2017 31st International Conference on Advanced Information Networking and Applications Workshops

Enterprise Digital Rights Management for Document Protection

Rajidi Satish Chandra Reddy, Srinivas Reddy Gopu
Tata Consultancy Services
Hyderabad, India
{rajidis.reddy,srinivas.greddy}@tcs.com

***Abstract*—An insider contributes maximum to the leakage of sensitive information knowingly or unknowingly in an enterprise. Therefore, the need for persistent protection of such information is critical. At the same time, enterprise friendly features for rights and ownership management are of great importance to a digital rights management (DRM) system considering the dynamism of enterprise workforce in a large enterprise. The current enterprise DRM systems have emphasized on offering persistent protection but do not support integration into existing enterprise infrastructure and workflows. In this paper, we present the enterprise digital rights management (eDRM) system that provides persistent protection to documents using cryptographic methods, and also includes enterprise friendly features.**

Keywords—security; protection; usage; rights; enterprise

I. INTRODUCTION

A variety of access control models have emerged over the years with the purposeful changes in the computing systems architecture and proliferation of the digitized services [1], [2], [3], [4], [5], [6]. In a content management environment, to protect copy rights of digital data and control piracy, systems must provide persistent protection with fine-grained and on-going access control [6]. Thus, digital rights management (DRM) model, which offers support for persistent protection emerged [7].

In the literature, there are many definitions for Digital rights Management (DRM). The DRM focuses on persistent protection of digital data and it is everything that can be done to define, manage, and track rights to digital content [8]. In another paper, DRM is defined as the business and technological process for controlling and managing rights to digital intellectual property [9]. The DRM techniques mainly have two applications:

1. Content distribution and management to control piracy
2. Access control and rights management for sensitive documents within enterprise.

Our work focuses on the needs of the second application.

Even though the general DRM requirements for consumer space and enterprise are similar, they differ in their operating environments. The DRM systems have mainly been used by record companies to protect music sold on the Internet. The general DRM requirements for the enterprise differ with the consumer space because enterprises do not need to cater for the flexibility required by consumers as enterprises operate in a more closed environment. Enterprise Digital Rights Management (eDRM) refers to the use of DRM technology to control access to corporate documents (Microsoft Word, PDF, MS Excel, etc.), rather than consumer playable media. Though most of the Enterprise DRM systems [10], [11] offer support for persistent protection for various document formats, they have not given much importance to the enterprise friendly features[1].

The prime motivation for this work is that while there are many enterprise DRM products available in the market today, they do not support integration into existing enterprise infrastructure and workflows, and have a feature set that does not take into account the dynamism of an enterprise workforce, its movement between projects and geographies and its role changes.

In this paper, we propose the eDRM system that has the following distinct properties:

1. Secure distribution of keys (User and Server) that are used for encryption/decryption of document content and communication between the eDRM client and the server**. This is discussed in sub-section V.B**
2. Secure transmission of messages between the eDRM client and server to ensure authenticity of the user and server, integrity and confidentiality of the message. **This is discussed in sub-section V.C**
3. Restriction on derivative objects, which may be created as a result of opening a document. **We have discussed about this in sub-section VI.B**
4. Ongoing controls to enforce document locking, rights revocation. **This is discussed in sub-sections VI.C and VI.H**
5. Inclusion of enterprise friendly features, which increase efficiency, scale in a large enterprise. **This is discussed in the section VI**.

The rest of the paper is organized as follows: We discuss the related work in section II. We describe the problem and the need for digital rights management in section III. We describe the eDRM architecture, security model, the eDRM features and performance results in sections IV, V, VI and VII. Finally, we present the conclusions.

[1] If not specified otherwise, usage of "Enterprise features or Enterprise friendly features" denotes a list of features such as delegation, transfer of ownership, rights manager, temporary ownership, rights assignor, user locking, document locking, offline access and rights request.

978-1-5090-6231-7/17 $31.00 © 2017 IEEE
DOI 10.1109/WAINA.2017.48

II. RELATED WORK

Yang et al. [9] reviewed DRM architecture and commercial DRM systems: Windows RMS; Liquid Machine; Authentica's page recall. They also described design, implementation of a system called Display-Only File Server (DOFS). The DOFS enforces stronger protection to enterprise sensitive content by isolating the content bits from the end user's desktop. This solution intends to concentrate most of the computations about file access and right enforcement on a well-protected server instead of on each individual client machine. Beek [10] discussed about various EDRM systems with respect to EDRM requirements with a purpose. Beek concluded that Windows RMS was the most complete solution for EDRM requirements and DOFS was least appropriate system because of integration and other issues.

