Superdistribution: Testability, Security and Management of Digital Applications

Vijay Anand
Industrial Engineering and technology,
Southeast Missouri State University,
Cape Girardeau, USA 63701
vanand@semo.edu

Jafar Saniie
Electrical and Computer Engineering Department
Illinois Institute of Technology
Chicago, USA, 60616

***Abstract*—Application frameworks have transitioned from isolated computing units to mobile, distributed and connected cyber infrastructure. To meet the needs of mobility and distributive storage and execution application frameworks need to support application distribution and updated mechanisms that can be managed effectively and securely. An application distribution update and execution schemes for consumers such that each and every consumer gets the similar application experience is termed as superdistribution. Superdistribution classically was accomplished by using digital media which required special hardware and software for consumption. Digital content in today's cyber infrastructure have transitioned from basic consume only content to interactive updatable content. Hence controlling superdistribution to guarantee security requirements of application infrastructure is critical for the success of the application content framework. Also important is the ability of the framework to provide a testing framework for application development. With increased mobility and perpetual connectivity the problems of distribution and management digital content in a secure way for different cyber architectures are significant. In this paper we show how the constructs of X509 certificate can be used to manage superdistribution frameworks. We also show the usage of X509 certificate constructs to allow local distribution for application development and its limitations.**

Keywords: superdistribution, X509, Digital Signature

I. INTRODUCTION

The application stack of the cyber infrastructure has transitioned from isolated and contained operating system centric framework to a mobile, distributed and perpetually connected framework [1]. The storage and consumption of digital assets also has mimicked this transition. Figure 1 represents the operating stack of the cyber infrastructure which tries to deliver the mobile, distributed and perpetually connected experience to end users of the infrastructure [2]. In a classical cyber infrastructure the local computation hardware, with the Operating System (OS) and native applications formed the fundamental computing unit where digital content was being consumed locally. In the current cyber infrastructure the location of computation hardware, operating system and applications, storage of digital assets follows the mobile, distributed paradigm with access provided by ubiquitous connectivity referred to as the cloud. In a cloud based architecture depending on which aspect a computing unit is targeted as highlighted by Figure 1 different classes of architectures are formulated. The different recognized architectures are [3]:

- Architecture to provide computation hardware resources (IaaS – Infrastructure as a Service)
- Architecture to provide an operating framework (PaaS - Platform as a Service)
- Architecture to provide application specific framework (SaaS – Software as a Service)

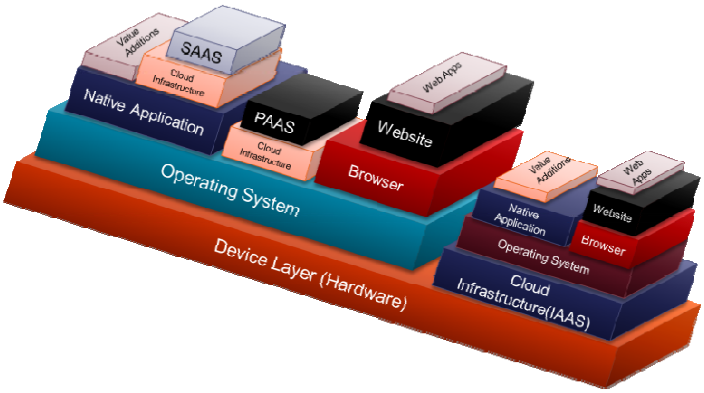

Figure 1: Application Stack of the Cloud (IaaS – Infrastructure as a Service, PaaS – Platform as a Service, SaaS – Software as a Service)

One of the challenges is that of the application distribution to encompass this architectural diversity. The challenges of application distribution are secure distribution of applications, framework for secure application installation and update for end users [4]. An application distribution and update scheme for application usage by consumers such that each and every consumer gets the similar and consistent application experience is termed as superdistribution. This model formulates a departure from the model in a classical cyber infrastructure where applications and digital content were distributed by digital media like DVD, CD etc.. The primary problem in classical digital content was legitimate consumption of digital content which gave way to technologies like Digital Rights Management (DRM)[5]. With the cloud based cyber infrastructure the challenge to provide perpetually interactive content that can be amended brings new avenues for consideration in application distribution. The other departure from a classical content distribution scheme is that of the providing a framework such that digital applications can be created by manufacturers and be sold to end consumers. The DRM framework only provided a one way communication between the content maker and content consumer. In the new age cyber infrastructure content management involves content interaction, content creation in a distributed manner and content update [6].

