

Research & Development Proposed Investigation



4205 S. Miami Blvd
Research Triangle Park 27703
USA

redpath@us.ibm.com

April 11th 2016

Penetration Study for Digital Identity Card Certificate Management

Prepared for: IBM Mobile Identity Project

Prepared by: Richard Redpath
IBM Senior Technical Staff Member
Mobile Identity Emerging Technologies

Description

Digital Identity Card Certificate Management is controlled by the IBM CMS (Certificate Management Server). The CMS is a Certificate Authority (CA) which manages the issuance of certificates for SSL and DSA verification as well as providing DSA for data. This penetration study is for evaluating the strengths of the security controls in place for authenticating the life cycle of Digital Identity Cards. The study provides a framework in which to examine those aspects of procedural, operational, and technological security mechanisms relevant to protecting the particular aspects of the authenticity of Digital Identity Cards.

Another study is providing for security of delivery, ownership and usage of Digital Identity Cards.

Proposal Number: N/A

Table of Contents

Background	2
Executive Summary	3
Certificate Design	4
Certificate Practice Statement (CPS) Key Life Cycle	15
CMS a Fully Functional Server	19

Background

The security behind digital identities is the primary concern of IBM Mobile Identity. In order to properly sign generated documents, certificates must be created and managed for the Issuer over a time period. The IBM CMS (Certificate Management Server) is the Certificate Authority (CA) used to issue and manage these digital certificates. This study is based on the use of Tomcat for the server platform of the CMS.

Executive Summary

A Digital License Card is a standard ITU-T V3x509 ext OID 1.3.18.0.2.18.6 Certificate employing Military and Commercial (EC192/RSA2048) DSAs with AES128 encryption of privacy extension data unique to each device it resides. The Digital Identity Card Certificate Management is controlled by the IBM CMS (Certificate Management Server) a Certificate Authority (CA) managing the issuance of certificates for SSL and DSA verification as well as providing DSA for data for life cycle of Digital Identity Cards. The CA uses two new 2016 patented Technologies alleviating access to backend sensitive systems to authenticate for a tighter security model. Additionally the Certificate Practice Statement employs a patent pending backend technology to manage the Lifecycle of Certificates for DSA of Digital Identity Information for security strength.

Certificate Design

The structure of a certificate issuing system is important to manage their strength as well as address security issues and repercussion of threats. For convenience and ease of reading the issues that are addressed for a well defined structure are listed as statements that can be reviewed for simple ease of checking coverage. When a threat is outlined [Statements](#) can be referenced as security mechanisms to address.

Statement 1

The Tomcat SSL certificate is configured in the /conf/server.xml file as a PKCS12 file. This file is auto created by the CMS. The PKCS12 file is encrypted with openssl default of triple DES using a password. The password is provided in the CMS Tomcat /conf/cmsibm.cnf file. The Tomcat /conf/server.xml file has the name of the PKCS12 file as well as the password. Standard practice is to have the passwords in these files and the files are protected by the system operations. All SSL certificates are created by the CMS and made available as PKCS12s via HTTPS and controlled connection access by IP as a prerogative of the CMS. The servers that need these SSL certs must know the password for the PKCS12 to use them. Hence if you could get these SSL certs via the network you would still need to know the triple DES password to use them. The SSL certs are signed by the IBM CA of which the CA is available for importing Certificate Authorities to trust the SSL bound to the host (such as a Browser).

Statement 2

All SSL certs are RSA 2048. SSL certs are created for a term of 731 days. The Mobile Identity SSL is used for 360 days before it is renewed and made available for update. The CMS SSL cert is used for 365 days before it is renewed and made available for update. The CMS cert is intended not to overlap update with the Mobile Identity server.

Statement 3

The CMS is the CA and creates a CA term of ten years for signing SSL certs as well as DSA certs for revocation management. The ten year term is typical for a CA. The CA is used to validate trust for a SSL connection as well as the prerogative of using an OCSP.

Statement 4

The DSA certs are signed by the CA as to enable them to be managed for revocation if necessary. If the CA is revoked the DSA certs do not need to be reissued as the certs by themselves are valid since the private key has not been breached. The CA is simply used to manage the chain of revocation list. This avoids impacting the whole population.

Statement 5

A DSA cert is created for a term of 731 days and is used for verification for 540 days before being rotated out of the timeline of usage. This is possible since a DSA cert is created every 170 days for providing a digital signature. A digital signature created by the CMS is assumed to be valid for one year. Since the CMS creates new certs every 170 days the maximum time usage for a cert is 535 days ($365+170=535$) and the strength of the RSA 2048 is consider valid for 730 days.

Statement 6

Mobile Identity require two digital signatures called a DSA pair. Two different technologies are used to create the digital signature pair. An RSA 2048 and a ECDSA 224r1 using SHA256.

Statement 7

The CMS provides a means to enforce population usage limit for a DSA cert pair other than a 170 day term to provide signatures for a population as a prerogative of the Mobile Identity Server for customers.

Statement 8

The CMS stores DSA certs for an issuer in a password protected JKS which contains password protected PKCS12s. The passwords are configured in the /conf/ibmcms.cnf file.

Statement 9

The CMS has a root directory structure for the management of files for the Certs called ibmCA. Simply backing up this directory will secure the state of the system. Backup should be performed every day but CMS only updates this directory every 170 days other than adding new customers.

Statement 10

The CMS is the only creator of signatures for data. This privilege is made secure and contained at the CMS since the CMS manages the certs and their security. There is no partnership with any customer. The CMS does not require any access to customer data other than a message digest to produce signatures as to protect privacy of the customer.