Zeng et al. [11] examined various DRM systems: Adobe life cycle policy server; Oracle rights information Rights Management (formerly sealed media E-DRM); Microsoft windows Rights management services; Documentum IRM services (Authentica); Liquid machines document control; Secure2trust (S2T); PDF document security from LockLizard; Workshare protect. Their analysis was based on perspectives about product motivation, administrative models, user monitoring, encryption and access authentication, offline data access, tamper-proofing and platform support.

Recently, many attempts have been made to enhance security and improve efficiency in authentication [13], [14], [15], access control [16], computation, and storage requirements [17] of E-DRM systems. Our eDRM system focuses on improving efficiency and productivity in performing job functions by providing enterprise friendly features without compromising on security aspects.

We have also come across two Enterprise DRM products such as Seclore[2] and WatchDox[3]. Seclore's Filesecure is an information rights management product that offers protection support to various file formats such as PDF, Open office, image formats. The WatchDox's Enterprise File sync and share is another product that provides support to securely sync files, share documents and work with files by applying access controls.

Most of the commercial EDRM systems, based on the four layered model [12], provide basic features but provide limited support for enterprise features. Also, existing eDRM systems have not talked about restricting derivative objects such as temporary files created as a consequence of opening a document. Our eDRM product differentiates from other products in these aspects.

III. PROBLEM DESCRIPTION

Organizations use perimeter-based security methods, and encryption and authentication technologies to protect sensitive data. These methods help organizations to provide either "broad or no" access to data, lacking usage control [6]. Despite of proper access controls, an authorized user may steal confidential information from the unprotected derivative objects such as temporary files.

Enterprise features help in improving productivity with efficiency and scalability in the enterprise systems and also offer convenience to users. For instance, if an employee in a managerial role resigns from a company or transferred to a different location, "transfer ownership" feature will increase efficiency in productivity as it will transfer ownership of documents to the new employee. The "Rights request" feature reduces the cycle time of contacting the grantor for rights, especially, when a rights manager can do the rights assignment much faster. In some of the DRM systems [10], rights are encrypted with digital content. In this scenario, granting additional rights requires the entire document to be retransmitted to the requestor which is not scalable and efficient in a large enterprise.

IV. ENTERPRISE DIGITAL RIGHTS MANAGEMENT

We have developed the eDRM system[4] that provides protection to documents in a corporate environment using cryptographic primitives. The eDRM also displays the decrypted contents in a secure Viewer which restricts the operations that can be performed on the content. It also includes enterprise-friendly features for right and ownership management.

A. Architecture

In this section, we describe the eDRM system architecture. The eDRM system comprises five major components: eDRM client; Administrative client; eDRM server; eDRM Webserver; eDRM database. The "Fig. 1" illustrates the eDRM architecture, its various entities and the communication flow between the components.

1) eDRM client

The eDRM client, a java application on the users' system, allows a user to create, view an eDRM document, and enforces restrictions for a specific user and document pair. It comprises security, secure viewer and rights manager modules. The "Security" module consists of cryptography libraries used to protect a document. The "Secure Viewer" module renders the content of a document and exercises user rights. The "Rights Manager" module allows assigning rights on the document. The eDRM client provides two login mechanisms for user authentication: user certificate; windows logon.

2) Administrative Client

The eDRM Administrative Client allows the designated eDRM administrators to do various operations, such as document and user locking, transfer of ownership, generate user keys.

3) eDRM Server

The eDRM server is an intermediate server that takes care of all communication between the client and the database. It entertains all client requests and understands the client communication protocols. The server interacts with the eDRM database as well as an LDAP-compliant directory service (such as the Windows Active Directory Service), which is used to integrate with the existing user and group infrastructure.

[2] http://www.seclore.com/seclore-edrm/

[3] http://www.watchdox.com

[4] The current version supports protection for pdf, open office documents on Windows and Linux operating systems.