978-1-4673-0818-2/12/$31.00 ©2012 IEEE

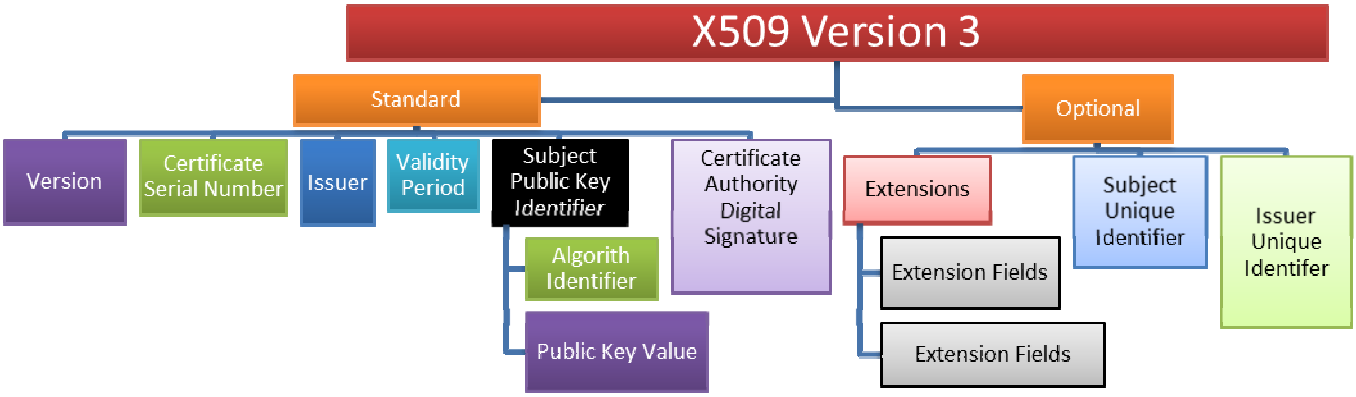

Figure 2: X 509 Certificate Structure

Another challenge for the application framework creator is that of creating a testing framework which provides all the features of the real application while allowing the application to go through a process such that the application framework creator has control and management of the application distribution framework. The most notable examples of such an framework are the Apple App Store [7], Android App Market [8] etc.. All these frameworks have to provide two distribution structures 1) Distribution through the application framework portal, 2) Distribution through the testing framework for application development. The challenge in this architecture is to provide strict boundaries of distribution such that developer applications are not superdistributed. The management of this application framework also implies minimization of risk to the users and the framework itself to malicious applications subverting the acceptable superdistribution process.

To achieve the goals of superdistribution, for testing, for update and revocation in the spirit of mobility and distributiveness of the application framework the public key infrastructure [9] is used. In this paper we show how the certificate handling mechanism of the public key infrastructure can be used for testing and revocation in the superdistribution of digital goods. The certificates in question here are termed as X509 [10] certificates. In Section II we provide an overview and operation of X509 certificates. In Section III we highlight the revocation mechanism of X509 certificates. In Section IV we show how the public key infrastructure be used for superdistribution, testing and revocation.

II. Overview of X509 Certificates

X 509 certificates provide a framework for authentication and key distribution for secure consumption of content in a centralized manner in a public key infrastructure (PKI) [9]. The X509 framework of the public key infrastructure relies on the certificate authority (CA) to issue certificates to users. The issued certificate needs to guarantee that the encryption and/or digital signature mechanism is associated with legitimate users private key in an asymmetric ciphering context. The binding is guaranteed with the association of public key certificates which binds public key values to the subjects as identified by the certificate fields in Figure 2. To create a tight cryptographic association the trusted CA would digitally sign each and every certificate which can be asserted by variety of means. As shown in Figure 2 a certificate has limited valid lifetime and the range of this life time is a part of the signed content of the certificate[10]. The cryptographic check of the certificate and its validity can be autonomously asserted by any certificate checking mechanism (software or hardware) allowing the transmission and storage of the certificate via untrusted channels. The certificate format for such an association was defined the original version of the X509 certificate which were appended with two extra fields to create support of directory access control in the second version. The third version of the certificate adds extension fields which can be populated by various mechanisms, like certificate user organization, standards body etc.

