PKI Technology: A Government Experience

Dr. Ali M. Al-Khouri

Emirates Identity Authority, Abu Dhabi, United Arab Emirates

Abstract

As government operations are moved online, information technology security services based on cryptography become essential. Public key cryptography can play an important role in providing enhanced security services related to data protection and strong credentials for identity management. This article attempts to contribute to the limited domain of knowledge available about government practices and projects. The purpose of this article is to provide an overview of the public key infrastructure (PKI) components deployed in the United Arab Emirates national identity system. It provides a comprehensive overview of PKI technology and its primary components. It then provides an overview of the existing cryptographic components and the digital certificates stored in the PKI Applet of the smart ID card, with the purpose of shedding light on what is needed to fulfill the needs for future e-government requirements in the country.

Keywords: UAE PKI, e-government, e-commerce, digital signature, encryption.

1. Introduction

Many governments in the past decade have initiated advanced identity management systems that incorporated PKI technology. This global interest in the technology is based on the need to meet the requirements for higher levels of authentication, confidentiality, access control, non-repudiation, and data integrity. Perceptibly, governments have been under tremendous pressure to deliver internet-based electronic services in light of increasing citizens' demands for improved and more convenient interaction with their governments.

Many researchers argue that PKI is a key pillar for e-government transformation and e-commerce enablement. Although the concept of PKI may sound simple, many deployment experiences have shown catastrophic results from both technical and operational standpoints (GAO, 2001; Judge, 2002; Pluswich and Hartman, 2001; Rothke, 2001; Schwemmer, 2001). The major deficiency in the existing literature is related to the lack of reported experiences from governments' projects. There are still ruthless attempts by government-backed projects to push the introduction of PKI in identity systems. This outlines the need for sharing knowledge of implementation experiences from various government projects. This is article is written with this scope of need.

The government of the United Arab Emirates initiated its major PKI initiative as part of its national identity management infrastructure development program in 2003. This project is considered to be one of the early systems in the Middle East region, and with the objective to issue 10 million digital identities by the year 2013. This article discusses key components

of the PKI project related to private key activation, certificate validation, and encryption, in the context of e-government applications. This article is primarily structured into two sections. The first section provides an overview of PKI technology, describing its key components and how it provides security. The second section discusses the cryptographic components of PKI in the UAE project in light of e-government and e-commerce future requirements.

2. Public Key Cryptography

Cryptography is the branch of applied mathematics concerned with protecting information. Confidentiality is the protection of data against unauthorized access or disclosure through application of functions that transform messages into seemingly unintelligible forms and back again. These processes are called encryption and decryption. One kind of cryptography that can provide confidentiality, authentication, and integrity is symmetric key cryptography, in which an algorithm makes use of a single key used to encrypt data. The same key is also used to decrypt or return the encrypted data into its original form. This one key, called the symmetric key, is very efficient in terms of processing speed and using minimal computing resources, but is limited in the sense that (1) it is difficult to exchange the key securely without introducing public key cryptography, and (2) because both the sender and the receiver of a message share the same symmetric key, the authentication and integrity is not provable to a third party who does not also hold the key—thus, symmetric cryptography cannot provide the additional security service called non-repudiation.

Public key cryptography is an attempt to solve these particular shortcomings of symmetric key cryptography (Ferguson et al., 2010). Public key cryptography employs an algorithm using two different but mathematically related keys, one for creating a digital signature or decrypting data, and another key for verifying a digital signature or encrypting data. Computer equipment and software utilizing such key pairs are often collectively termed an asymmetric cryptosystem. The complementary keys of an asymmetric cryptosystem for PKI technology are arbitrarily termed the private key, which is known only to the holder, and the public key, which is more widely known. If many people need the public key for various PKI applications, the public key must be available or distributed to all of them, perhaps by publication in an online repository or directory where it is easily accessible. Although the keys of the pair are mathematically related, if the asymmetric cryptosystem has been designed and implemented securely it is computationally infeasible to derive the private key from knowledge of the public key. Thus, although many people may know the public key of a given holder, they cannot discover that holder's private key. This is sometimes referred to as the principle of irreversibility.

Another fundamental process, termed a hash function, is used in PKI technologies. A hash function is an algorithm that creates from a message a digital representation or fingerprint in the form of a hash value or hash result of a fixed length (Spillman, 2005). The hash result is usually much smaller than the message, but nevertheless substantially unique to it. Any change to the message produces a different hash result when the same hash function is used; the hash is unique to a given message for all practical purposes. In the case of a secure

hash function, sometimes termed a one-way hash function, it is computationally infeasible to derive the original message from knowledge of its hash value. Hash functions therefore enable the PKI application software to operate on smaller and more predictable amounts of data, while still providing robust correlation to the original message content.

