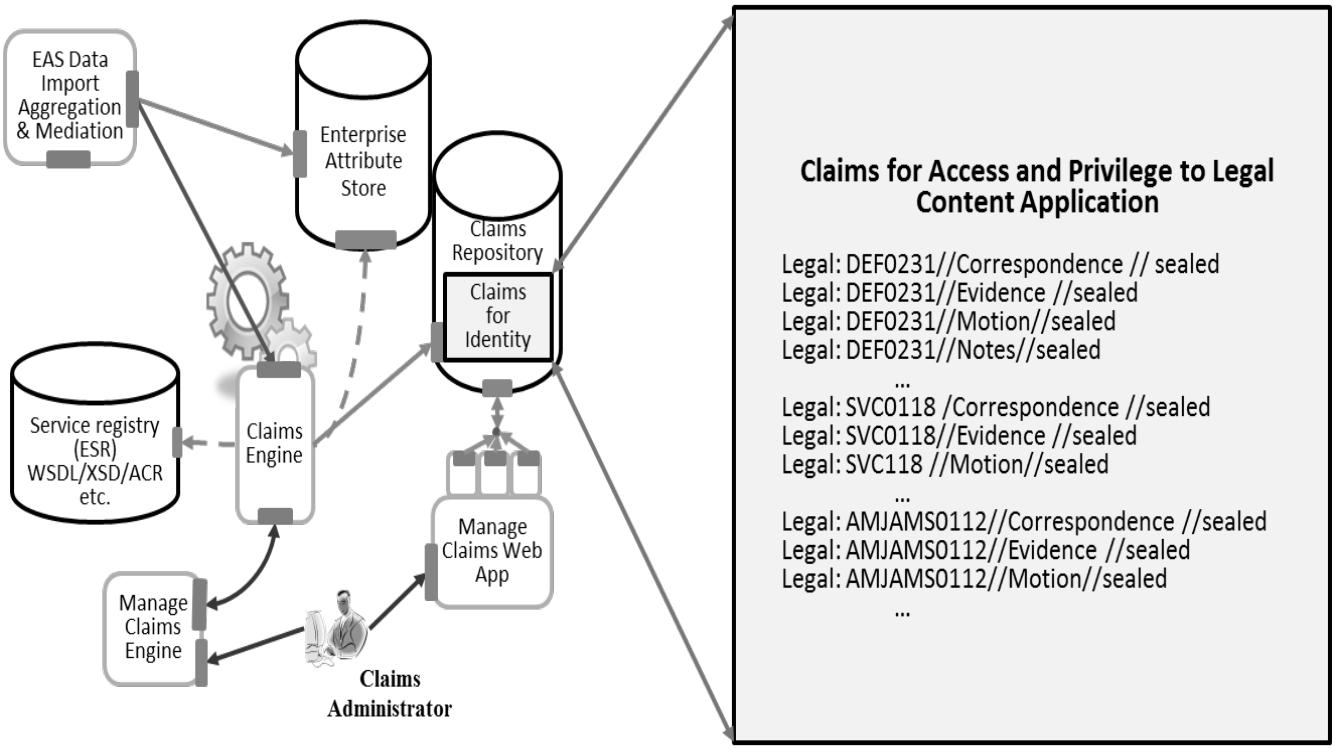


Figure 21 Processing of Claims

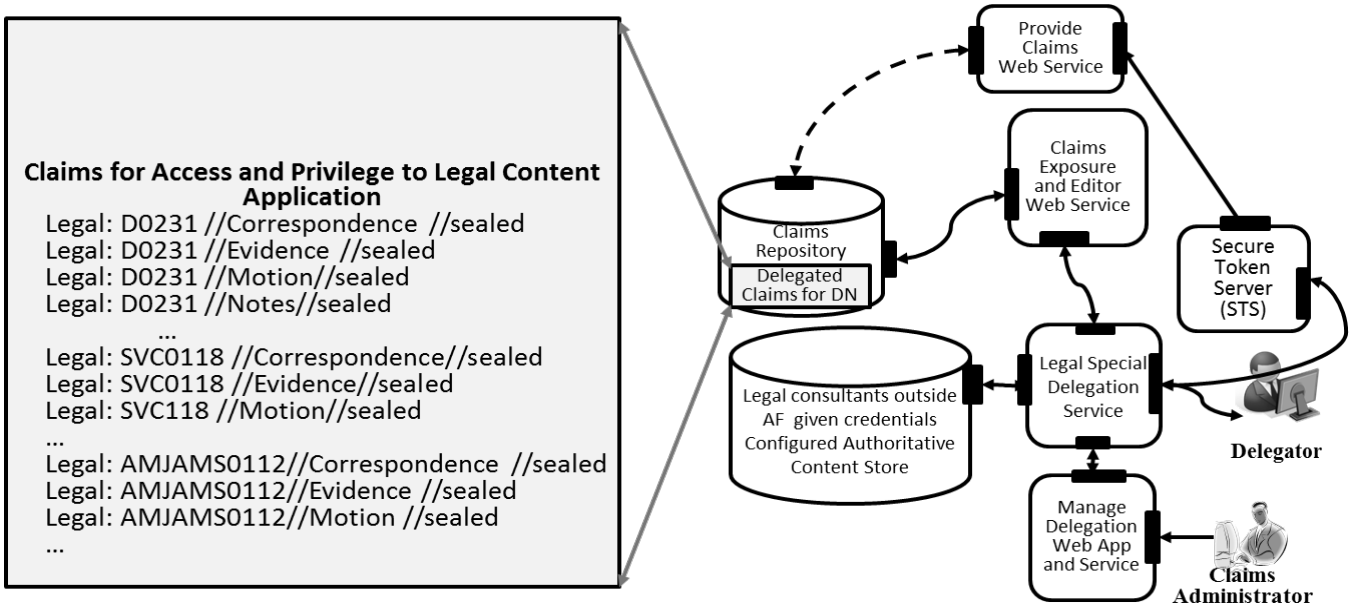


Figure 22 Delegation of Claims Process