The extensions in the version 3[9] of the X509 certificate can create bindings for information about keys, subject identification, policy and certificate path constraints which, in effect provide us with a framework for adding optional elements for application distribution and control.

These extensions can convey such data as additional subject identification information, key attribute information, policy information, and certification path constraints.

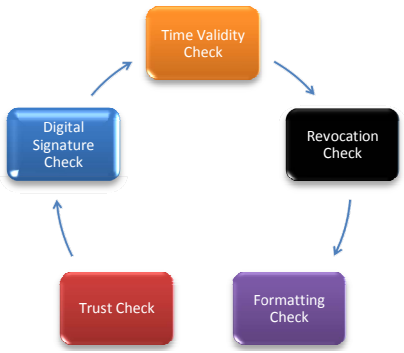

Figure 3: X509 Certificate Checks

An important aspect of certificate usage is the concept of certificate validation. As shown in Figure 3 the various steps of path validation are done by going through different checks of the X509 certificate components:

- Trust Check: A certificate signed by a certificate authority needs to do a trust path validity check [9]. What this implies is that the check of the certificate through various intermediate certificates eventually leads up to the root

certificate of the certificate authority. If any of the intermediary certificates fails then the trust check of the certificate fails.

- Digital Signature Check: For the public key infrastructure the X509 certificate needs to provide cryptographic binding between a public key with that of identity information. The identity information generates a hash which is encrypted using the CA private key. If the computed hash and the encrypted hash do not match signature check fails for the certificate [9], [10].
- Time Validity Check: This check validates whether the certificate has expired or not. Every certificate comes with a lifetime validity period and this check gets the check of time constraint within this range [10].
- Revocation Check: This check validates whether the certificate has been blacklisted by the certificate authority. A certificate authority generates a black list depending upon need of the certificate. To guarantee a certificate is good or not a signed black list of the certificate agency is referred to after other checks are done [9], [10]. The certificate validation fails if the revocation check fails.
- Formatting check: The formatting check guarantees the compliance of a certificate with the X509 RFC certificate format specification [9], [10]. The reason this check is important is to guarantee interoperability between different certificate authorities.

For superdistribution schemes few aspects of the X509 certificates can be used to achieve the goals of distribution, security, testing and mobility. Those schemes are:

- Revocation schemes [11] that provide mechanisms for secure control and management of application signed by the certificate.
- Extensions [9] [12] to provide bindings for testing applications in a controlled way such that the process of application distribution is not subverted.

In the next section we show how the certificate revocation allows us secure control and management of applications signed by a certificate. In section IV we go over the extension bindings to provide a complaint model to test and distribute applications for the application framework.

III. Application Revocation

As highlighted in Figure 3 the various avenues for revocation lies in the realm of certificate checks which can be utilized in a unique way to control and manage application signing. To guarantee secure distribution and management of applications so as to minimize the risk of malicious transgression of an application framework certificate revocation schemes that can be institutionalized are:

- Schemes based on date validity checks
- Schemes based on the certificate revocation checks

The other checks discussed cannot be easily institutionalized with absolute guarantees of application revocation. Application revocation allows two different aspects of control of application framework management. One is to provide a framework to revoke malicious application that has been legally signed and distributed within the application framework. The second is that of mechanism of testing such that the application maker does not subvert the distribution process of the framework maker. To achieve this date validity checks and the revocation checks are discussed in detail.

A. Date Validity Check

A certificate has the field for validity period. If the time of the checking system does not fall under the validity period the check fails. This is referred to as certificate expiration [9], [10]. In an application development framework if the validity of the certificate can be used for testing application behavior modifying the time constraints application testing can be done in a way that does not subvert the distribution process of the application framework maker. Although this solution works it puts a lot of burden on the application developer. The application developer has to continuously reset the time of the testing frameworks for testing the behavior of the application making this a severe limitation on this method. Another shortcoming of this method is there is no way to use the date validity when application needs to revoked in a distributed way.

