

A New Paradigm in PKI Architecture: OTPK Technology For Online Digital Signature

Data Security Systems Solutions Pte Ltd

teikguan@dsssasia.com

efroni@datasecurity3.com

<http://www.dsssasia.com>

In this paper, we present a paradigm shift in PKI architectures. The OTPK concept is alarmingly simple to understand. Whenever a digital signature is required, the private key is generated, certified, used to compute the digital signature and immediately deleted. All that remains is the digital signature and the public key certificate from the Certification Authority (CA) that is used to verify the digital signature. There is no possible compromise on the private key, no need for user smart cards/USB tokens, no need for CRLs, no need for LDAP directories, no need for OCSP. It is compliant to international digital signature laws. The OTPK technology should be evaluated as a new and cost effective solution for on-line digital signature providing full mobility for mass usage of the public in different industries. It should be evaluated for this perspective, not from a CA perspective.

1 Introduction

The growth of on-line activities strongly depends on the reliability and availability of the service. The recent attacks on the on-line services such as phishing, man-in-the-middle and others have forced us to react. To meet the needs, a strong two-factor authentication has been introduced and become increasingly popular. In Singapore and Hong Kong, it is already mandatory to use two-factor authentication in finance sectors. In USA, two-factor authentication solutions were recommended last year. Subsequently, many organizations have been moving to a strong two-factor authentication using different solutions, for example, tokens, Vasco, RSA, Verisign, Active Card and many others. In addition, software tokens such as the OATH tokens, a Java midlet on mobile devices, SMS, scratch card, biometrics have been deployed to protect online applications against the hackers who try to steal users' identities and make use of them.

It is well known that PKI authentication and PKI digital signatures provide the best security for on-line activities for authentication and data

integrity. But as of today, a full PKI solution for applications such as retail internet banking for banks that have a few millions of users have not been successfully deployed. This is because the cost of deployment is huge, the key management is troublesome, annual renewal of certificates and the operation of CA is a 'big headache'. But, can we simplify the process to take the advantages of the non-repudiation by means of digital signature? DSSS introduces a new concept with the OTPK.

The objective of this document is not to introduce a new CA technology. The aim of the paper is to introduce a new flow to implement digital signature, based on the recent endorsement of strong user authentication, to deliver an On-Line Digital Signature with full mobility that will make the digital signature affordable to all internet and intranet users.

2 About OTPK

2.1 Background of OTPK

The OTPK technology relies upon using a strong authentication infrastructure

(e.g. OTP token authentication) to provide the functionality of on-line digital signature.

As for today, the use of asymmetric cryptographic keys to carry out data security functions such as digital signatures is becoming prevalent. Many countries including USA, European countries, Australia, Japan, Hong Kong and Singapore have passed some forms of legislation to recognize PKI, thus, to legalize the use of digital signatures as equivalent to hand-written signatures in contracts, transactions, etc. The applications that can use digital signatures range from a paperless-office to high-value B-to-B transactions over the Internet to high-security health-care information systems.

In a PKI, all communicating entities¹ or users rely on a trusted body, typically a trusted-third-party, to perform the necessary validations on the identity. This trusted-third-party, known as the Certification Authority (CA), will issue (directly or indirectly via a Registration Authority) to each of the participating entities, a digital certificate containing information about the entity such as the name, organization, country, the policies governing the use of the certificate and, most importantly, the entity's public key. The certificate asserts that the entity is the rightful and sole owner (and user) of a secret private key with which the public key is associated. If there are asymmetric cryptographic operations, such as a digital signature which is carried out using the particular private key, it can only be carried out by this certified entity and easily verified using the entity's public key published in the digital certificate and up the certificate chain to the trusted key by CA.

Naturally, we can equate the integrity and non-repudiation of the

transaction to the security accorded to the protection of the entities' private keys. Each entity's private keys need sufficient protection to ensure that the keys are always in the possession and control of the rightful owners and cannot be stolen or duplicated. Smart cards (or USB tokens) are commonly used, as the medium, to protect private keys. But smartcards or USB tokens introduce additional problems of costs and logistics. The cost of a large PKI rollout using smart cards (taking into account of the cards, personalization, certification, etc) has been estimated at about US\$100 per user. This estimation is extremely prohibitive for a regular day-to-day PKI usage and probably has been one of the most widely quoted reasons for the apparent slow adoption rate of PKI in most of the countries worldwide. In contrast, the total cost for a recent Internet Banking Security 2-factor authentication implementation using Vasco OTP tokens that we deployed for 1 million users was less than US\$18 per user.