Statement 11

The CMS provides OID 1.3.18.0.2.18.6 extension V3 x509s as a service for the Mobile Identity server for packaging Mobile Identity information for an owner. This V3 x509 is self signed by a CA chained SSL x509 (ibmMOssl) key to enable back trace of the origin of the x509 document as well as verification of trust if necessary. In other words an owner Mobile Identity is a signed security document. See Statement 26 for usage of certs.

	IBM Schema & OID Database	IBM Confidential
	CertificateExtension MobileIdentity	Created by Timothy Hahnon 10/09/2013 at 12:52 PM Last modified by Timothy Hahnon 10/09/2013 at 12:56 PM Click for here authors who have edited this document

General

The following general information is defined by X.500 and IETF standards. A X.509 Certificate Extension represents an extension that can be included as part of an X.509 certificate. Refer to the X.509 standard for details on the usage of X.509 Certificate Extensions.

Required Information

Name: MobileIdentity **OID:** 1.3.18.0.2.18.6

Extension: Always Non-Critical
Criticality:

Description: Mobile Identity Container Object Extension

Optional Information

IBM Description: Mobile Identification object for Secure Identification Document Items.

Format of the data is ASN.1 OctetString

Management

The following information is specific to management within this DB.

Registration Specific

The following is general registration information about the schema definition.

Required Information

Schema Module: [MobileIdentity \(v 1\)](#) **Current Status:** Active

Author: [Richard Redpath](#) [Click to select](#)

Definition's Change Log

- 1 International Organization for Standardization (ISO)
- 1.3 Organization identification schemes registered according to ISO/IEC 6523-2
- 1.3.18 Systems Network Architecture/Open Systems Interconnection (SNA/OSI) Network
- 1.3.18.0 IBM Objects
- 1.3.18.0.2 IBM Distributed Directory
- 1.3.18.0.2.18 X.509 Certificate Extensions
- 1.3.18.0.2.18.6 V3 X.509 extension Mobile Identity Container

Statement 12

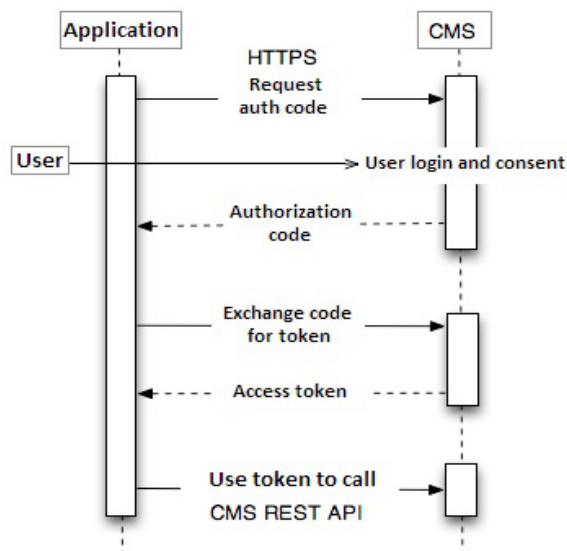
Access to the CMS REST API is https (port 8443) as an option in the /conf/ibmcms.cnf file and it **should be turned on**. IP access (white list) is also an option in /conf/ibmcms.cnf file and **should be used**. Additionally Tomcat Basic Authentication with HTTPS should be used for the REST API services. The tomcat-users.xml file should be augmented with a username and password. In this example the user is "ibm".

```
<tomcat-users>
  <role rolename="tomcat"/>
  <user username="tomcat" password="password" roles="tomcat"/>
</tomcat-users>
```

The web.xml of the Branding servlet should have security constraints added for the complete cms REST API services.

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Wildcard means whole app requires authen</web-resource-name>
    <url-pattern>/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>
  <auth-constraint>
    <role-name>tomcat</role-name>
  </auth-constraint>
  <user-data-constraint>
    <!-- transport-guarantee can be CONFIDENTIAL, INTEGRAL, or NONE -->
    <transport-guarantee>NONE</transport-guarantee>
  </user-data-constraint>
</security-constraint>