Table: Mapping of Security Services to Cryptographic Techniques

Cryptography Techniques/ Security Services	Encryption/Decryption	Message Authentication Codes/Keyed Hash	Digital Signature Generation/Verification
Confidentiality	Symmetric or Asymmetric	-	-
Authentication	-	Symmetric or Asymmetric	Asymmetric only
Integrity	-	Symmetric or Asymmetric	Asymmetric only
Non-Repudiation	-	-	Asymmetric only

2.1 Digital Signatures

Digital signatures are created and verified by public key cryptography. The signer has a key pair consisting of a private key and a public key. The signer holds a private key known only to the signer, which the signer uses to create the digital signature. The signer also has a public key, which is used by a relying party to verify the digital signature. Relying parties must obtain the signer's public key in order to verify the signer's digital signature. As applied here, the principle of irreversibility means that it is computationally infeasible to discover the signer's private key from knowledge of the public key and use it to forge digital signatures. The digital signature cannot be forged unless the signer loses control of the private key by divulging it or losing the media or device (smart card) in which it is contained, or an attacker is, through the application of massive computing resources-performing cryptographic analysis, able to derive the private key from the public key.

This impossibility for retrieval of the input message is pretty logical if we take into account that a message's hash value could have a hundred times smaller size than the input message. Actually, the computing resources needed to find a message by its digest are so huge that, practically, it is infeasible to do it. It is also interesting to know that, theoretically, it is possible for two entirely different messages to have the same hash value calculated by some hashing algorithm, but the probability for this to happen is so small that in practice it is ignored (see also Stallings, 2006). From a technical point of view, the digital signing of a message is performed in two steps, and as depicted in Figure 1.

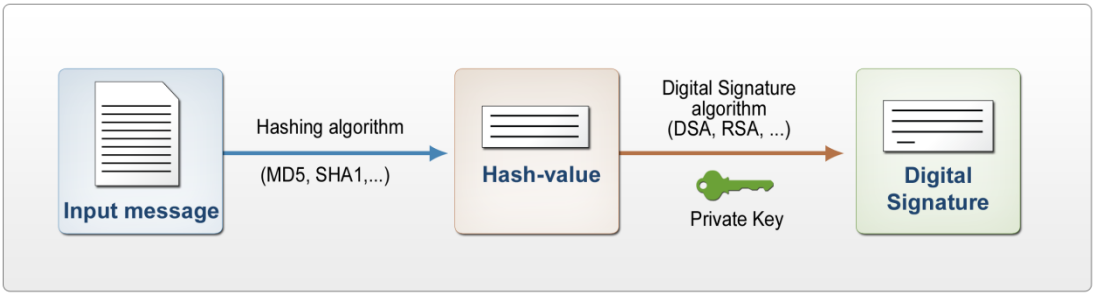

Figure 1: Digital Signing Process

2.2.1 Calculating the Message Digest

In the first step of the process, a hash value of the message (often called the message digest) is calculated by applying some cryptographic hashing algorithm (e.g., MD2, MD4, MD5, SHA1, or other). The calculated hash value of a message is a sequence of bits, usually with a fixed length, extracted in some manner from the message. All reliable algorithms for message digest calculation apply mathematical transformations such that when just a single bit from the input message is changed, a completely different digest is obtained.

2.2.2 Calculating the Digital Signature

In the second step of digitally signing a message, the information obtained in the message's first-step hash value (the message digest) is encrypted with the private key of the person who signs the message and thus an encrypted hash value, also called digital signature, is obtained. The most often used algorithms are RSA (based on the number theory), DSA (based on the theory of the discrete logarithms), and ECDSA (based on the elliptic curves theory). Typically, a digital signature (the transformed hash result of the message) is attached to its message and stored or transmitted with its message. It may also be sent or stored as a separate data element, so long as it maintains a reliable association with its message.

2.2.3 Verifying Digital Signatures

Digital signature technology allows the recipient of given signed message to verify its real origin and its integrity. The process of digital signature verification is designed to ascertain if a given message has been signed by the private key that corresponds to a given public key. The digital signature verification cannot ascertain whether the given message has been signed by a given person. If we need to check whether some person has signed a given message, we need to obtain his real public key in some manner. This is possible either by getting the public key in a secure way (e.g., on a floppy disk or CD) or with the help of the public key infrastructure by means of a digital certificate. Without having a secure way to obtain the real public key of given person, we are not able to check whether the given message is really signed by this person. From a technical point of view, the verification of a digital signature is performed in three steps as depicted in Figure 2.