This paper, presents a revolutionary method to implement and deploy PKI, ensuring the same, if not higher, level of integrity and non-repudiation of the transactions, and yet not needing to incur the costs and logistics involved in deploying smart card solutions to apply to the digital signature law. We predict that this is the catalyst that the PKI boom has been waiting for.

2.2 The Concept of OTPK

The main concept behind OTPK is that the private key is a "One-Time Private Key" that works in connection with a short time certificate and is used for digital signature only to secure on-line transactions. As it is, OTPK cannot be used effectively for data encryption and user authentication.

There are essentially four steps that are carried out for each OTPK digital signature:

¹ An entity here is used loosely in this paper to represent a machine, a user, a group of users, an organization, a country, etc.

- i. Generate the asymmetric key
- ii. Send the public key for certification with the CA. At this step, OTPK relies on some form of authentication (strong 2-factor authentication is recommended) with the CA
- iii. Receive the certificate and sign the transaction
- iv. Delete the private key.

The validity of PKI certificate in this case need only be an extremely short term (in the order of minutes or seconds) to remove any chances of compromise. Since OTPK would result in a one-to-one mapping between the certificate and the transaction to be signed, details of the transaction can even be embedded in the certificate request for time stamping purposes.

In a typical PKI system, the user does a one-time generation and registration, and stores the certified key in a smartcard (or USB token) for a longer period of use. In contrast, the private key in the OTPK system is for one-time use only. A user always generates a new private key and authenticates securely with the CA in order to get a digital certificate for every transaction. Once the private key is used, it is expired and erased. There is no need to permanently store the private key in any media. Such a process sounds cumbersome; however, the overheads are actually not much more than any mobile credential solution.

The setup of OTPK requires the CA to have an online authentication and certification facility to fulfill all certification requests at a much higher throughput than existing setups of PKI. The entity could require a plug-in, implemented entirely in software to generate the private key, send the public key for certification, perform the digital signature operation, and delete the private key securely. The plug-in can be implemented as a PKCS#11, CAPI DLL or

even as a zero-install Java applet embedded within the web browser.

3 Related Work

There have been many attempts to address the cost and logistics problems, each with varying degrees of success. Among all the attempts, perhaps the most widely deployed solution is the Microsoft CSP (Cryptographic Service Provider) [1] that is installed with the Windows Operating System. The Microsoft CSP is implemented as a software-token that operates as if it is a smart card and would perform the cryptographic functions of digital signing, encryption, key storage, etc. Access to the CSP can be protected by a password. The obvious problem behind the Microsoft CSP and all software tokens *per se* is that the private keys are typically stored in local hard disk storage which opens the chance for hackers to make duplicated keys.

A HSM (Hardware Security Module) is a widely deployed solution, too. While HSMs have traditionally been used as a host-attached appliance to carry out cryptographic operations for the host, many commercial HSM vendors, such as Eracom [2], nCipher [3], SafeNet [4] and Thales [5] have implemented HSM devices that communicate via network and, hence, can support multiple client/user connectivity. The network-attached HSM could, thus, function as a pseudo smart card for each of the entities connected on the network and would be responsible for the cryptographic storage and operations. However, the legal definitions of PKI may be violated, as the private key would not be technically in the possession of the entity and the digital signature is carried out on behalf of the entity.

Another alternative that can be seen to address the problem is the Keon Web Passport solution [6] by RSA Security, Inc. Keon Web is a “virtual” smart card

implementation which relies on a backend server to secure and store private keys. When the entity requires the private keys, the keys can be downloaded securely to the entity's machine for usage, but it is still not foolproof. Backups in the backend server mean that multiple copies of the private keys exist. Moreover, the private keys are not always in the physical possession of the entity. This is a point of contention with some of the legal definitions [7, 8, 9] of a trusted and reliable PKI.

The Cosign [11] by Algorithmic Research Ltd is a very good example of a full digital signature solution. Cosign is mainly designed to support the enterprise that have a central user management like Microsoft Active Directory and strip most of the management issues. But again, the signing keys are stored in the server and the server signs on behalf of the user.