<login-config>
  <auth-method>BASIC</auth-method>
</login-config>
```



Statement 13

Revocation is provided by the CMS which will execute procedures to rebuild the necessary artifacts and SSL certs with notifications for updates to the dependent servers.

Statement 14

Secure bluetooth communication is provide simply with an RSA 2048 exchange of a public key for return of an AES128 key for data encryption.

Statement 15

Securing group connection of DSA elements for privacy filtering for delivering data is achieved using a Group Homomorphism the latest security technology. A material equivalence of two groups having homomorphism to X a unique identifier. **US Patent 9230133 B2** issued Jan 5th 2016

$$\begin{array}{ccc} G & & H \\ f(u+v) & \iff & f(u) + f(v) \end{array}$$

where f is the DSA of the data included with x a unique identifier of which there is a homomorphism for the data. If f(u+v) is a valid DSA with x then f(u)+f(v) is a valid DSA each having x and are equivalent. The unique identifier is created for an owner by the Mobile Identity server.

Statement 16

Owner to Verifier communication is RSA 2048 handshake for AES128 symmetric-key for data exchange. Loosely referred to as TLS for transport of data.

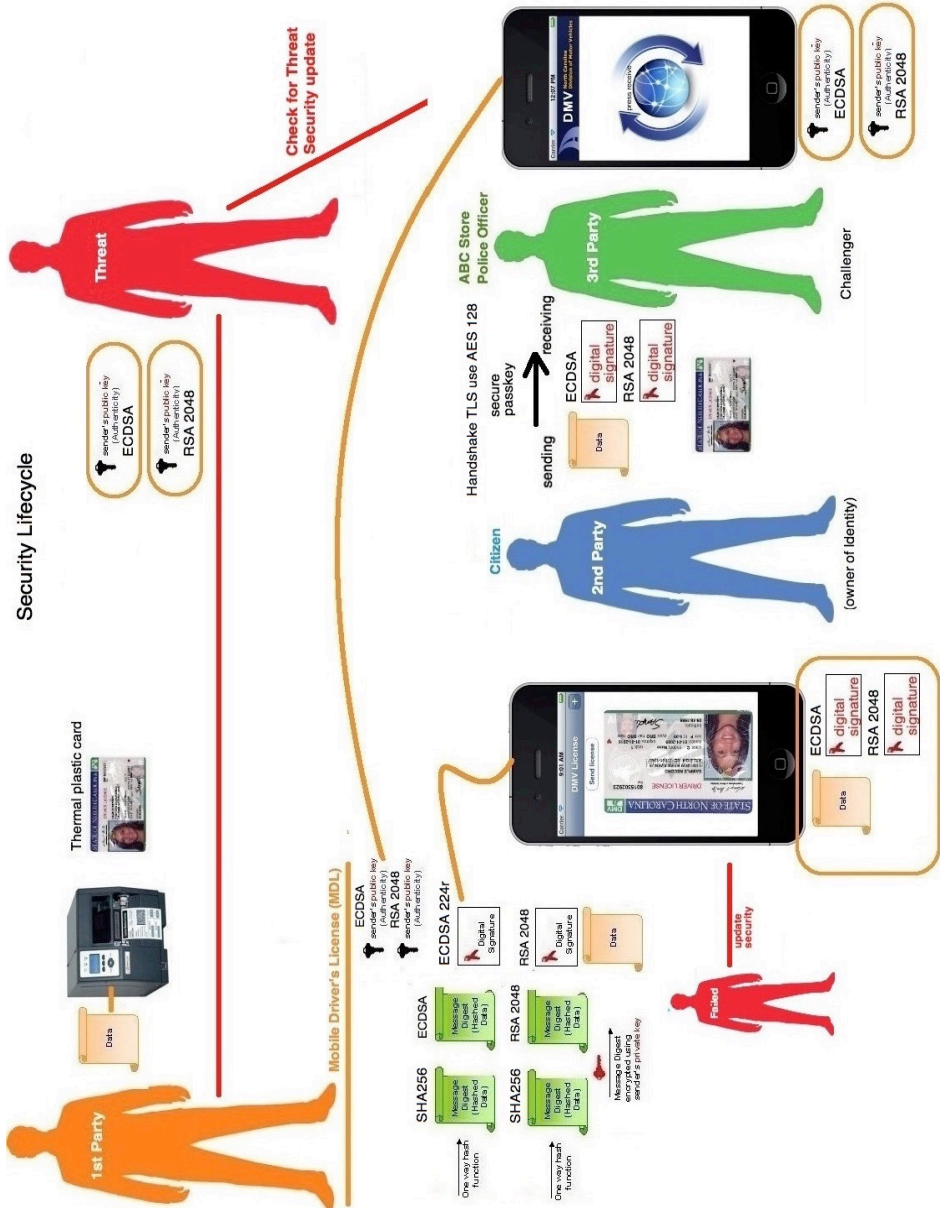
Statement 17

The Tomcat server has a SHUTDOWN (default port 8005) command with password and it should be configured and secured.

Statement 18

US Patent 9065805 B2 issued Jan 23rd 2015. Authentication of Digital Identity Document between owner and verifier.

An overall picture is shown of the structure



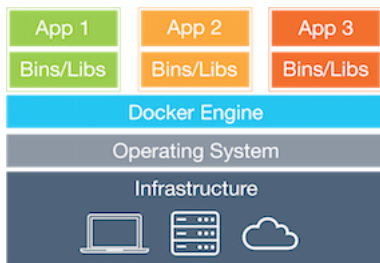
Statement 19

Internal attacker with access to the system. The CMS conf/ibmcms.cnf file has a #whitelist field to identify IP addresses that have access. This field should be set to the CMS local machine of which the authorized administrator should have login access to use a browser for Administrative work locally. Additionally this field should be set to the IP address of the machine that the CMS must service (whitelist machine). The whitelist machine is required to have login access for an Administrator. Overall the CMS is a dedicated backend service machine with limited intranet access. The intent is not to have remote login to the CMS via such processes as Basic authentication it is strictly a service machine with a single port access. For example ports below should be closed.

ftp	21/tcp File transfer
telnet	23/tcp Telnet
smtp	25/tcp Simple Mail Transfer
finger	79/tcp Finger
sunrpc	111/tcp remote process execution
exec	512/tcp remote login (rlogind)
login	513/tcp remote login (rlogind)
shell	514/tcp rlogin style exec (rshd)
printer	515/tcp spooler
uucp	540/tcp uucpd
nfs	2049/tcp network file system
xterm	6000/tcp x-windows server

Statement 20

External attacker with no knowledge of the system. The CMS should have one port address open typically 9292 as a backend service inside an intranet behind a firewall. **HTTPS should be used and the port is 8443 see statement 12.** If Docker is being used the only port accessible for CMS in a Docker Engine to the Operating System is the HTTPS port for the CMS.



Statement 21

External attacker with knowledge of the system. The CMS machine should have a login and stranded on the intranet. The CMS can only service whitelist machines which have login password access. The only access to the CMS machine is via a single open port by the CMS. Whitelist machines should have remote access disabled and be secured behind a firewall. The external attacker can have complete knowledge of the Mobile Identity system design and not penetrate the system if standard security processes are in place.

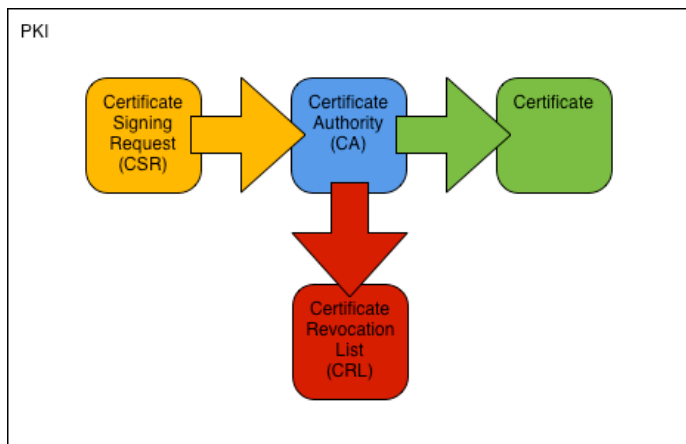
Statement 22

Physical security of the CMS machine must prevent unauthorized access as well as access to machines with backups of the CMS state for recovery.

Statement 23

The setup for a Certificate Authority (CA) is the OpenSSL package FIPS 140-2. The CA manages the creation of keys as well as creation of the certs (V3 X.509). OpenSSL is standard with most operating systems. The Federal Information Processing Standard (FIPS) Publication 140-2, (FIPS PUB 140-2) is a U.S. government computer security standard used to accredit cryptographic modules. The National Institute of Standards and Technology (**NIST**) issued the FIPS 140 Publication Series to coordinate the requirements and standards for cryptography modules that include both hardware and software components.

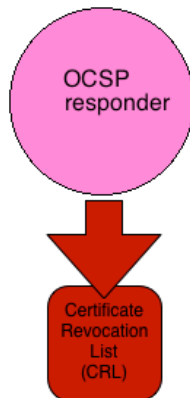
The OpenSSL CA is used to create Certs that are signed by a CA. A picture is shown below which illustrates the enablement of revocation for a complete Certificate Authority system.



This is called the ibmCAcert. The ibmCAcert is rsa:2048 for ten years. The CA is used to sign issued keys for a digital attribute DSA. There are two different types of keys used EC224r and RSA2048 for digital attribute DSA. Compromise must abridge two different technologies for a significant threat.

Statement 24

The IBM CMS is the CA and manages the certificates. It is not necessary to have an OCSP since it is used for indirect customers that have CA signed certs that need to verify their revocation from the (Certificate Revocation List) CRL . The Mobile Identity Infrastructure does not use OCSP requests from verifier devices but periodic checks for updates as well as push notifications to the devices. This enables the intrinsic characteristic of a disconnected environment to prove trust.