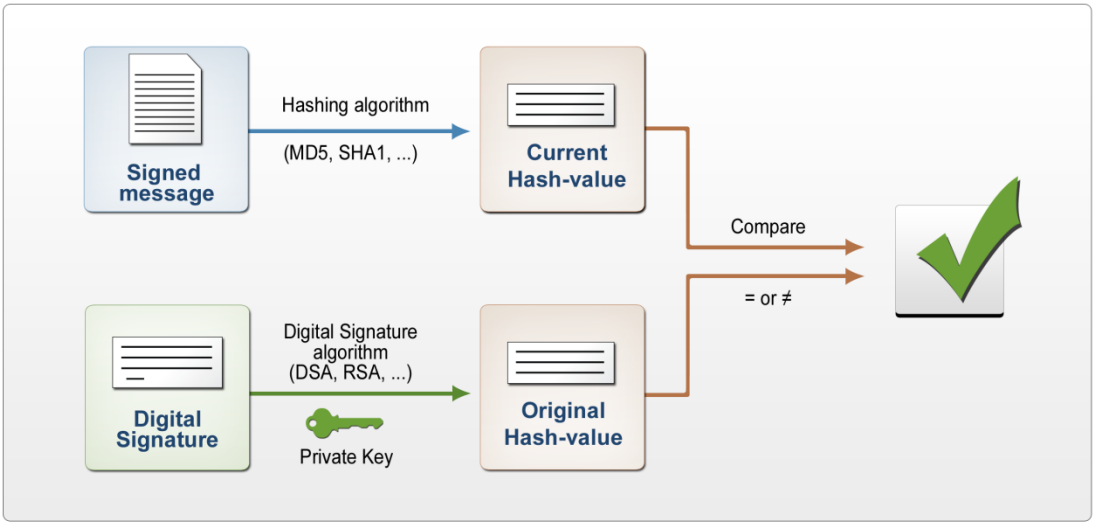

Figure 2: Digital signature verification process

Step 1: Calculate the Current Hash Value

In the first step, a hash value of the signed message is calculated. For this calculation, the same hashing algorithm is used as was used during the signing process. The obtained hash value is called the current hash value because it is calculated from the current state of the message.

Step 2: Calculate the Original Hash Value

In the second step of the digital signature verification process, the digital signature is decrypted with the same encryption algorithm that was used during the signing process. The decryption is done by the public key that corresponds to the private key used during the signing of the message. As a result, we obtain the original hash value that was calculated from the original message during the first step of the signing process (the original message digests).

Step 3: Compare the Current and the Original Hash Values

In the third step, we compare the current hash value obtained in the first step with the original hash value obtained in the second step. If the two values are identical, the verification is successful and proves that the message has been signed with the private key that corresponds to the public key used in the verification process. If the two values differ from one another, this means that the digital signature is invalid and the verification is unsuccessful.

2.2.4 Reasons for Invalid Signatures

There are three possible reasons for getting an invalid digital signature:

- If the digital signature is adulterated (it is not real) and is decrypted with the public key, the obtained original value will not be the original hash value of the original message but some other value.

- If the message was changed (adulterated) after its signing, the current hash value calculated from this adulterated message will differ from the original hash value because the two different messages correspond to different hash values.
- If the public key does not correspond to the private key used for signing, the original hash value obtained by decrypting the signature with an incorrect key will not be the correct one.

If the verification fails, in spite of the cause, this proves only one thing: The signature that is being verified was not obtained by signing the message that is being verified with the private key that corresponds to the public key used for the verification. Sometimes, verification could fail because an invalid public key is used. Such a situation could be obtained when the message is not sent by the person who was expected to send it or when the signature verification system has an incorrect public key for this person. It is even possible that one person owned several different valid public keys with valid certificates for each of them and the system attempted to verify a message received from this person with some of these public keys but not with the correct one (the key corresponding to the private key used for signing the message).

In order for such problems to be avoided, most often when a signed document is sent, the certificate of the signer is also sent along with this document and the corresponding digital signature. Thus, during the verification, the public key contained in the received certificate is used for signature verification; if the verification is successful, it is considered that the document is signed by the person who owns the certificate.

2.2 Digital Certificates

The description of the use of digital signatures above leaves open one security question that must be resolved in an infrastructure for secure electronic commerce: How can the verifier obtain the alleged signer's public key in a way that ensures that the public key is, in fact, that of the signer? Some mechanism is necessary to avoid the scenario of an attacker intercepting the message, rewrapping the plaintext of the message with his own digital signature, and giving the verifier his own public key. The attacker could pass off his own public key as if it were the public key of the intended signer. The verifier, using the attacker's public key, will find that the public key is able to process the digital signature on the message he received. Moreover, the verifier will think that the message originated with the signer, not the attacker. The verifier needs a mechanism to obtain the public key of the signer in a reliable way to avoid this kind of substitution.

Within a PKI, the method for preventing this kind of substitution attack is the digital certificate (Barr, 2002). A certificate is a message stating that a public key belongs to or is associated with a given individual, organization, or device. The party issuing the certificate is a certification authority, or "CA," and the party receiving it is called the subscriber. A digital certificate is itself a digitally signed message. The issuing CA signs the message with its private key. The digital signature on the certificate itself provides assurances of the origin of the CA signing it, and the fact that the certificate has not been tampered with since issuance. Thus, the certificate is the CA's signed assertion that a particular public key belongs to a specific individual, organization, or device.

To the extent the relying party trusts the CA, the relying party can trust in this binding and use the public key in the certificate with confidence to verify digital signatures of the subscriber. Of course, if the certificate is a digitally signed message binding a subscriber to a public key, it is also necessary to obtain the CA's public key, or root certificate, to verify the digital signature on the certificate.