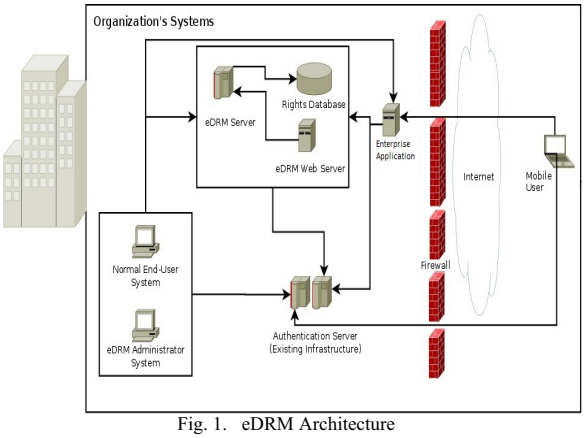

Fig. 1. eDRM Architecture

4) eDRM Web server

The eDRM Web server has two main features: deploy latest versions of the eDRM components (eDRM client, eDRM Server, Administrative Client); hosts the eDRM user key generation web application that provides the key generation utility for the users. The eDRM user key application is integrated with an enterprise portal.

5) Database

The database holds all information about the protected documents that have been created and registered using the eDRM application. It holds the rights that have been assigned to various users on the documents and the keys required for decrypting the files.

V. SECURITY

The eDRM security module utilizes known crypto protocols to ensure confidentiality and integrity of information. It also ensures authenticity of users and the eDRM server by using crypto keys. In this section, we discuss about cryptography specifications, key management and the communication between the eDRM client and the eDRM server. The "Table I" gives overview of notations used in the document.

TABLE I. NOTATIONS

Notation	Description
B, C, D, E	Boolean: True or False; Cipher text; Decryption; Encryption
d, Comp	Data; Comparision function
ID	Identifier
K	Key; Initialization Vector
RQ, RP	Request; Response
RQ_d, RP_d	Request data; Response data
PK, SK	Public Key; Private key
U_{PK}, U_{SK}	User Public key, Private key
ES_{PK}, ES_{SK}	eDRM server public key, private key
RQ_K,DOC_K, RP_K, OF_K	Request;Document;Response; Offline :A random 256 bit key
CON	Document Content
O_{ID}, U_{ID}, DOC_{ID}	Owner, User, Document Identifier
OF_{DB}, COF_{DB}	Offline database; Encrypted offline database
COF_K	Encrypted: offline key;
O_{name}	Owner name
UR, U_{DEL}, U_{DE}	User Rights; Delegator, Delegatee

A. Cryptographic Specifications

The algorithm for asymmetric cryptographic operations will be RSA, with PKCS1v1.5 padding. The algorithm for symmetric cryptographic operations will be AES-256. The cipher, configured in cipher block chaining (CBC) mode (128 bit IV), uses ISO 10126 padding. The algorithm for hashing is SHA-256.

Even though various cipher suites are enabled, the default cipher suite will be TLS_RSA_WITH_AES_128_CBC_SHA, since RSA for cryptographic operations is being used. The eDRM uses PRNG Algorithm for random number generation.

Some eDRM signatures are not pure signatures, as defined by the RFCs, but only an encrypted hash. These will be mentioned explicitly. Otherwise, the signatures are generated using SHA256withRSA.

All keys are generated using a cryptographically strong random number generator. The key sizes currently supported are: user (RSA) keys would be 2176 bits, whereas the server key would be 2048 bits; the length of the AES algorithm key, which is used to encrypt and decrypt document data, is 256 bits.

B. Key Management

The eDRM system utilizes user and server keys to ensure: confidentiality and integrity of a document; authenticity of the eDRM user and server. The eDRM system offers two ways for generating the user keys: the admin user, using the administrative client, can generate keys for the eDRM users and distribute it; the eDRM user can generate the user key using the "eDRM user key" web application, which binds the user private key (U_{SK}) in a java key store (JKS). The eDRM system has a utility that is used to generate the eDRM server key at the time of eDRM deployment. The eDRM server maintains the user public key (U_{PK}) and the eDRM server private key (ES_{SK}). When a user logs into the eDRM client, it sends a request to the eDRM server for the eDRM server public certificate (ES_{PK}). The eDRM server transmits the certificate to the eDRM client and the ES_{PK} is stored on the user's system.