The Mobile Identity Infrastructure makes available the CRL and an OCSP SSL certificate if ever needed for someone who wants to integrate it into an OCSP responder. Though the use of the OCSP stapling is not applicable to a disconnected environment but most commonly used for a browser.

Most CAs publish CRLs, but most do not run OCSP responders. A number of public OCSP responders collect CRLs from a number of different CAs and are capable of responding for each of them. Such responders are known as chain responders, and they should only be trusted if their certificate can be verified or if it is trusted and it contains the extKeyUsage extension with the OCSPSigning bit enabled. A reasonably up-to-date list of these public responders is available from <http://www.openvalidation.org>.

For those CAs that run their own OCSP responders, it's best to contact them directly rather than relying on a chain responder, because the information from a CA's responder is more likely to be the most up to date.

Statement 25

The IBM CMS notifications are configured in the `ibmcms.cnf` as
`notifyurl = http://127.0.0.1:8082/`
 for the AMS. The CMS supports HTTPS notifications simply by changing this config. This has not been tested yet since AMS does not have HTTPS available yet. In general the CMS REST API supports HTTPS or HTTP (request) and the notification process (push) supports both.

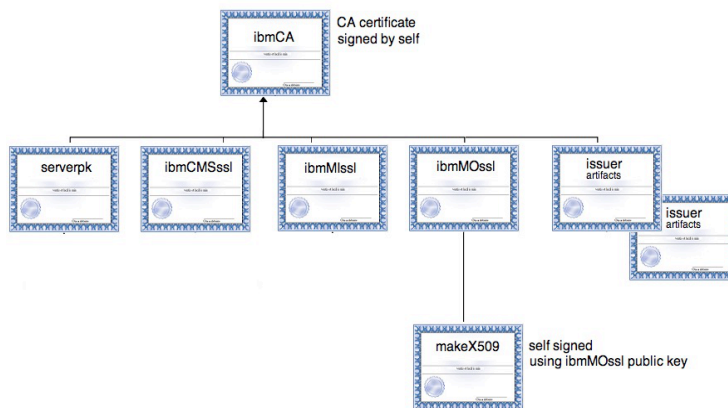
Statement 26

The CMS is the Certificate Authority with the following certificates are managed under CA are shown below of the operations executed for an event such as revocation.

Type	Reissue due	Revoke	Days till reissue	Term	Type
ibmMIssl	Create new ibmMIssl notify MI Server of SSL	Create new ibmMIssl notify MI Server	360	731 NIST 2048	ephemeral
ibmMOssl	Create new ibmMOssl notify MI Server of device cert ibmmo	Create new ibmMOssl notify MI Server of device cert ibmmo	360	731 NIST 2048	linger
Serverpk ibmpk	Create new ibmpk notify MI Server (known as servepk)	Create new ibmpk notify MI Server	700	731 NIST 2048	ephemeral
ibmCMSssl	Create new ibmCMSssl Reboot CMS	Create ibmCMSssl Reboot CMS	365 We do not want the CMS SSL on the same date as MI SSL as to avoid a reboot	731 NIST 2048	ephemeral
ibmCA	Create new ibmCA ibmMIssl ibmCMSssl ibmMOssl Reboot then notify MI Server of ibmca, ibmmo, ibmmi (issuers CRL serial file remains unchanged, issuer certs are not effected)	Create new ibmCA ibmMIssl ibmCMSssl ibmMOssl Reboot then notify MI Server of ibmca, ibmmo, ibmmi (issuers CRL serial file remains unchanged, issuer certs are not effected)	3600	3650 NIST 2048	ephemeral
Issuer	Rotated 180 term or Forceable notify MI Server of artifacts update for Issuer	Create new, rotate out revoked and notify MI Server of artifacts update for Issuer	Rotated 180 max usage intent 540 days	731 NIST 2048	linger

Tables below discuss usage and operations that can be performed

Type	token	Usage
ibmMlssl	ibmmpi	This is the MI Server SSL certificate for Tomcat. If you receive this and you are the MI Server you need to get a new SSL cert. This notification occurs due to expiration or revocation. A P12 is delivered eg: http://127.0.0.1:8080/cms/api/v1/download/ibmmissl
ibmMOssl	ibmmo	This is the cert that is used to sign x509 documents such as a PIB. This is for back tracking if the MI server or device ever wants to validate that the signed x509 originated from us. If you receive this notification it is a prerogative of the Server to get the ibmmo cert that can be used to validate x509s. An x509 is delivered. http://127.0.0.1:8080/cms/api/v1/download/ibmmoss1
ibmCMSssl	ibmcms	This is the CMS SSL certificate. If your receive this Cert and your are the MI Server there is nothing really you do just notification that our SSL has changed;in other words ignore this
Serverpk known as ibmpk	serverpk	This is the serverPK Cert used to sign things and verify transaction. If you receive this and you are the MI Server you need to get the serverPK called ibmpk or isserpk. A P12 is delivered eg: http://127.0.0.1:8080/cms/api/v1/download/serverpk
ibmCA	ibmca	This is the CA root cert used for the SSL certs. This is used by browser to trust where the SSL came from. If you receive this and you are the MI Server you should get the new ibmca and ibmmissl and ibmpk. An x509 is delivered for the ibmCA. eg: http://127.0.0.1:8080/cms/api/v1/download/ibmca http://127.0.0.1:8080/cms/api/v1/download/ibmmissl http://127.0.0.1:8080/cms/api/v1/download/ibmpk
Issuer	issuename	If you receive this and you are the MI Server you should get the new Artifacts for the issuer. This notification occurs for expiration and rotation of certs over time as well a revocation of a cert. A zipfile is delivered with x509s. Eg: http://127.0.0.1:8080/cms/api/v1/artifacts/{issuename}



Type	Keyname/token	Processes
ibmMlssl	ibmml	Revoke serial/renew
ibmMOssl	ibmmo	Revoke serial/renew
ibmCMSssl	ibmcms	Revoke serial/renew
Serverpk known as ibmpk	serverpk	Revoke serial/renew
ibmCA	ibmca	Revoke serial/renew
Issuer	issuename	Revoke serial/renew or force issuer

Statement 27

The IBMserver (AMS) uses a CMS to obtain P12 files (serverpk and ibmmissl). P12 files are managed by the CMS and are triple DES encrypted with a password in the ibmcms.cnf. This is typical for Tomcat config for P12 files. The default passwords are shown below from the ibmcms.cnf. These are required to be changed and the AMS needs to know the change. See [statement 1](#)