If the verifiers that need the root certificate are small in number, it is possible to distribute the root in person. Root certificates may also be distributed on media using trustworthy non-Internet delivery mechanisms, such as reputable courier services or even postal mail. While this option may be satisfactory for small communities, it is difficult to scale this solution to large populations. As a result, many CAs have arranged with software manufacturers to embed their roots within the software itself. Under this solution, when a verifier needs to refer to a root certificate, the root certificate is already within the verifier's software and is available for use. To date, this solution has proved to be the most effective method of distributing roots widely.

2.3 Data Encryption

In addition to digital signatures, public key technology may be used to encrypt messages in order to protect the confidentiality of the information contained within them. In the encryption process, the sender of the data to be kept confidential uses the recipient's public key to encrypt the data. The recipient uses the recipient's private key to decrypt the data. The principle of irreversibility here means that it is computationally infeasible for anyone intercepting the message and having knowledge of the recipient's public key to derive the private key and decrypt the data. Moreover, only the recipient, who holds that private key, will have the ability to decrypt the data.

Widely deployed encryption software, such as e-mail clients, can perform these encryption functions. This software, however, does not use the asymmetric key to encrypt the entire plaintext of the message. Asymmetric key operations tend to be costly in terms of time and computing power. Therefore, software commonly uses a symmetric key used only for this one operation (called a "session key") to encrypt the plaintext message and then, in turn, uses the recipient's public key to encrypt the symmetric session key. The message sent to the recipient includes the encrypted message and the encrypted session key. The recipient then uses the recipient's private key to decrypt and recover the session key. The session key is then used to decrypt the message itself. As with digital signatures, a sender of a confidential message can obtain the public key of the recipient using the recipient's certificate.

2.4 Secure Sockets Layer (SSL)

One of the best-known uses of public key technology is the protocol known as the Secure Sockets Layer (SSL), which protects the communications between a browser on a client machine and a server over an insecure network, such as the Internet. People every day access e-commerce sites to purchase goods and services over the Internet, and wish to

secure their sessions with these sites to protect the confidentiality of information such as credit card numbers. The magnitude of this everyday use of SSL to protect these sites indicates that SSL is by far the most widespread commercially deployed PKI technology. An SSL session consists of the following procedures:

- A browser sends a request to connect to a site that has a server certificate. The user performs this request by clicking on a link indicating that it leads to a secure site or the user types in a URL with an "https" protocol specifier.
- The server responds and provides the browser with the server's certificate.
- The browser verifies the digital signatures on the server certificate with reference to a certificate chain leading to a trusted root certificate.
- The browser also compares the server's domain with the domain listed in the certificate to ensure that they match. If these steps are successful, the server has been authenticated to the user, providing assurances to the user that the user is accessing a real site whose identity was validated by a CA. This process is called server authentication.
- Optionally, the server may request the user's certificate. The server can use the user's certificate to identify the user, a process called client authentication.
- The browser generates a symmetric session key for use by the browser and server in encrypting communications between the two.
- The browser encrypts the session key with the server's public key obtained from the server certificate and sends the encrypted key to the server.
- The server decrypts the session key using its private key.
- The browser and server use the session key to encrypt all subsequent communications.

Following these procedures, the user may notice a padlock symbol appearing on the screen. In addition, the user will be able to inspect the certificate on the site using the browser.

2.5 Biometrics

Biometrics is a term referring to the measurement of one or more biological characteristics of an individual, such as fingerprints, voice recognition, eye imaging, hand geometry, and the like. Primarily a form of identification and authentication, biometrics can enhance PKI and can be enhanced by PKI.

- A biometric can augment or replace the access control placed over a subscriber's private key.
- The integrity and authenticity of the biometric template can be ensured via digital signature and can even be enveloped within a digital certificate.
- The biometric reader device can be authenticated via PKI (similar to existing mechanisms used for point of sale (POS) and automated teller machines (ATM).

2.6 Key Management

Because cryptographic keys are very special pieces of data that require extraordinary handling, the subject warrants particular attention. Symmetric and asymmetric algorithms

and their cryptographic keys all have different strengths, weaknesses, and properties that require distinct policy and practices to protect them.

The controls over the asymmetric private and public keys inherent in a properly deployed PKI ensure its reliability. For the public key, a digital certificate ensures the integrity and authentication of the subscriber's public key and provides the cryptographic binding between the subscriber's identity (and/or other attributes) and public key. Key recovery is the ability to reconstitute a decryption key for the purposes of recovering encrypted data. This may be necessary in the event of a hardware failure, where the key has been lost, the untimely death of a person when the PIN guarding access to the key is no longer available, or other circumstances where encrypted data must be recovered.

In order to provide key recovery services, the PKI service provider may store activation data or the decryption key itself. The design and implementation of a storage and retrieval process will usually be specific to the PKI service provider and may involve a combination of chain of custody, dual control, split knowledge, encryption, and other techniques by the parties involved to provide procedural protections for the private key.