C. Request and Response

The communication between the eDRM client and the eDRM server happens over a secure channel using TLS. On top of this, to ensure the confidentiality, integrity and authenticity of the entities, the eDRM system uses the user and the server keys for encryption and signing the request (RQ) and response (RP).

1) Request Encryption and Decryption

1. The request data RQd is encrypted with RQ_K using AES algorithm.
   $CRQd \leftarrow E(RQ_K, RQd)$
2. The RQ_K is then encrypted with ES_{PK} using RSA algorithm.
   $CRQ_K \leftarrow E(ES_{PK}, RQ_K)$
3. Using a signing algorithm, signature is generated on U_{ID}, CRQ_K, CRQ_d

SIGRQ ← SignU_{SK}(U_{ID}, CRQ_K, CRQ_d)

4. The eDRM client forms a request RQ ← (U_{ID}, CRQ_d, CRQ_K, SIGRQ) and transmits RQ to the eDRM server.
5. The eDRM server verifies the signature using a verification algorithm that outputs B → {True , False} B ← VRFYUpk(RQ, SIGRQ)
6. Decrypts the RQ_K: RQ_K ← D (ES_{SK}, CRQ_K)
7. Decrypts the Request data: RQd ← D(RQ_K, CRQ_d)

2) Response Encryption and Decryption

1. The response data RP_d is encrypted with RP_K using AES algorithm: CRP_d ← E(RP_K, RP_d)
2. The RP_K is then encrypted with U_{PK} using RSA algorithm: CRP_K ← E(UP_K, RP_K).
3. Using a signing algorithm, signature is generated on U_{ID}, CRP_K, CRP_d: SIG_{RP} ← SignES_{SK}(U_{ID}, CRP_K, CRP_d)
4. The eDRM server forms a response: RP ← (U_{ID}, CRPd, CRP_K, SIG_{RP}). It transmits RP to the eDRM client.
5. The eDRM client verifies the signature using a verification algorithm that outputs B ← True || False B ← VRFYESpk(RP, SIG_{RP})
6. Decrypts the RP_K: RP_K ← D(U_{SK}, CRP_K)
7. Decrypts the Response data: RP_d ← D(RP_K, CRP_d)

The eDRM system ensures confidentiality, integrity of the communication, and authenticity of entities by encryption and signing the request and response.

VI. FEATURES

A. Document Protection

The user converts a normal document to an eDRM document in order to protect a document. The security module generates a document key → DOC_K. The document content is encrypted with the DOC_K using AES algorithm: Ccon ← E(DOC_K, CON). The DOC_K, is encrypted with the ES_{PK} using RSA algorithm: $CDOC_K$ ← E(ES_{PK}, DOC_K). The eDRM client forms the request data RQ_d → {U_{ID}, $CDOC_K$} and transmits to the eDRM server.

The eDRM server processes the request data RQ_d. It generates a document identifier DOC_{ID} and stores it along with $CDOC_K$ in the eDRM database. The eDRM server sends back the DOC_{ID} to the client. The eDRM client computes a hash of the DOC_{ID}, $CDOC_K$, CCON: DOC_H ← Hash(DOC_{ID}, $CDOC_K$, CCON) and encrypts the DOC_H with the ES_{PK}: $CDOC_H$ ← E(ES_{PK}, DOC_H). It creates a file with extension ".enc" and writes DOC_{ID}, $CDOC_K$, CCON, DOC_H, $CDOC_H$ in the header part of document content.

B. Document View

The user selects an eDRM document and attempts to open in the eDRM client. The eDRM client extracts the metadata in the protected document and fetches DOC_{ID}, $CDOC_K$, DOCH and $CDOC_H$. It forms the "OPEN File" request with the extracted data from the protected document and encrypts the request to the eDRM server.

The eDRM server decrypts the request data CRQ_d with the decrypted RQ_K: RQ_d ← D(RQ_K, CRQ_d). It verifies the integrity of the document content by comparing the hash DOC_H with D(ES_{SK}, $CDOC_H$). If the comparison results to "TRUE" then the eDRM server fetches the $CDOC_K$ and decrypts it to get: the DOC_K ← D(ES_{SK}, $CDOC_K$), and user rights for the combination of the U_{ID} and DOC_{ID}.