```
JKSpass = JKSpasword #The general password for the JKS files to open them
P12pass = P12pasword #The general password for the encrypted P12 files in a JKS to read
CApass = CAPasword #CA private key password
PVpass = PVpasword #Private key passwd for certs
```

Statement 28

Open ports must be secured and they are:

```
8005 #Shutdown port of Tomcat has a password (required to be changed)
8000 #Port used by tomcatserv which only accepts local connections
      this is a startup/reboot program for CMS.
9292 #Port defined for CMS HTTP (this should not work since HTTPS is on)
8443 #Default port used for HTTPS connections
```

Statement 29

Prevent adding a leaf node to the CA chain. The Certificate Issuer, Subject text and Basic constraints are listed for the certs. The `ibmCA` root is enabled to create certs from the Root only. This prevents SSLSNIFF (SSL sniff) for man in the middle to use a chained cert to the Root and add a leaf node. In other words, the `serverpk`, `ibmMIssl`, `ibmMOssl`, Issuer certs cannot add a leaf node to the chain from the `ibmCA` root.

`ibmCA`

Issuer: C=US, ST=North Carolina, O=IBM Corporation, OU=SWG
 Subject: C=US, ST=North Carolina, O=IBM Corporation, OU=SWG
 X509v3 Basic Constraints:
 CA:TRUE

`ibmCMSssl`

Issuer: C=US, ST=North Carolina, O=IBM Corporation, OU=SWG
 Subject: C=US, ST=North Carolina, O=IBM Corporation, CN=www.mi-project.org
 X509v3 Basic Constraints:
 CA:FALSE

`serverpk`

Issuer: C=US, ST=North Carolina, O=IBM Corporation, OU=SWG
 Subject: C=US, ST=North Carolina, O=IBM Corporation, CN=www.mi-project.org
 X509v3 Basic Constraints:
 CA:FALSE

`ibmMIssl`

Issuer: C=US, ST=North Carolina, O=IBM Corporation, OU=SWG
 Subject: C=US, ST=North Carolina, O=IBM Corporation, CN=www.mi-project.org
 X509v3 Basic Constraints:
 CA:FALSE

`ibmMOssl`

Issuer: C=US, ST=North Carolina, O=IBM Corporation, OU=SWG
 Subject: C=US, ST=North Carolina, O=IBM Corporation, CN=mobile.bt.ibm.com
 X509v3 Basic Constraints:
 CA:FALSE

Issuer:

Issuer: C=US, ST=North Carolina, O=IBM Corporation, OU=SWG
 Subject: C=US, ST=North Carolina, O=IBM Corporation, CN=www.mi-project.org
 X509v3 Basic Constraints:
 CA:FALSE

`make X509`

Issuer: C=US, ST=NC, L=RTP, O=IBM, OU=IBM Emerging Technology
 Subject: C=US, ST=NC, L=RTP, O=IBM, OU=IBM Emerging Technology

Certificate Practice Statement (CPS) Key Life Cycle

The Certification Practice Statement (CPS) for the IBM Certificate Management Server (CMS) is the practices which a Certification Authority employs in issuing and managing certificates. The Certificate Policy (CP) is a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements.

The CP for CMS is an Authenticity of Digital Identity information via DSA (Digital Signature Algorithm) a mathematical scheme based on asymmetric cryptography commonly referred to as Certificates for cryptography. The IBM CMS is a standalone Server that is an agent that issues digital certificates. An agent that issues Certificates is commonly called a CA (Certificate Authority). The CA manages the Lifecycle of Certificates for DSA of Digital Identity Information for security strength defined by a specific time period. The Certification Practice Statement (CPS) is **patent pending** and follows.

Certificates are issued every 180 days (6 months) and the term for a certificate is 720 days (close to 2 years). Certificates should only exist for 540 days (eighteen months) and are rotated out of the list of certificates checked in order of issue. A DML license is valid for one year 365 days. The actual Certificate Management Server uses a 170 day (365+170=535) as to assure the one year used on a certificate is satisfied before being rotated out at 540 days.

This is illustrated below. On day zero the first certificate is issued and used to issue DMLs for the next 6 months (180 days). It is the current certificate that is used to issue Digital Mobile Licenses (DSAs).

The Authorizing application would only have one Certificate on the mobile device to start.

Issue Day	X509.0	X509.1	X509.3	X509.4	current
0	A				A
180	A	B			B
360	A	B	C		C
540	B	C	D	E	E
720	C	D	E	F	F

Table: shows certificates that exist at days