In the market, there are solutions that use on-line remote registration to acquire PKI credential including MyProxi, Kx509, Kerberized-CA and MyCA, to name a few. For authentication purposes, these solutions acquire long term or short term certificates and store them in either the server or client machine or other external storage device like smartcards. But they do not provide support for a short time certificate for digital signature.

4 OTPK vs PKI

The advantages of OTPK over the existing PKI systems include:

4.1 No Need for Smart Cards for Entities

In the OTPK system, since the entities' private keys are generated prior to a transaction and discarded after use, there is no need for traditional smart cards (or USB tokens) to store and protect the private keys.

This represents very significant savings in terms of costs, resources and time overheads in implementing and maintaining a PKI system.

4.2 Much Smaller Window of Compromise

In the OTPK system, the duration of validity of the private key and certificate is extremely short. Also, by tying the certificate request to the content of the transaction and by adding time stamp we reduce misuse of the signing key. Typically the private key is used to generate only one or a few digital signatures for its lifetime. Moreover, the private key is erased after use. The combination of short duration, the lack of substantial signature data and absence of any key storage makes the OTPK system more difficult to compromise.

4.3 No Need for Large LDAP Systems

In a typical PKI system, the CA, after issuing the user's certificate, would publish the certificate with a LDAP system. This is to allow other participating entities to retrieve the certificate for verification purposes. Such LDAP systems have to be able to handle large amounts of load in order to support the verification process. In the OTPK system, since each certificate has small and limited time validity, the use of the LDAP for storing and publishing the entities' certificates is not feasible. Instead, the OTPK protocol would require that the certificate be attached with the digital signature in the transaction, for validation purposes only.

4.4 No Need to Maintain CRL or OCSP for User Certificates

In a typical PKI system, a CRL (Certificate Revocation List) and/or OCSP (Online Certificate Status Protocol) mechanism has to be in place to maintain the up-to-date status of the certificates. If a user has lost the private key, the corresponding certificate should be revoked and listed in the CRL. By doing so, the corresponding entities do not rely on the certificate from that point onwards. However, the CRL and OCSP mechanisms add significant overheads to the entire PKI process. In the OTPK system, the CRL and OCSP are no longer relevant because the private keys and certificates have limited time exposure and would not be compromised.

4.5 Lower Learning Curve

One of the problems with existing PKI implementations is the need to educate and re-educate the users. Most users find PKI rather confusing with the need to understand how to use smart cards including installing smart card readers, entering pins, changing the pins on a regular basis, how to use certificates, and what to do when the certificates expire, etc. Educating users takes up significant time, costs and resources. For the OTPK system, all the confusing cryptographic technology and PKI protocols are abstracted from the users. Instead, the users will need to use a more familiar 2-factor OTP authentication to approve the transaction. The complexity of the certificates and digital signature is either made redundant by the OTPK design or handled easily by the client plugin's interaction with the CA.

4.6 Easy Interface into 2-Factor/Biometric and Other Authentication Solutions

In a typical PKI system, there exists two points of authentication. One is with the CA for issuing an initial certificate which is carried out once in a long time. The other is

with the media such as a smartcard that contains the private key. Authentication to the media is usually static PIN-based, as it is the media that enforces the authentication. If the protection of the media requires more complicated or stronger authentication, a lot of more complexity will have to be built into the media, resulting in higher costs. Moreover, not all media can support all forms of strong authentication.

In the OTPK system, only one point of authentication (with the CA) is needed. It is carried out when a private key needs to be used. Since the authentication can be centralized to a CA or a collection of CAs, there is economy of scale in implementing a strong authentication (such as 2-factor, biometric etc) to the CA and the cost can be shared across a large pool of entities. There is also no constraint on the media. This makes it easier to integrate a strong authentication mechanism into the OTPK system.

However, we do recognize that there are limitations to the implementation of the 2-factor or biometric authentication to the CA. For example, while OTP tokens are suitable for Internet-based transactions, remote authentication over Internet using biometrics is inherently insecure, and subject to replay attacks. On the other hand, using biometrics within a controlled office or a Kiosk environment for paperless e-Document systems is more convenient as compared to the OTP tokens. These considerations will have to be taken into account when designing the OTPK deployment.