Private key recovery presents the security risks of unauthorized access to the private key, which can be used to decrypt sensitive information. In the case of single key pair schemes, in which one key services both signature creation and decryption purposes, there may be reasons for escrowing or managing the single key. When such systems are used, unauthorized access to a private key also entails the risk that an attacker could create digital signatures using the recovered key and thereby impersonate the subscriber. Consequently, there is a business need to limit the circumstances under which a private key can be recovered and also control access to the private keys to prevent unauthorized private key recoveries. Circumstances under which recovery is appropriate or required generally fall into two categories: voluntary requests from the subscriber and requests for a subscriber's private key that originate from another responsible and authorized party, which are likely to be involuntary from the perspective of the subscriber.

2.7 Certificate Revocation List (CRL)

A CRL is a digitally signed list of revoked certificates' serial numbers that is generally issued by the CA that issued the (revoked) certificate. CRLs provide information regarding a certificate's status. CRLs are issued periodically and downloaded to relying party systems on a scheduled basis (e.g., every twenty-four hours. CRLs contain an issue date as well as the date that the next CRL should be issued.

The frequency of CRL issuance tends to reflect the risks and assurances associated with the certificates. In some cases, unscheduled "interim" or "delta" CRLs may be issued, particularly in the event of key compromises. CRLs have other advantages, and they have disadvantages as well. In addition to the short response time that a local CRL provides, a CRL may be a cost-effective means to validate certificates in low-value transactions in which the infrequent revocation of a certificate keeps the CRL relatively small. In such situations, the relying party's system can be designed to check for and pull down updated CRLs as often as convenience and risk management dictates. However, a CRL may only be considered valid at

the time it is published. As the size of the CRL and the value of the underlying transaction grow, the CRL becomes a less cost-effective solution.

The solution chosen in some situations might include a combination of both CRL and status checking (for example, Online Certificate Status Protocol). Online mechanisms are capable of communicating the current (real-time or near real-time) status of a certificate. These mechanisms eliminate latency issues affecting CRLs, although they may introduce other risks (certificate status responder and Internet connection downtime). The predominant online revocation/status-checking mechanism is the IETF Online Certificate Status Protocol (OCSP). OCSP provides a standardized protocol for online status requests for specific certificates. Upon request, an OCSP "responder" provides a signed status response message that reflects the current status of the certificate. The responder's signature can be verified by the relying party.

The timeliness of any certificate status information depends on the implementation. Some OCSP responders are merely front-ends for CRL-based revocation systems or base their response on the most current operational records of the CA. In these cases an OCSP response will not contain more current information than the CRLs. Relying parties may need to retain OCSP responses used to verify signatures, since each response is unique to a particular transaction. OCSP is only one of many types of online checking mechanisms.

3. UAE PKI System

During the early phases of the project, the PKI Applet on the card was a contentious issue. Although the purpose of the PKI applet was understood and the need for e-services realized, the application and services associated with it were only broadly understood at the time. It was decided to have a container that would have three key pairs, one for logical access or authentication, one for digital signing, and a third reserved for possible future use for data encryption and decryption. The container was designed to be personalized with the rest of the card and protected with a user PIN. The validity of the digital certificates (keys) in the card has the same lifetime as the card, which could be a maximum of up to five years. A new set of keys would need to be reissued with a new card and the expired public key certificates published in a revocation list.

From a system perspective, no copy is kept of the keys that are generated for an ID card. The keys are generated by the HSM and securely exported and loaded onto the card, after which the keys are deleted from the system. No key backup or archival (except for the CA root key) is done. Due to the fact that no data encryption and decryption is done, there was no need for key recovery at that stage. However, if a third certificate is implemented for data encryption and decryption, the need for key recovery has to be investigated. Not only is private key recovery important for data decryption purposes, but also public key archival might be needed for digital signature certificates. For example, when a legal document is signed and has a lifespan that exceeds the validity period of the card and therefore the certificates, the signature can no longer be verified unless the public key of the original digital signature is archived with the document or available from the CA authority for verification.

It is clear that the use and application of the PKI component of the ID card needs to be well defined and communicated in the Certificate Practice Statement. All partners and role players will also play a significant role in determining the scope and parameters of PKI use of the ID card. Finally, federal and international law will dictate certain aspects of PKI usage and aspects like key recovery, usage, and conditions for non-repudiation will determine implementation requirements. Taking all this into account, the current PKI is only in its beginning phase. In order to fulfill the usage requirements of the PKI component on the ID card, it will be necessary to plan a proper and well-designed architecture for the PKI needs in context of other government departments and GCC1 countries. The following subsections elaborate on the primary components of the UAE PKI project.

3.1 ID Card PKI Applet

The PKI applet provides the digital signature and authentication features in the form of digital certificates. The applet can accommodate three separated key sets and their associated digital certificates, as well as a user PIN code to prevent unauthorized use of these functions. One certificate is for authentication, one is for the digital signature, and one free empty key container is for future use. With the installed certificates and keys the user can be authenticated and can digitally sign e-commerce and e-government applications.