Through this process the number of Digital Mobile Licenses (DMLs) issued on a certificate is spread among a set of certificates as to limit the threat range. If a 180 day range is used to issue new certificates then the maximum number of certs that can exist is four implementing a 540 day rotation process from the oldest cert. If this range is changed to 90 days implementing a 540 day rotation process from the oldest cert then the maximum number of certs is seven.

To address the removal of a certificate due to a compromise the table below is provided for illustration.

The certificates that have been issued is shown below. The current cert for issuing new DMLs is serial number F.

On Day	X509.0	X509.1	X509.3	X509.4	current
720	C	D	E	F	F

Certificate D has been compromised and must be removed on day 750. The Certificate set is now shown below. Certificate F is still being used for new issued DMLs.

On Day	X509.0	X509.1	X509.3	X509.4	current
720	C	D	E	F	F
750	C	E	F		F

On Day 780 Certificate F has been compromised and must be removed. The Certificate set is now shown below. Since F is the current Certificate a new one must be issued Certificate G. Certificate G is now used for new issued DMLs now.

On Day	X509.0	X509.1	X509.3	X509.4	current
720	C	D	E	F	F
750	C	E	F		F
780	C	E	G		G

On day 780 the Artifacts delivered to the Challenger App to authorize DMLs looks like this.

Filename	Cert Serial Number
ecdsapublic.x509.0	C
ecdsapublic.x509.1	E
ecdsapublic.x509.2	G

The security model used by DML uses a ECDSA and RSA Certificate pair. The actual Artifacts zipfile delivered is excerpted below as an example. The client will use these certs to authorize a license.

Filename	Cert Serial Number
ecdsapublic.x509.0 RSAPublic.x509.0	C1, C2
ecdsapublic.x509.1 RSAPublic.x509.1	E1, E2
ecdsapublic.x509.2 RSAPublic.x509.2	G1, G2

The server stores the RSA and ESDSA key. These are numbered for rotation.

Name	Date Modified	Size
ecdsakey.pem.0	Oct 12, 2012 6:19 AM	278 bytes
ecdsapublic.x509.0	Oct 12, 2012 9:11 AM	1 KB
rsa.pem.0	Aug 31, 2012 6:50 AM	887 bytes
RSAPublic.x509.0	Aug 31, 2012 8:52 AM	1 KB

The server should have a pair that is marked as the current to be used for processing licenses.

More About Rotation Model

The Table below shows the result of addressing a compromise. On day 780 we have Certificates C, E and G. Certificate C was issued on Day 360.

The rotation model must look daily at Certificate zero to know if there is a rotation needed. On day 780 Certificate C is 420 days old, so it does not need to be rotated. Certificate C should be rotated out on day 900.

On Day	X509.0	X509.1	X509.3	X509.4	current
720	C	D	E	F	F
750	C	E	F		F
780	C	E	G		G

Below is a table that illustrates the history of issuing Certs and addressing the compromise.

- Certificate D has been compromised on day 750
- Certificate F has been compromised on day 780

On Day	X509.0	X509.1	X509.3	X509.4	current
0	A				A
180	A	B			B
360	A	B	C		C
540	B	C	D	E	E
720	C	D	E	F	F
750	C	E	F		F
780	C	E	G		G
900	E	G	H		H
1080	E	G	H	I	I

Table: shows certificates that exist at days

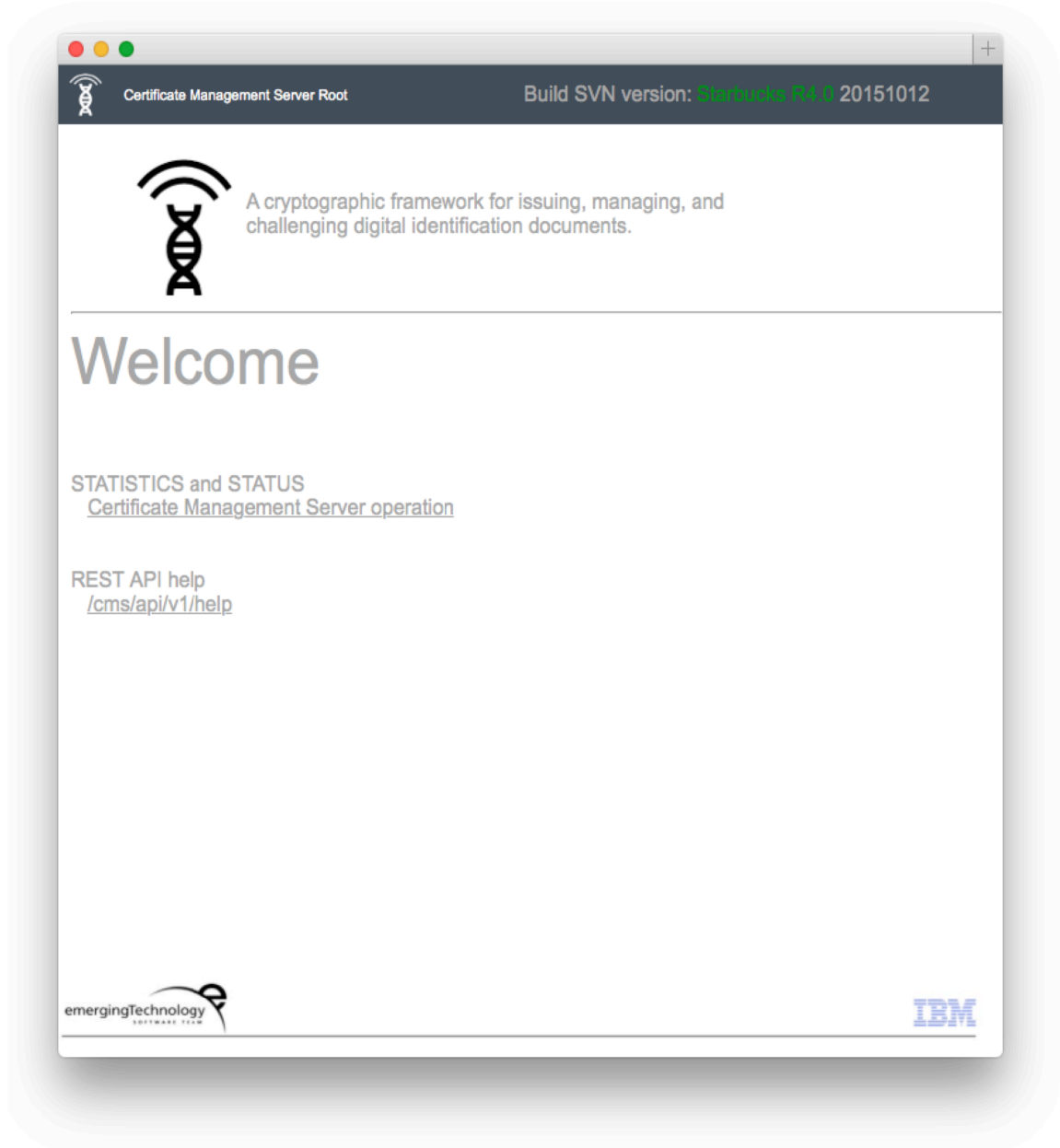
The rotation period should have some buffer for overlap and so the 540 Days should be 550 days. New Certs are created every 180 Days, rotation is 550 days. This is done in this way since a DML can be issued on the last Day of a Cert and be valid for 365 days which pushes the rotation to $180+365=545$.

Best to keep the rotation at 550 days
Issue Certs every 180 Days
DML is valid for 365 Days

The actual Certificate Management Server uses a 170 day ($365+170=535$) as to assure the one year used on a certificate is satisfied before being rotated out at 540 days.

CMS a Fully Functional Server

The CMS is a fully functional server for administration such as status information, revocation,



population forcing and Administration mail announcement (for example). The Server provides a root Main splash screen as a starting point shown below for links.

The statistics page is obtained through this link on the Main Splash page.

STATISTICS and STATUS Certificate Management Server operation

