J. Lopez, R. Oppliger, G. Pernul. Why have Public Key Infrastructures failed so far. Internet Research, Vol. 15, Issue 5, Oct. 2005. Emerald, Bradford, England

WHY HAVE PUBLIC KEY INFRASTRUCTURES FAILED SO FAR?

Javier Lopez[1], Rolf Oppliger[2], Günther Pernul[3]

Abstract:

Since public key cryptography is a fundamental technology for electronic commerce, people have often argued that public key infrastructures and corresponding certification services are the gold-mines of the information age. Contrary to these relatively high expectations, public key infrastructures have not really taken off and many certification service providers have even gone out of business. In this paper, we overview and discuss the technical, economical, legal, and social reasons why public key infrastructures have failed so far, summarize the lessons learnt, and give our expectations about the future development of the field.

Keywords:

Public key certificates, digital certificates, public key infrastructures, certification service providers

[1] University of Malaga, Department of Computer Science, E.T.S. Ingenieria Informatica, Campus de Teatinos, E-29071 Malaga, Spain, Phone: +34 952 13 1327, Fax: +34 952 13 1397, E-mail: *jlm@lcc.uma.es*

[2] eSECURITY Technologies Rolf Oppliger, Beethovenstrasse 10, CH-3073 Gümligen, Switzerland, Phone/Fax: +41 (0)79 654 8437, E-mail: *rolf.oppliger@esecurity.ch*

[3] University of Regensburg, Department of Information Systems, Universitätsstrasse 31, D-93053 Regensburg, Germany, Phone: +49 (0) 941 943 2742, Fax: +49 (0) 941 943 2744, E-mail: *guenther.pernul@wiwi.uni-regensburg.de*

1. INTRODUCTION

There is an extended opinion that the Internet may become a universal communications medium that may eventually replace dedicated telephone, television, and radio networks. In spite of the fact that the Internet is growing at explosive rates, it is still constrained by security concerns. Every company and individual needs the confidence that business conducted electronically over the Internet is safely completed with the expected parties.

This is not a problem in the physical world because of the physical interaction of parties. However, things become more complicated in a digital environment because the technology available provides means for the interception, monitoring and forging of messages, and even impersonation of the participating peers.

The employment of cryptographic techniques using symmetric keys can be considered as a simple way to protect data. The major problem is that on the Internet one cannot assure that parties involved in a transaction can meet physically, or even know each other beforehand. In these circumstances, the provision of security services is challenging, and this is particularly true for user authentication.

The need for user authentication is evident in many public administration and e-commerce applications. For example, there are multiple instances where two unknown officials in different branches of the public administration need to securely exchange documents. Several systems, such as Kerberos (Kohl 1989; Kohl et al. 1993), have been proposed to provide authentication over public networks using secret key cryptography. Those systems are not easily scalable for large groups of users (possibly belonging to different organizations). Some efforts have been accomplished to solve this problem, like Davis (1995), Ganesan (1995), and Schiller et al. (1995), but the resulting systems are not widely deployed.

On the other hand, public key cryptography, as introduced by Diffie et al. (1976), is a very powerful technology and seems to be well suited to satisfy the requirements of the global Internet. In fact, it is commonly agreed that this technology is fundamental for a flourishing *electronic commerce* (e-commerce) in Internet, and has become the foundation for many e-commerce applications.

Public-key systems are based on the fact that every user has two keys, a public and a private one. The public key is accessible by the public and can be requested from a directory service, whereas the private key is kept secret by its owner. The dual way of using the public and private parts of the key pair - encrypting with the public key and decrypting with the private one, or encrypting with the private key and decrypting with the public one - allows to apply asymmetric cryptography for encryption/decryption of data, distribution of shared secret keys, and generation/verification of digital signatures.

The widespread use of public key cryptography requires a *public key infrastructures* (PKI), (Ford et al. 2000; Adams et al. 2002). The aim of a PKI is to make sure that a public key in use really belongs to the claimed entity. Without a PKI, public key cryptography would only be marginally more useful than traditional secret key cryptography. During boom time, the developers of PKIs expected not only to solve most problems concerning the security of transferred data, but also to provide general solutions for e-commerce, for example, concerning the provision of various non-repudiation services (Zhou 2001). Against this

background, people have often argued that PKIs and corresponding certification services are the gold-mines of the information age.

However, and in contrast to these relatively high expectations, PKIs have not really taken off, and many *certification service providers* (CSPs) have even gone out of business. In 1997, Frost & Sullivan envisaged the beginning of the PKI thriving that was expected to endure for at least the next decade. While according to Datamonitor (1999) the revenues for the entire PKI market were expected to reach 1.4 billion USD by 2003 , today there are only a few companies actually making profit from selling PKI products or acting as CSPs.