The certificates are in accordance with the X.509 standard. Together with the PKCS#11 cryptographic library and a CSP (cryptographic service provider for Microsoft Cryptographic API) the user uses Web Browsers (SSL v3 sessions), e-mail (S/MIME), VPN, and other PKI applications securely. For the hashing function SHA-1 is used and for asymmetric cryptographic functions the RSA algorithm is used.

During card personalization, the digital signature and the authentication certificates are loaded onto the ID card chip. The keys on the ID card are able to perform cryptographic functions but the decryption function is blocked. The blocking is necessary so that a key cannot be misused for a decryption function. For additional encryption and decryption and/or digital signature keys, the additional container can be used, but the keys are not loaded during the personalization. If the keys for encryption/decryption are loaded, a mechanism of key recovery must be implemented to guarantee that the private key can be restored in case of a lost or damaged card. Otherwise, the user would not be able to decrypt any data that has been encrypted previously.

The private keys for the digital signature and authentication functions are not held in the system, but are loaded once into the smartcard. It cannot be offloaded later on and will be used by the smartcard to perform cryptographic functions. If a card has been damaged, lost, or renewed, the user should be issued with a new card with new certificates.

3.2 UAE PKI Infrastructure

The UAE system employs multiple Certificate Authorities (CAs) to deal with key requirements for the population, time, and technical aspects of the system. For the purpose

[1] GCC is the acronym for Gulf Cooperation Council, also referred to as the Cooperation Council for the Arab States of the Gulf (CCASG). It includes six countries: Bahrain, Kuwait, Oman, Qatar, Saudi Arabia, and the United Arab Emirates. The number of GCC population is estimated to be around 40 million people.

of this discussion we are concerned with the population CA, which issues and sign keys and certificates for use in the ID card. The next subsections highlight possible caveats of the PKI infrastructure.

3.2.1 Activating the Private Key

The private key is the most important piece of data that needs to be protected. For the CA, the private key security is so important that it is physically stored on a hardware cryptographic module and protected by split authentication and various other physical access controls. Compromise of this key invalidates all the issued certificates by the CA and thus any digital transaction performed with the keys and certificates in question. The CA private key is stored in a hardware security module and protected adequately through various means.

The ID cardholder's private key is generated in the HSM and transferred to the ID card during personalization. The private key is activated through the use of a user PIN. This means that the most important piece of data, as far as an ID cardholder is concerned, is currently only protected by a minimum four-digit numeric password, better known as a PIN. If the ID card is stolen and the PIN very easily stolen through mechanisms like social engineering or shoulder surfing, the private key is compromised.

A more secure way to protect a private key is to activate it using a biometric instead of a PIN—in some cases using both. Match-on-card technology is a great way to authenticate a cardholder using his smartcard without having to have a complex online system available. Because the cardholder can be uniquely authenticated using his biometric, it serves as the best way to unlock protected information on his/her card (like the private key). In the UAE system, the match-on-card functionality is separate and serves no access control purpose, only cardholder authentication.

There is always a balance between security, ease-of-use, and functionality in any system—increase one and the other two will decrease. By using the biometric to activate the private key will provide the ultimate security but will limit the use of the PKI services to match-on-card-only applications and readers. The decision to use the biometric as access control mechanism for the PKI applet instead of a PIN can only be made once the nature of e-services is better defined and the requirements from partners and role players better understood.

3.2.2 Verifying Digital Signatures

As already explained, digital signatures are verified with a signer's public key and the public key certificate is verified with the CA's root certificate. Furthermore, the validity of both the public key certificate and the CA root certificate is verified against the CRL. Off hand, there is some serious infrastructure needed to fulfill the requirements just mentioned.

First of all, public key certificates must be distributed to all partaking entities to verify digital signatures. Depending on the application, the public keys might be queried from a central repository or embedded as part of the signed transaction and thus verified without any additional infrastructure. This is the preferred way but is not always supported by the application software of a specific digital signature application. The problem with the central

repository for public keys is not only the fact that it requires infrastructure but also heavy maintenance: Once cards get renewed there will be more than one version of a public key for a particular cardholder; transactions signed must be verified with the correct version of the public key; obsolete public keys must be removed, and so on.

Secondly, the public key certificate of a cardholder must be verified against the CA root certificate. This is to ensure that the cardholder is whom he says he is and his certificate has indeed been issued and signed by the Identity Authority. In a commercial scenario the root certificates of the most popular CAs are embedded in software like browsers, but for standalone CAs the root certificate has to be made available to any entity that requires validation. The pitfall is that the certificate itself does not need to be secure (it has already been signed with the CA's private key) but the mechanism in obtaining It has to be secure. This is because a fraudulent certificate, checked against the correct CA but fraudulent one, will validate correctly.

Many organizations embed their root certificates as part of the PKI-enabled applications. Although this mostly solves the problem, it is difficult to keep up to date as to the correct version of the root certificate (if it was renewed or compromised). Other organizations prefer to publish the certificate on their corporate websites. This might sound like the best and simplest idea, but it is not secure and an entity may never be sure whether it is referring to the correct site and correct certificate. The alternative is to install the root certificate on the card, together with the cardholder's certificates. This way, the root certificate is always accessible and up to date because when the card gets replaced so does the root certificate. It travels with the cardholder and can always be verified without any external checking.