The eDRM server forms the response comprising DOC_{ID}, DOC_K and user rights UR and transmits to the eDRM client in encrypted form. The eDRM client decrypts the response; decrypts the content with the DOC_K; renders the content in the secure viewer module and exercises the user rights. The secure viewer utilizes ICEPDF[5], and apache openoffice[6] java api's for rendering the decrypted content of pdf, open office format respectively.

The eDRM system has a mechanism that restricts the user accessing the content in the temporary files. The temporary files may be created as a consequence of opening a document. The ICEPDF has an option to disable temporary file creation for Pdf documents. For open office documents, we use AES encryption to encrypt the temporary files and decrypt the temporary file whenever the open office requires accessing them. Since the temporary files are encrypted, there is no harm even if the user attempts to share them.

C. Rights Management

The "Rights Manager" module of the eDRM client is used to assign rights on a protected document. The owner opens the encrypted document and selects "Rights Manager" option which invokes "Rights Manager" module. The operations that can be controlled in the eDRM system are: Open, Edit, Copy, Print, Offline, Delegate, Screen capture.

The "Right Manager" module allows adding a user, group, or a template to a document. A group is a predefined set of users and a template is a predefined set of users with rights. The owner selects a user and then assigns rights for that user. The eDRM system provides "temporary rights" option where the owner can assign rights for a certain time period.

The eDRM client periodically checks whether or not any changes performed on user's rights and applies rights instantaneously on the user's documents. For instance, if a user's "open" right is revoked at the time when the right is already exercised then the eDRM system will apply the revocation and close the document on the user's system.

D. Working Offline

The eDRM system allows a user to view a protected document in offline mode provided that user has the "offline" permission on the document. This feature is beneficial when a user is out of enterprise. The user has to request offline database in order to view document in offline mode.

1) Request Offline Database

The eDRM users with "Offline" privilege can obtain database of the documents and rights, and cache the offline database OF_{DB} on user's system in encrypted form.

[5] http://www.icesoft.org

[6] http://www.openoffice.org/api

2) Document offline mode

The user, outside the enterprise network, uses this feature to open a document in offline mode. The eDRM client uses same method as described in the section VI.B to decrypt and render the content in the secure viewer. However, it uses OF_{DB} to check integrity and fetch user rights on the document.

E. User Profile

The user can use "User Profile" feature to set the temporary owner and Rights Assigner. This feature is useful when a user is out of office and wishes someone to take ownership of documents or the role of assigning rights.

1) Temporary Owner

Temporary Ownership shall allow a user to act as the owner of all the documents owned by the original owner. Any user can grant Temporary Ownership to another user. In order to do so, the user must specify the expiration day. The temporary ownership will lapse on the chosen day at 23:59:59 in the owner's time zone. If the user who enabled Temporary Ownership has some documents transferred to a new owner, the Temporary Ownership is not affected as other documents owned by that user are still under Temporary Ownership. The transferred documents will continue to be under their new owner's Temporary Ownership, if enabled. If the Temporary Owner had some rights over the owner's documents, those rights are overridden.

The temporary owner will have all the rights of owner except transfer ownership for a certain period of time specified as "Expiry Date".

2) Rights Assigner

A normal eDRM user can choose to set a Rights Assigner. The Rights Assigner obtains the right granting permission on all documents owned by the user who has set the Rights Assigner. The Rights assigner will be able to grant rights on a document but will not be able to view the document.

F. Transfer Ownership

The eDRM system shall allow the owner to be permanently changed to another user. Only the owner of the document or an eDRM administrator can do this operation. This feature is useful for users who have moved to a new job with managerial responsibilities. The new job responsibilities require ownership to project related documents.

G. Delegation

Delegation is an important and useful feature in an enterprise. In the eDRM system, the owner assigns "delegate" permission to a user U_{DEL} so that U_{DEL} becomes "Delegator". The delegator U_{DEL} uses the "Set Delegation" option in the eDRM client to delegate his permissions to another user U_{DE} that becomes "Delegatee". After delegation, U_{DE} will have all permissions except the "Delegate", "Offline" permission. In the case the "Delegatee" already has some rights over the document, the union of their original rights with the delegated rights will be granted when opening the document. In case of conflict, the right that has the longer validity is kept. If the destination user also has the delegation right, that right is preserved, but the user can delegate only their original rights.