We envision several scenarios where OTPK can be deployed:

- Internet Transactions.
A merchant operates an Internet trading portal which requires the user to digitally sign transactions to signify approval. Users will login to the portal using a browser. In such case, the

OTPK client is a Java-Applet that is dynamically downloaded within the browser and users can be issued an OTP hardware token (e.g. Vasco or RSA SecurID) to authenticate with an Internet-based online CA for OTPK certificates.

- Enterprise eDocument

A large enterprise operates an electronic document system to digitize the entire business workflow for processing efficiency and regulatory compliance. The application requires Microsoft Office and Adobe Acrobat documents to be digitally signed during the creation and approval process. For this scenario, the OTPK client can be in the form of a pre-installed CSP (Cryptographic Service Provider) DLL, and users can use UserID-Password, or Active-Directory authentication to authenticate to the enterprise OTPK CA for OTPK certificates. Biometrics, in the form of hand-written signatures, can also be used as the authentication means to the OTPK CA.

- Banking Kiosk

A bank operates an ATM (auto-teller machine) network and requires the use of digital signatures for high-value transfers. The ATM can be deployed with finger-vein or palm-vein biometrics to serve as a stronger form of authentication. In this case, the end-user can use biometrics to authenticate to the OTPK CA (via the bank's internal ATM network) to get the certificate.

- Mobile phone

A healthcare provider operates an e-prescription system that allows doctors to issue patient prescriptions electronically. All prescriptions need to be digitally signed. In this scenario, the doctor's mobile phone can be installed with an e-prescription application with OTPK capability. Doctors can be issued with a hardware OTP token. When

issuing a prescription, the doctor will enter the OTP from the token to the e-prescription application which will generate the one-time-use key and authenticate to the OTPK CA for the certificate before submitting the prescription and signature to the health-care e-prescription system.

4.7 Private Key Always in the Possession of Users

Many of the legislation regarding Digital Signatures and PKI explicitly require that the user's private keys be always in the possession and control of the user [7, 8, and 9]. Such requirements imply that some of the mobile credential solutions would not be recognized as compliant to the Act. The OTPK system relies on a client plug-in to generate and temporarily store the private key for the short duration that the Private Key is used. In the entire process, the private key remains in the possession and control of the user.

4.8 Protocol Is Interchangeable for All Asymmetric Algorithms

The OTPK system does not differentiate between different asymmetric algorithms and allows for entities using different asymmetric algorithms (e.g. RSA, DSA, ECDSA, etc) to participate within the same PKI. This means that one user can use RSA to perform digital signatures while another user can use ECDSA. Since the CA handles the certification collectively at the point of performing the digital signature, the OTPK solution is flexible enough to allow different entities using different algorithms to participate together. For example, the same user may use RSA in one country and ECDSA in another country depending on the electronic regulations and laws governing the countries.

Also, in the event that an algorithm is deemed undesirable, due to whatsoever reason such as cryptographically broken, insufficient key length, licensing, poor performance, platform constraints, etc, the user can easily use a different algorithm or key length without affecting all the other participating entities. The CA may also seamlessly migrate entities from using one algorithm to another without affecting the PKI or the PKI operations.

This flexibility allows different entities to use different applications. It also allows entities with certain platforms and restricted type of algorithms to participate in the system. Finally, it allows entities that are unable (or not allowed) to use certain types of algorithms or certain key lengths to participate in the PKI. Such flexibility is currently not practical within the existing PKI system.

4.9 Solution Is Very Scalable

Since most of the cryptographic load (i.e. Key generation, etc) is actually carried out at the user end, the load on the CA is only the cost of 1 asymmetric key signing operation per transaction. Each signing operation is also stateless, meaning that multiple CAs performing the OTPK certification need not synchronize the keys or certificates with each other.

From an operational perspective, the OTPK solution can be easily scaled up to handle larger volumes by adding more points of presence of the certification and authentication servers. The implementation can rely on a certification chain, leading up to the root CA, where sub-CAs that operate the certification and authentication servers can perform the user certification on behalf of the root CA. These sub CAs spread out the certification load and do not compromise the overall security of the OTPK solution.

4.10 Efficient and Effective Business and Pricing Model for CA

In the typical PKI, the CA charges on per-certificate basis. However, since the private key to the certificate can be used to sign many transactions, the CA charges a significant amount of money per certificate. Such a pricing model does not efficiently charge according to the actual usage since a user that uses the private key regularly versus another user that uses the private key rarely are charged the same amount.