```

Certificate Management Service Status
=====
version: DEV DRAFT STARBUCKS 28 September 2015

  tool directory: /usr/etc/cmaroot/bin
  tomcatserv: on port [8000]
  Time started: Tue Oct 27 15:55:33 EDT 2015
  Current Time: Mon Nov 16 08:33:44 EST 2015
  CMS Monitor state: TIMED_WAITING (daemon)
  Periodic update rate: 86400 seconds (24hours 0min 0sec)
  Total updates: 19
  Next update: 22961 seconds (6hours 22min 41sec)
  MI Domain: www.mi-project.org
  MI Server: 127.0.0.1
  MI Server port: 8082
  MI Server status: No 'pingurl' set in ibmcms.cnf file
  HTTPS only: false
  White list: none
  cleanup: true
  sign: 0
  makex509: 0
  sendmail: 42
  forces: 0
  revoked: 0
  Pending notification: ibmmi,ibmca,serverpk,ibmcms,ibmmo,ibmocsp,sample,ncvessel

ibmca(3600) ibmocsp(365) ibmssl(365) serverpk(700) ibcmssl(365) ibmssl(360)
IBM:
  ibmca [ffa74ab6e38965e0]
    days old: 277
    Certificate not before: Wed Feb 11 08:48:21 EST 2015
  ibmocsp [14b78e5cf5e]
    days old: 111
    Certificate not before: Tue Jul 28 06:40:02 EDT 2015
  ibmssl [14b78e5cf5d]
    days old: 111
    Certificate not before: Tue Jul 28 06:40:01 EDT 2015
  ibcmssl [14b78e5cf5c]
    days old: 111
    Certificate not before: Tue Jul 28 06:40:01 EDT 2015
  serverpk [14b78e5cf62]
    days old: 111
    Certificate not before: Tue Jul 28 06:59:05 EDT 2015
  ibmssl [14b78e5cf61]
    days old: 111
    Certificate not before: Tue Jul 28 06:58:00 EDT 2015