Thirdly, both the public key certificate and the CA root certificate have to be validated against the CRL. The CRL simply keeps a list of certificate serial numbers and whether they are valid or not. Currently the CRL is published and stored as a single file in the Demilitarized Zone (DMZ).[2] However, no infrastructure currently exists to access and check against this file from an external entity. The CRL will be explained later in more detail.

3.2.3 Verifying Access Control Certificate

The second certificate in the card is used for authentication, or more technically, logical access control. This includes things like SSL client authentication, for example. Because logical access is one-time only or while a session exists, it is considered temporally and therefore does not have the certificate lifetime and archival issues associated with digital signature certificates and data encryption certificates. Instead, the only issues are validating the CA root certificate and CRL-related issues, as explained in the previous section with digital signature validation.

3.2.4 Certificate Validation

[2] The Demilitarized Zone (DMZ) is a physical or logical sub-network that contains and exposes an organization's external services to a larger un-trusted network, usually the Internet. The purpose of a DMZ is to add an additional layer of security to an organization's local area network (LAN); an external attacker only has access to equipment in the DMZ, rather than any other part of the network.

Certificate validation is what it is all about. Verifying the authenticity and validity of a certificate (it is the public key) is a crucial process in ensuring the integrity of a PKI system. The CRL performs this function and the CRL is generated on a daily basis and stored in a single file output and transported to the DMZ. There are basically three problems to address when dealing with CRL information:

A single CRL file grows too big—as certificates gets added to the CRL the file will grow in size, and each time a client requires the CRL information it will attempt to download the file in order to check it. In a relatively small environment this will still suffice, but dealing with thousands of certificates will make the single file solution inadequate. In the context of the UAE system, in which millions of certificates will eventually end up on the revocation list, a single file will grow up to hundreds of megabyte and will be impractical to use in this environment.

A single file is not suitable for a distributed environment—accessing a single file is fine when the environment and the number of CRL queries are fairly small. However, from industry evolution it is clear that a single file solution for any scenario has its limitations and cannot scale very well. Take Windows NT and its single authentication file in the form of a SAM file: It was just a matter of time before Microsoft had to replace this technology with something that could scale and be distributed across a large infrastructure. The answer was a directory service, and Active Directory (like NetWare, iPlanet, IBM, and others) is a lightweight directory access protocol (LDAP). A LDAP is usually implemented with an online certificate status protocol (OCSP). A directory service with an access protocol makes the update and query much more efficient and reliable.

A CRL is accessed by both the secure CA as well as the less secure client queries—with a single CRL file, both secure agents, like the CA and less secure agents such as clients, require access to the CRL file. This is not the ideal way to maintain the level of security usually associated with a CA environment.

As the CRL grows in size and the number of requests increases, a distributed environment is the only proper solution for the demands of such an infrastructure. In a distributed environment multiple nodes, called responders, provide up-to-date CRL information in multiple points across the PKI footprint. Security is also addressed in the sense that the responders live in an unsecured environment but receive signed copies of the revocation information from the secured LDAP and distribution server, which can either be online or offline. With the size of the UAE PKI infrastructure in the form of number of certificates and role players in the future, this is definitely a recommended option to look at in order to comply with scalability, reliability, and security needs.

3.2.5 Level of Trust

As already mentioned, validating the CA certificate is an important part of the trust associated with a PKI infrastructure. The identity issuing authority implemented a standalone CA, which does not have a trusted path associated with the commercial CAs. This in itself is not a problem, but it means that the root CA certificate of the issuing authority has to be distributed to all entities that will require CA certificate validation. This becomes a

challenge, as other departments or even other countries use their own CAs with their own infrastructure. In the commercial world, cross-certification is the way to carry the trust of one CA over to another CA, resulting in implied trust through a trust path.

Due to security, autonomy, and other considerations, this might not be an appropriate solution, especially not when other countries are involved. Using a concept called bridge-CA might be a better solution, in which different departments or countries may use a joint bridge-CA instead of explicitly signing each other's root certificates. With the focus on GCC cooperation and interoperability, this solution might be considered in the future and is recommended above cross-certification.

3.2.6 Data Encryption

The final point to discuss is that of data encryption services, the one thing that currently does not exist in the UAE PKI infrastructure. Although a third certificate slot has been reserved in the PKI applet of the ID card for future purposes of data encryption/decryption, a very important aspect of the PKI infrastructure needs to be considered first—that of key recovery.

In an environment in which data is encrypted for storage—not only for the duration of a session like with the authentication certificate—key recovery becomes an issue. First of all, a private key holder (typically a cardholder) may lose his/her card or damage it. In this scenario the private key is lost and any data encrypted with the particular person's public key can no longer be decrypted. The ramifications might be huge if large amounts of sensitive information and even personal information are lost this way.

Second, in today's information age, in which criminal activities usually have a digital footprint and trail, the necessity for authorities to have access to data of any organization or individual when the need arises is crucial. Having no way to access encrypted information literally puts a blindfold on law enforcement activities.