H. Locking

1) Document and User Locking

The owner or temporary owner can lock a document in case it is discovered that document rights have been misused. The user selects "Lock Document" option in the eDRM client and selects the document to be locked. In the event it is discovered that any user has misused his rights or leaked sensitive documents, an admin user can lock a user using the eDRM administrative client.

2) Lock Checking

The eDRM client will periodically check, if the documents or the eDRM user is locked. If a document is locked, the eDRM Client will close the document immediately. In the event that the user is locked, the eDRM client will close all open documents, as well as delete the offline database for that user.

I. Rights Request

In an enterprise, a user may have a need to work on protected document. However, the eDRM system does not allow the user to perform the operation because of insufficient privileges. In this scenario, the eDRM provides an option for the user to request rights on the protected document. The user will select an encrypted document, may enter a message to the granter specifying the rationale for the request, and select rights for the request.

The eDRM client runs a utility in the background that polls rights requests for users who is either "Owner", "Temporary Owner", or "Rights Assignor" for their protected documents. It will display a notification for the user in case of any new requests. The user will open "My Requests" screen and check his "Inbox" for the requests. The owner may "grant' or "Deny" or "Partially Grant" the request. The requester can see the status of the request in his "Outbox" in the "My Request screen".

VII. PERFORMANCE RESULTS

In this section, we present the test strategy, test conditions, and response time of various eDRM operations. The "Table II" provides the testing environment details of the various servers in the eDRM system.

TABLE II. TESTING ENVIRONMENT

Server	Description	Operating system	RAM
Web server	Hosts the eDRM userkey application	RHEL 5.6	2GB
Application server	The eDRM server	Windows server 2008, R2 Enterprise Edition – 64 bit	4GB
Database server	TheeDRM database server	RHEL 5.6	4GB

A. Test Strategy and Conditions

For testing the eDRM user key generation, IBM Rational Performance Tester (RPT) [7] was used. Since the eDRM client is a native application, it could not be tested using the RPT tool. We had to create an environment by using tools for performance testing to simulate load on the eDRM server. We created a simulation tool that creates hits on the eDRM server for a combination of users and various eDRM transactions. The regression testing team performed testing under these conditions: 50 concurrent key generation requests; 75 hits/5 seconds on eDRM server using the simulation tool; 10 users manually using the eDRM client; all eDRM transactions included; testing duration - 30 minutes.

B. Response Time

The "Fig. 2" demonstrates the eDRM client, admin client and userkey generation performance results. We have observed that the response time for most of the eDRM client transactions was below 6 sec. We have observed response time around 15Sec for Get offline Database; 6 seconds for Login; 9 seconds for CreatePDF transaction. We have observed 90th percentile response time for the admin client "Login", and "get DocDetails" transactions, which was above the guidelines (6Sec); normal response time for the transfer of ownership, and other transactions.

VIII. CONCLUSIONS

We have presented the eDRM system, its security model, and features. Most of the commercial Enterprise DRM products emphasize on basic features such as protection for many file formats, policy based access control, secured file sharing, and given least importance to the enterprise features and do not take into account of the dynamism of enterprise workforce. We have demonstrated that how our eDRM product provides persistent protection and supports enterprise features. Our future work includes applying DRM concepts in a Data market place where there is a need to control the usage of the data.