B. Revocation Check

To provide an elegant way for application revocation various schemes were formulated in the public key infrastructure. These schemes in the PKI infrastructure refer to the Certificate Revocation List (CRL)[9] scheme and the Online Certificate Status Protocol (OCSP) [13]scheme.

- The certificate authority(CA) publishes a data structure called as the certificate revocation list (CRL)[9] which contains a time stamped list of revoked certificates which were signed by the certificate authority. The process of certificate validation involves the download of an updated CRL and a cryptogrpahic check done for the serial number of the certificate on the CRL. Although the freshness of the the CRL is dependant on the update period thereby effecting the time accuracy of revocation list. In the current infrastructure CRL distribution are through a centralized mechanism of distribution(usally by certificate authority) via untrusted communication paths. Another important aspect of a CRL distribution is the size of the CRL for a commercial certificate authority which can be significantly large.
- To overcome the time accuracy and size of the CRL method for revocation of certificates the Online Certificate Status Protocol (OCSP) is utilized [13]. OCSP based certificate verification suspends the certificate

consumption until the online request for certificate status gets a response.

A typical CRL/OCSP based certificate distribution is shown in the Figure 4. As highlighted in the interaction the application framework maker has the root key of the framework. The software makers request certificates whose creation is rooted with the framework maker certificate. The process of certificate validation goes through the trust check path of the software maker certificate with the intermediary certificate of the application framework maker and finally the certificate of the certificate authority. The application framework maker can create a blacklist of any software maker as the framework deems necessary.

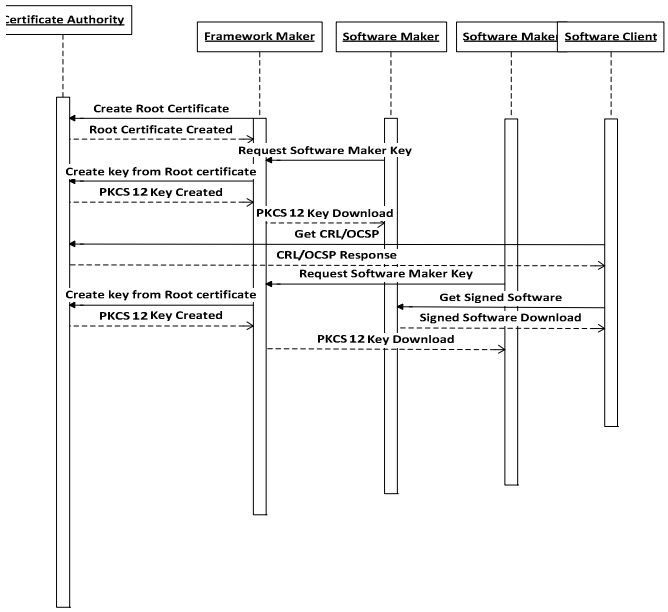

Figure 4: CRL/OCSP interaction

This blacklist is used by the software user to direct the consumption of an application [14]. If the software is blacklisted the user client would terminate the application execution. If the CRL mechanism is utilized then the time granularity of certificate revocation is affected where as in the usage of OCSP the need to perpetual connectivity [13] is mandated. The following section provides usage of constructs of X509 which can be invoked for application testing and distribution.

IV. X509 Extensions For Application Management

The success of an application framework depends on the popularity and acceptance by a third party application software creator as a platform for application commerce. The application developer creates applications which are sold to the end user through a portal either supported by the device maker, OS maker or any third party distribution channel which can host this application for distribution. The challenge is to create of framework that is simple and effective with minimum steps for testing and distribution for application maker in a commercial framework. A simplistic application creation workflow is shown in Figure 5.

The various steps of the application creation workflow are highlighted in Figure 5 and this workflow needs to be seamless with as few steps as possible. A key requirement for the Application Development Kit (ADK) is the ease of application creation and seamless ability to sign development code for testing and development.