It is clear that both from availability and law enforcement points of view, the need for key recovery has to be evaluated and considered very closely. An issue with having a form of private key backup is the fact that these keys need to be protected very well and the process of recovery has to be well defined and justified. This aspect of any PKI infrastructure has to be communicated to all certificate holders in very clear terms in the certificate practice statement. The certificate holder should have the confidence that he/she may be able to recover lost data in the event of a lost private key, but also that the Certificate Authority will not abuse its position of maintaining a copy of the backup keys.

A very important point to highlight is that the keys for digital signatures and data decryption will preferably not be the same keys. This might sound strange, but think of it for a moment. If the private key is backed up or escrowed (managed by a third party), the non-repudiation associated with a digital signature is no longer valid. This is because another private key exists, which is fine for recovering encrypted data in the event of key loss, but not fine when it comes to ensuring non-repudiation with singed transactions. This is the reason why there are two distinct certificates—digital signature certificates require archival of the public keys while data encryption certificates require archival/back-up of the private keys.

In the UAE system no key recovery or private key escrow exists. It was very important that key recovery requirements be understood before attempting to implement data encryption services within the ID card. It is recommended to have the architecture in place before any decisions are made as to making data encryption/decryption available on the ID card.

Conclusion

This article has presented an overview of the major PKI components deployed in the UAE national identity management infrastructure, with emphasis on the practical side of the implementation. The UAE PKI infrastructure is only in its beginning phase and will grow as the need for e-services increases. It was important to understand all the aspects of the infrastructure and what the limitations and strengths of various implementations and uses are. It was also very important to get the architecture right up front before implementing the bulk of e-services.

Due to the complex nature and security requirements of a PKI infrastructure, mistakes in the architecture cannot be easily rectified at a later stage and proper planning is of the utmost importance. There was a clear need to understand the full specifications of CA, its supported standards, and related services. This should facilitate the development of e-service applications and extending the existing infrastructure.

Furthermore it was recommended to establish and maintain a regular interaction with other e-government role players, Etisalat, which the local telecom operator in UAE that is in charge of the commercial CA, and any UAE lawmakers as far as e-commerce and digital communications security is concerned. The proposed approach was to have a forum in which role players can share knowledge and experience, and in which the future roadmap of requirements and interoperability is discussed on a regular basis.

References

Barr, T.H. (2002). Invitation to Cryptology. Upper Saddle River, NJ: Prentice Hall.

Ferguson, N., Schneier, B., & Kohno, T. (2010). Cryptography Engineering: Design Principles and Practical Applications. New York: John Wiley & Sons.

GAO (2001) "Advances and Remaining Challenges to Adoption of Public Key Infrastructure Technology", United States, General Accounting Office report GAO-01-277, February 2001.

Judge, P. (2002) PKI is failing, say Sun and Microsoft, ZDNet.com.au. [Online]. Available from: http://www.zdnet.com.au/pki-is-failing-say-sun-and-microsoft-120268957.htm

Kuhn, D.R.; Hu, V.C.; Polk, W.T. and Chang, S.J. (2001) Introduction to Public Key Technology and the Federal PKI Infrastructure, National Institue of Standards and Technology (NIST) [Online]. Available from: http://csrc.nist.gov/publications/nistpubs/800-32/sp800-32.pdf Accessed [12 October, 2011].

Lloyd, S. and Adams, C. (1999) Understanding the Public-Key Infrastructure: Concepts, Standards, and Deployment Considerations. Sams Publishing.

Pluswich, L. and Hartman, D. (2001) "Prime-Time Player?", Information Security Magazine, March.

Rothke, B. (2001) “PKI: An Insider View”, Information Security Magazine, October 2001.

Schwemmer, J. (2001) Solutions and Problems - (Why) It's a long Way to Interoperability. Datenschutz und Datensicherheit 25(9).

Spillman, R.J. (2005). Classical and Contemporary Cryptology. Upper Saddle River, NJ: Pearson Prentice-Hall.

Stallings, W. (2006). Cryptography and Network Security: Principles and Practice, 4th ed. Englewood Cliffs, NJ: Prentice Hall.

About the Author

Dr. Al-Khouri is the Director General (Under Secretary) of Emirates Identity Authority; a federal government organisation established in 2004 to rollout and manage the national identity management infrastructure program in the United Arab Emirates. He has been involved in the UAE national identity card since its early conceptual phases during his work with the Ministry of Interior. He has also been involved in many other strategic government initiatives in the past 21 years of his experience in the government sector.

He holds an engineering doctorate degree in strategic and large scale programs management from Warwick University, UK; Masters Degree (M.Sc.) in Information Management from Lancaster University, UK; and a Bachelors Degree (B.Sc., Hons.) from Manchester University, UK. He is also a member in several academic and professional institutions.

He is an active researcher in the field of advanced technologies implementation in government sector, and the approaches to reinventing governments and revolutionising public sector services and electronic business. He has published more than 50 research articles in various areas of applications in the past 10 years.