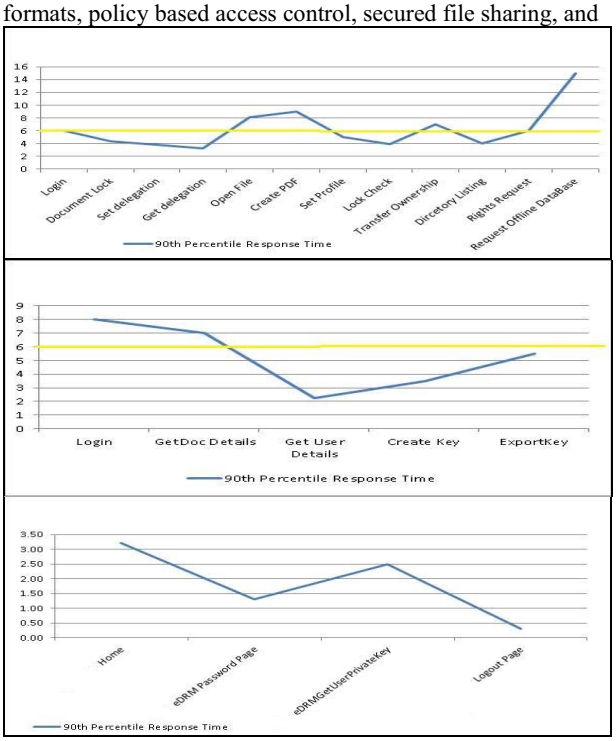

Fig. 2. eDRM client, Admin client, Userkey generation performance results

IX. REFERENCES

[1] Butler Lampson, Protection, In Proceedings of the 5th Annual Princeton Conference on Information Sciences and Systems, pages 437-443, Princeton University, 1971.

[2] D. E. Bell and L. J. LaPadula, Secure computer systems: Mathematical foundations. Technical Report 2547, MITRE, March 1973.

[3] K. J. Biba, Integrity considerations for secure computer systems. Technical Report ES-TR-76-372, Electronic Systems Division, Hanscom Air Force Base, April 1977

[4] [4] D.D. Clark and D.R. Wilson, A Comparison of Commercial and Military Computer Security Policies, In IEEE Symposium on Computer Security and Privacy, April 1987

[5] David F.C. Brewer and Michael J. Nash, The Chinese Wall Security Policy, In Proceedings of the IEEE Symposium on Security and Privacy, pages 206-214, May 1989.

[6] Jaehong Park and Ravi S. Sandhu, "The UCONABC Usage Control Model," ACM Trans-actions on Information and System Security, 7(1):128{174, 2004.

[7] Qiong Liu, Reihaneh Safavi-Naini, and Nicholas Paul Sheppard, "Digital Rights Management for Content Distribution," In Proceedings of the Australasian Information Security Workshop Conference on ACSW Frontiers 2003 (ACSW Frontiers'03), pages 49{58, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[8] A. Arnab and A. Hutchison, "Digital Rights Management - An overview of Current Challenges and Solutions," in Proceedings of Information Security South Africa (ISSA), 2004.

[9] Y. Yu and T.-C. Chiueh, Enterprise Digital Rights Management: Solutions against Information Theft by Insiders, In: Research Proficiency Examination (RPE) report, pp. 2--24 (2004)

[10] Beek, M.H.V.(2007), "Comparison of Enterprise Digital Rights Management systems," Advice report, Aia Software

[11] W. Zeng, S. E. Parkin, and A. van Moorsel, "Digital Rights Management," Technical Report: CS–TR-1223, School of Computing Science, Newcastle University, 2010

[12] E. Diehl, "A four-layer model for security of digital rights management," in Proceedings of the 8th ACM workshop on Digital rights management, ser. DRM '08. New York, NY, USA: ACM, 2008,pp.19–28.

[13] E.-C. Chang, K.-H. Huang, A.-B. Lu, and F. Lai, "Enterprise digital rights management system based on smart card," in 2011 IEEE 15th International Symposium on Consumer Electronics (ISCE), 2011, pp. 363–368.

[14] C.-C. Chang and J.-H. Yang, "A Group-oriented Digital Right Management Scheme with Reliable and Flexible Access Policies.," I. J. Netw. Secur., vol. 15, no. 6, pp. 471–477, 2013.

[15] C. L. Chen, Y. Y. Chen, and Y. H. Chen, "Groupbased authentication to protect digital content for business applications", The International Journal of Innovative Computing, Information and Control, Vol. 5, No. 5, pp. 1243-1251, 2009.

[16] C. C. Lin, S. C. Wu, P. H. Chiang, C. C. Chen, "Enterprise-Oriented Digital Rights Management Mechanism: eDRM", Proceedings of the 2009 International Conference on Availability Reliability and Security, pp. 923-928, 2009.

[17] A. H. Soliman, M. H. Ibrahim, and A. E. El-Hennawy, "Improving security and efficiency of enterprise digital rights management," in proceedings of the 6th IEEE International Conference on Computing, Communications and Networking Technologies (ICCCNT 2015). IEEE, July 2015.

[7] http://www-03.ibm.com/software/products/en/performance