Since the application commerce is managed and regulated enterprise, signing becomes an important way to control super distribution or to identify any circumvention of the chain of

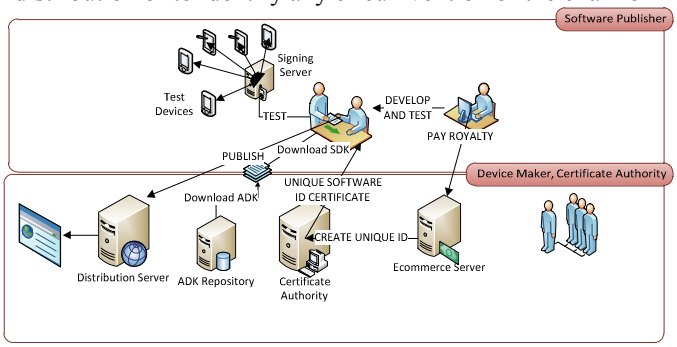

Figure 5: Application Creation Workflow

trust of a device. The various ways to sign applications are:

- **Remote Signing:** Developer can test sign via command line tools or web based upload of applications to a remote server where the uploaded application is signed and sent back to developer. Consistent Server connection is required for all testing situations. In terms of signing methods this is referred to as the PKCS7[15][16] method.
- **Local Signing:** In this scenario developers don't need any server support for signing their applications. Developers can locally sign their application using the ADK and this locally signed application is bound to a set of devices and cannot be used for super distribution. In terms of signing language this refers to a PKCS12 [17][18] signature.

The different ways the testing framework can be supported are:

- Signing by having a connection to the certificate authority server. The CA server in this case requires having signature generation capability by providing upload and downloading mechanism of application binaries. The limitation of testing signatures has to be configured on the server for each software maker making this a complicated management process for the certificate authority.
- Signing remotely by providing an initial list of testing devices during the registration process. This method still requires connectivity for testing as well as server capabilities for signing. What this method eliminates is the management process of testing instances.
- The method to aid local signing would require the creation of two pair of certificates. One certificate with a set of testing client's identities in the optional fields of the X509 certificates. The client testing can be achieved by localized signing of application and then installing them on the client. The certificate validation process has to match the client identity during the certificate validation process. The certificate validation process is shown in Figure 6. This

provides a clean way for application allowing distributed application signing mechanism. The second certificate is used when the application testing has been completed and the application needs to be super-distributed. The flow chart of such a system is highlighted in Figure 6. As it can be seen in the flowchart the software maker can sign any device till the time the device is in the list of devices of the extensions of X509. If the extensions don't have the device ID then there is a mechanism to add it using the certificate authority portal.

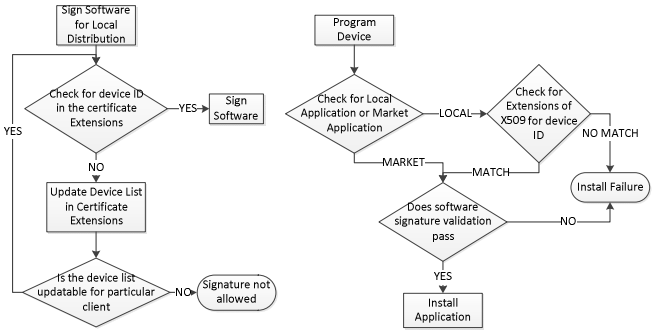

Figure 6: Flowchart for Local Signing and Application Installation

Thus a superdistribution scheme can be created by following the revocation mechanisms recommended by the public key infrastructure. To create a simple application testing [19] and distribution mechanism localized signing provides a reasonable mechanism. The framework maker has the responsibility to revoke software maker certificates in all the above cases. A comparison of the 3 frameworks presented

Table 1: Comparison of Signing Mechanisms

	CA based Signing	Remote Signature	Local Signatures
Network Access	Required for Every Signature	Required for Every Signature	Required for Signature for Unlisted Devices
Management of Devices	Required for every Device every instance	Required only for unlisted devices	Required only for unlisted devices
Revocation	Complicated due to similarity in testing and production certificate	Complicated due to similarity in testing and production certificates	Simple due to dissimilarity in testing and production certificates

To test the validity of the concept OPENSSL is used for testing. A simple certificate authority is setup using the OPENSSL framework [20] which provides the unique certificates with extensions in the certificates having identity information of the clients. Different telnet clients with unique identity are created. The application signatures are generated based on client ID's.