In the OTPK system, since the certificates are issued each time a private key is used, the CA can charge a much smaller amount for each certification. Such a pricing model will mean that entities that use the private key more often will incur more charges, and vice versa. This results in a fairer and more acceptable pricing model. It also allows CAs to price the certificates and services differently for different applications such as (but not limited to) the following:

- Mode - online certification versus batch certification are priced differently
- Timing - certification requests during peak hours will incur higher charges
- Loyalty - the more certificates are requested, the cheaper the cost of each certificate
- Branding - different classes of certificate with different certification policy are priced differently
- Algorithm - certificates for different algorithms are priced differently.
- Insurance - price of certificate includes insurance on the transaction that is tied to the certificate
- Duration - one-time use versus per-session use certificates cost differently

A further advantage is that the OTPK certificates can integrate transparently with current PKIs. Relying party software that can process traditional PKI certificates can also process OTPK

certificates, barring a possible X.509 extension indicating that status information is not published. An existing CA can choose to issue both traditional PKI as well as OTPK certificates and allow both systems to interoperate, ensuring the maximum flexibility for the CA to adjust the business model.

5 Addressing OTPK Issues

While OTPK is able to solve some of the very key issues (e.g. logistics, costs, compliance to laws) plaguing traditional PKI setups, OTPK introduces a number of new issues that have to be addressed in order to make OTPK a viable digital signature solution.

5.1 Online CA key

The most distinct difference in the backend setup of a traditional PKI CA versus an OTPK CA is the use of the CA Key, or the key that is used to certify the certificates.

For the OTPK CA, the CA Key has to be accessible online as the certification requests are expected to be fulfilled in real-time. The entire certification process is expected to be carried out within a couple of seconds to ensure that the transaction approval process is not delayed. In contrast, the CA Key in traditional PKIs need not be online as the certification process may take up to 48 hours to allow for manual processes to be carried out. The concern here is if the security of the CA Key is compromised in any way by making the key online, versus using some physical means to ensure that the key is not accessible on the Internet.

We argue here that while the concerns, on the surface, seem to point to a more vulnerable CA, having an online CA Key does not lower the security of the PKI setup. This is because:

- Stolen CA Key
The use of a high-level FIPS-certified HSM (at least Level 3) will mitigate this risk by making it impractical to extract the private key.
- Fake certificate requests
All OTPK certificate requests come embedded with the authentication credentials of the user. The authentication credentials can be in the form of a one-time-password from the user's token. This allows the CA to verify the user before issuing the certificate.

The process of verifying the authentication credentials + certifying the key should be done in one atomic step within the HSM to ensure that a compromised system is unable to illegally send certification requests to the HSM. By insisting on the use of strong authentication + HSM with OTPK, we are able to mitigate this exposure.

5.2 User Registration

Another difference is in the user registration process. In the traditional PKI CA, the user registers with the CA once to generate the user's private key, and get the public key certified by the CA. During the registration process, one key step is that the CA would verify the credentials of the user. Once done, the user is free to use the private key without needing to contact the CA.

For OTPK, this registration process seems to be missing while the certification process is repeated each time the user needs to sign a document or transaction. However, we need to clarify here that the registration process did happen. If we are to extrapolate backwards to the point in time when the user was first assigned the authentication token (assuming a one-time-password token), this

was when the registration process actually took place. As for the repeated certification processes, it is simply equivalent to the user authenticating to the CA and obtaining services from the CA.

5.3 Secure Time-stamping

Current time-stamping (or electronic notary) solutions rely on a central time-stamping server that essentially signs on the hash of the transaction and include a time-stamp with the transaction [15]. This is an issue that is relevant even for traditional PKI digital signature implementations which typically rely on the user's PC date/time for the time stamp. How can we prove that the user signed the transaction at a particular time?

For OTPK, the solution is rather apparent. This can be done by simply allowing the CA to also function as the secure time-stamping service. When the user generates the certificate request, the hash of the transaction to be signed is also embedded as part of the certificate request, along with the authentication credentials. This allows the CA to issue the short-lived OTPK certificate of the user key, and with a reliable time-stamp on the hash value (e.g. as one of the X.509 extensions) back to the user. This certificate can thus be used as the proof for the time-stamp.