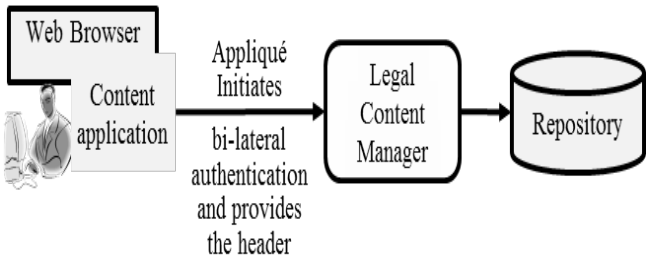


Figure 23 User at a Browser Requesting a Document

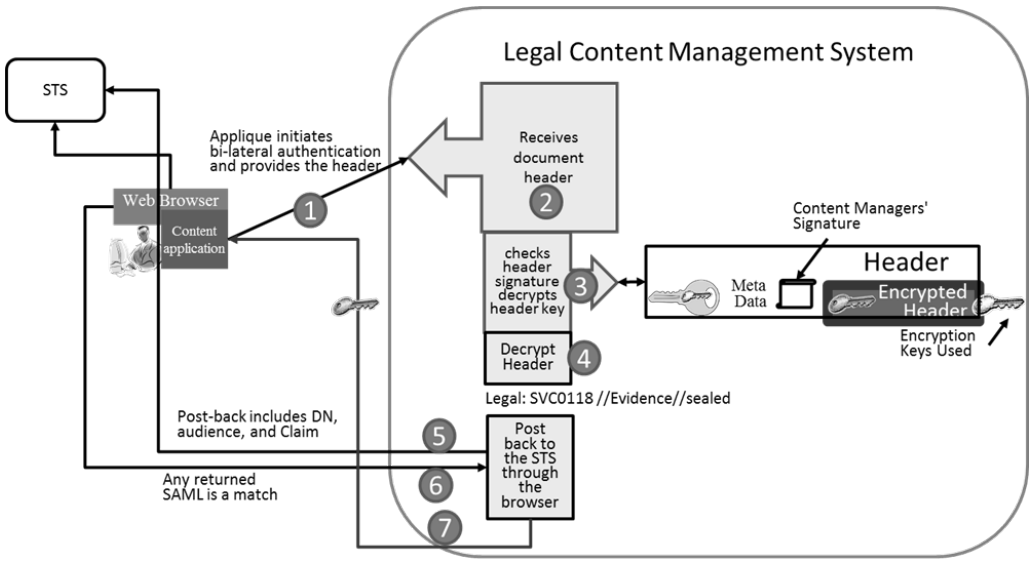


Figure 24 Access to a Document