Issuename: ncvessel
Art reqs: 0
CMAsets 1
Aliasindex: rsal447680815244 ecdsa1447680815244 days old 0
  RSA [14b78e5cf68] Cert notBefore: Mon Nov 16 08:33:35 EST 2015
  ECDSA [14b78e5cf67] Cert notBefore: Mon Nov 16 08:33:35 EST 2015
  
```


Certificate serial numbers are shown for the CA as well as security settings and statistics.

Administration Commands

The Commands page is obtained through this link on the Main Splash page called

REST API help
</cms/api/v1/help>

Certificate Management Server Root
Build SVN version: [Starbucks R4.0](#), 20151012



A cryptographic framework for issuing, managing, and challenging digital identification documents.

Help

GET commands:

<p>/cms/api/v1/help</p> <p>/cms/api/v1/download/(ibmmissl ibmca serverpk)</p> <p>/cms/api/v1/data/(ibmmissl publiccrl ibmca serverpk)</p> <p>/cms/api/v1/artifacts/{issuername}</p> <p>/cms/api/v1/cmd/(stat testmail ping help restart reload trace export pushnotify)</p> <p>/cms/api/v1/issuer/(add remove force date)/{isname}</p> <p>/cms/api/v1/cert/ (revoke)/{serial}</p>	<p>This help page</p> <p>Download data</p> <p>Stream data</p> <p>Download artifacts (x509s) of issuer for verifier</p> <p>Command keyname</p> <p>Add, remove or force an issuer</p> <p>Revoke a certificate</p>
--	---

POST commands:

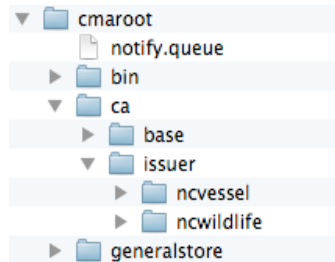
<p>/cms/api/v1/sign makex509 import</p>	<p>POST data to process and return</p> <p>sign URL parameters are the issuer name and the MD pairs</p> <p>issuer=xxx&data=pairs</p> <p>issuer=serverpk&data=rawdata</p> <p>issuer=serverpkold&data=rawdata</p> <p>makex509 URL parameter is the V3 extension data</p> <p>octet-stream</p> <p>import URL multi-part is a CMS exported zip file. Sample POST form code below</p> <pre><form method="post" id="formimport" action="/cms/api/v1/import" enctype="multipart/form-data"> <input class="is key" type="submit" name="import" value="Import CA" id="import" onclick="formvalidation('myform');return false;"> <input type="file" name="file" id="file"> </form></pre> <p>Import/Export CA link</p>
--	---

For internal

The Administrator can use these commands for addressing revocation, forcing population etc.

CMS Backup/Recovery

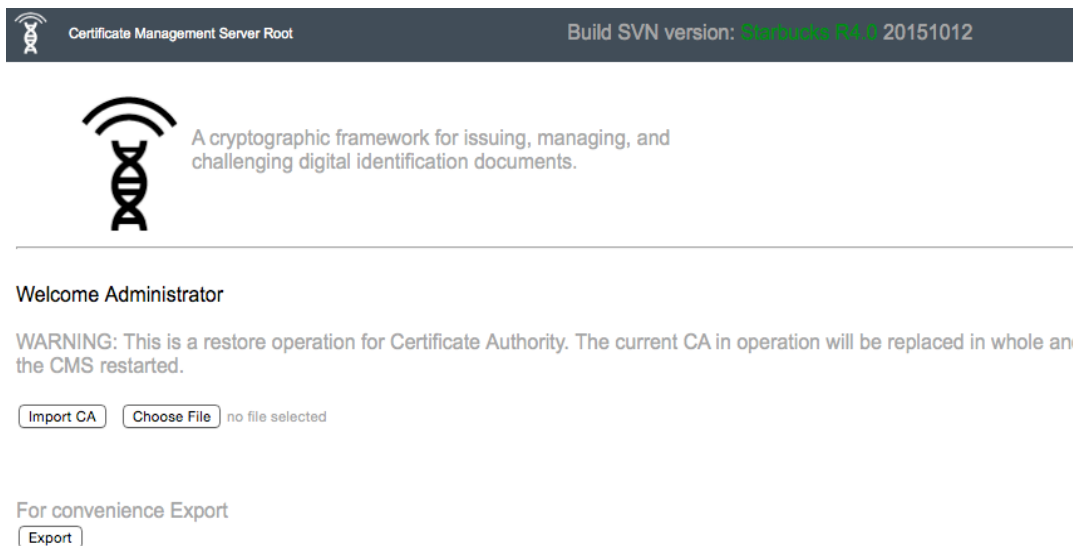
The CMS is a CA with a directory structure designed for backup for state preservation/recovery. Below is a snapshot of a CMS directory structure which has two issuers (ncvessel, ncwildlife) that have been configured by the AMS for an Institution. When an Institution is to go live they should backup this directory and the backup is good for the next 90 days. Best practices is that the Institution should backup daily though in most case this directory would not have changed for 90 days.



Using the Force or Revocation feature of the CMS will update this directory.

Having a CA root directory provides a very simple means to move/copy/restore the installation of a CMS preserving the state. The CMS provides a page for the commands Import/Export of the CA root. Of course access to this page is controlled by a white list configuration in the ibmcms.cnf file for the installation of the CMS.

<http://127.0.0.1:8080/cms/cmsimport.html>



Use of the export creates a zipfile and import takes as input a zipfile.