5.4 Secure Private Key deletion

The issue for private key deletion comes in when the certificate has expired, and we do not want the private key to be used for any other purposes (since it is no longer valid). For traditional PKI setups where the private key is securely stored in a smartcard or USB token, the destruction of the private key is more visible, and since certificate expiries do not happen as often (typically only once a year or once in 3 years) as OTPK certificates, the

requirements for key deletion is not as pronounced.

For OTPK certificates, a new key is used for each digital signature which may result in hundreds (or even thousands) of keys used by a user in a year. This gives a hacker potentially more chances to obtain a user's private key, albeit with an expired certificate.

For OTPK, we are able to address the problem in two ways: directly and indirectly. In the direct approach, we have to ensure that the private key is deleted as designed. This can be achieved by implementing the key deletion process in the OTPK client as part of the atomic function of the signing process (i.e. Generate Key-Get certificate-Sign-Delete Key), and ensuring that this process cannot be modified through secure programming techniques as well as sending the OTPK client for FIPS-140 certification. We recognize that this method is not foolproof standalone and is vulnerable to a crafty signer who fully intends to cheat the system.

In the indirect approach, we have to make sure that the private key cannot be used for any other transaction. This can be implemented similarly to the secure time-stamping in Section 5.3. Since the certificate and key is directly tagged with the transaction, using the private key to sign a different transaction would result in a signature validation failure.

6 Conclusion

The OTPK technology is bringing up a new concept in which a user will generate a signing key with an extremely short lived certificate to perform the digital signature. The PCT (Patent Cooperation Treaty) has defined the OTPK as '*novel and innovative*' [14]. The key of the innovativeness is that the OTPK technology allows an implementation of on-line digital

signature system that complies to the digital signature law with full mobility and low cost of ownership. The entity is generating the signing key and owns it during the whole process of “Key Generation” “Certification” “Signing” and after signing deleting the Signing key. It could be regarded as a new paradigm in the “PKI” technology that allows the population of the digital signature to many vertical markets.

To put things in perspective, we have benchmarked a Java applet OTPK implementation which uses RSA-1024 keys on an IE browser. The time taken for the key generation + certificate request + digital signing takes less than 7 seconds on a Pentium 3 machine and less than 2 seconds on a Pentium 4 machine. For the mobile phone, a J2ME OTPK application using ECDSA-P192 averages between 3 to 10 seconds on various mobile phones.

DSSS is currently implementing OTPK protocol and proof of concept into the DSSS Authentication Server for demo purpose only. It is planned to be further enhanced with XKMS and WS-Security. (The OTPK is patent-pending USPTO 60/590,348)

References

- [1] Microsoft Platform SDK: Cryptography. http://msdn.microsoft.com/library/en-us/seccrypto/security/microsoft_cryptographic_service_providers.asp
- [2] Eracom Technologies. <http://www.eracom-tech.com>
- [3] nCipher Corporation Ltd. <http://www.ncipher.com>
- [4] SafeNet Inc. <http://www.safenet-inc.com>
- [5] Thales e-Security. <http://www.thales-ecurity.com>
- [6] RSA Keon Web PassPort. <http://www.rsasecurity.com/node.asp?id=1230>
- [7] Electronic Transaction Act 1998, Singapore, Part IX – Duties of Subscribers, Clause 39. <http://www.ida.gov.sg/idaweb/pnr/info page.jsp?infopagecategory=regulation:pnr&infopageid=11944&versionid=1>
- [8] Utah Digital Signature Act, Part 3. Duties of certification authority and subscriber, Utah Code 46-3-303. <http://www.jus.unitn.it/users/pascuzzi/privcomp97-98/documento/firma/utah/udsa-3.html>
- [9] Georgia Digital Signature Act. Section 1, Article 3, Clause 10-12-33, Subscriber’s Warranties
- [10] Peter Gutmann, “Plug-and-Play PKI: A PKI your Mother can Use”, Usenix Security Symposium, 2003 <http://www.cs.auckland.ac.nz/~pgut001/pubs/usenix03.pdf>
- [11] Algorithmic Research Cosign – www.arx.com
- [12] AlphaTrust PRONTO™ Server- <http://www.alphatrust.com/>
- [13] Deepnet Technology Smart ID VSC <http://www.deepnettechnologies.com/>
- [14] PCT Review, PCT/SG2005/000226
- [15] RSA Security, “What is digital timestamping?”. <http://www.rsasecurity.com/rsalabs/node.asp?id=2347>