V. CONCLUSION

In this paper, we showed how to manage superdistribution schemes using the public key infrastructure using X509 certificates. We showed how X509 certificates can be revoked for the purposes of testing, superdistribution and legitimate revocation of applications. Although certificate revocation has been known for secure communication channels in this paper we show how it provides an excellent framework for application distribution management and control in a secure way to support testing and distribution. The aspect of localized testing allows the software maker to not be dependent of the certificate authority for testing of applications. Thus the objectives of ease of application development are met along with the goals of superdistribution of an application such that it does not subvert the processes of the framework maker.

VI. REFERENCES

[1] C. Albanesius, "*Smartphone Shipments Surpass PC Shipments for First Time. What's Next?*" PC Magazine, 8 Feb. 2011; www.pcmag.com/article 2/0,2817,2379665,00.asp.

[2] Pallis, G.; , "Cloud Computing: The New Frontier of Internet Computing," *Internet Computing, IEEE* , vol.14, no.5, pp.70-73, Sept.-Oct. 2010

[3] Yau, S.S.; An, H.G.; , "Software Engineering Meets Services and Cloud Computing," *Computer* , vol.44, no.10, pp.47-53, Oct. 2011

[4] Liang Zhong; Chunming Hu; Tianyu Wo; Jianxin Li; Weiji Zeng; , "Soft-Union: An Overlay Based Efficient Software P2P Distribution Scheme," *Cloud Computing (CLOUD), 2011 IEEE International Conference on* , vol., no., pp.740-741, 4-9 July 2011

[5] Schmidt, A.U.; , "On the superdistribution of digital goods," *Communications and Networking in China, 2008. ChinaCom 2008. Third International Conference on* , vol., no., pp.1236-1243, 25-27 Aug. 2008

[6] Fehling, C.; Leymann, F.; Mietzner, R.; , "A Framework for Optimized Distribution of Tenants in Cloud Applications," *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on* , vol., no., pp.252-259, 5-10 July 2010

[7] Security Overview of Apple App Store, https://developer.apple.com/library/ios/#documentation/Security/Conceptual/Security_Overview/Introduction/Introduction.html#//apple_ref/doc/uid/TP30000976

[8] Android Designing for Security, http://developer.android.com/guide/practices/security.html

[9] RFC5280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile

[10] RFC4210, Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)

[11] Naor, M.; Nissim, K.; , "Certificate revocation and certificate update," *IEEE Journal on Selected Areas in Communications,* vol.18, no.4, pp.561-570, Apr 2000

[12] RFC4262, X.509 Certificate Extension for Secure/Multipurpose Internet Mail Extensions (S/MIME) Capabilities

[13] RFC2560, X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP

[14] Wei Liu; Nishiyama, H.; Ansari, N.; Kato, N.; , "A Study on Certificate Revocation in Mobile Ad Hoc Networks," *Communications (ICC), 2011 IEEE International Conference on* , vol., no., pp.1-5, 5-9 June 2011

[15] RFC2315, PKCS #7: Cryptographic Message Syntax

[16] RFC5652, Cryptographic Message Syntax (CMS)

[17] PKCS #12: Personal Information Exchange Syntax Standard, http://www.rsa.com/rsalabs/node.asp?id=2138

[18] Peng Yinghui; , "The Application of PKCS#12 Digital Certificate in User Identity Authentication System," *Software Engineering, 2009. WCSE '09. WRI World Congress on* , vol.4, no., pp.351-355, 19-21 May 2009

[19] McDermid, J.A.; Qi Shi; , "A formal model of security dependency for analysis and testing of secure systems," *Computer Security Foundations Workshop IV, 1991. Proceedings* , vol., no., pp.188-200, 18-20 Jun 1991

[20] OPENSSL CA: http://www.openssl.org/docs/apps/ca.html