In this paper, we elaborate on the reasons for this PKI failure. More precisely, we discuss the major issues that have constrained the expected success of PKIs. The rest of the paper is organized as follows: In Section 2, we elaborate on PKIs. Section 3 is the main part of the paper. It comprises a discussion of the technical, economical, legal, and social reasons why PKIs have failed so far. Finally, in Section 4, we summarize the lessons learnt and conclude the paper by giving some expectations about how the field may possibly evolve in the future.

2. PUBLIC KEY INFRASTRUCTURES

The term certificate refers to "a document that attests to the truth of something or the ownership of something", RFC 2828 (2000). Historically, the term certificate was coined and first used to refer to a digitally signed record holding a name and a public key. As such, the certificate attests to the legitimate ownership of a public key and attributes a public key to a particular entity, such as a person, a hardware device, or anything else. The resulting certificates are called *public key certificates*.

According to the same RFC, a public key certificate is a special case of a digital certificate, namely one "that binds a system entity's identity to a public key value, and possibly to additional data items." As such, it is a digitally signed data structure that attests to the ownership of a public key (Oppliger 2002).

There are several possibilities to classify public key certificates. For example, it is common practice to define classes of certificates according to the quality of the registration process(es). In addition, certificates can also be classified according to the storage media in use (e.g., smartcards) and the type of functionality they can be used for (Lopez at al. 2005).

In accordance with the aforementioned RFC, a certificate can not only be used to attest to the legitimate ownership of a public key (in the case of a public key certificate), but also to attest to the truth of any property attributable to the certificate owner. This more general class of certificates is commonly referred to as *attribute certificates*. In short, the major difference between a public key certificate and an attribute certificate is that the former links the name of the user with his/her public key, whereas the latter links the user with a list of generic characteristics.

In either case, the certificates are issued (and possibly revoked) by authorities that are recognized and trusted by some community of users. In the case of public key certificates, these authorities are called *certification authorities* (CAs), whereas in the case of attribute certificates, these authorities are called *attribute authorities* (AAs). More recently, the term CSP has been coined to refer to a CA that is providing certification services.

This line of argumentation leads to the observation that, from a practical viewpoint, attribute certificates are going to play a very important role in the near future. For many applications, successful user authentication is only the first step, and what is additionally needed is to provide evidence that a particular user possesses the proper rights to perform a requested action. Therefore, authorization services are equally important. Lopez et al. (2004) give examples *authentication and authorization infrastructures* (AAIs) in which these functionalities are going to merge.

Generally speaking, we can consider that a PKI consists of one (or several) CA(s). From RFC 2828, a PKI is "a system of CAs that performs some set of certificate management, archive management, key management, and token management functions for a community of users" that employ public key cryptography. Another way to look at a PKI is as an infrastructure that can be used to issue, validate, and revoke public keys and public key certificates. As such, a PKI comprises a set of agreed-upon standards, CAs, structures among multiple CAs, methods to discover and validate certification paths, operational and management protocols, interoperable tools, and supporting legislation. A PKI and the certification service it provides must be specified in a *certification policy* (CP) and a *certification practice statement* (CPS), RFC 3647 (2003).

Many standardization bodies are working in the field of certificates and PKIs. Most importantly, the Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T) has released and is periodically updating a recommendation that is commonly referred to as ITU-T X.509, or X.509 in short. The current version of X.509 is version 3. Meanwhile, the X.509 has also been adopted by many other standardization bodies, including, for example, the ISO/IEC JTC1.

The format of an X.509v3 certificate is specified with the abstract syntax notation one (ASN.1) as shown below (Kaliski 1993).

```
Certificate  ::=      SIGNED { SEQUENCE {

  version                    [0]  Version DEFAULT v1,
  serialNumber                    CertificateSerialNumber,
  signature                       AlgorithmIdentifier,
  issuer                          Name,
  validity                        Validity,
  subject                         Name,
  subjectPublicKeyInfo            SubjectPublicKeyInfo,
  issuerUniqueIdentifier     [1]  IMPLICIT UniqueIdentifier OPTIONAL,
                                       -- if present, version shall be v2 or v3
  subjectUniqueIdentifier    [2]  IMPLICIT UniqueIdentifier OPTIONAL,
                                       -- if present, version shall be v2 or v3
  extensions                 [3]  Extensions OPTIONAL
                                       -- If present, version shall be v3 –

} }
```

The description of the fields is as follows:

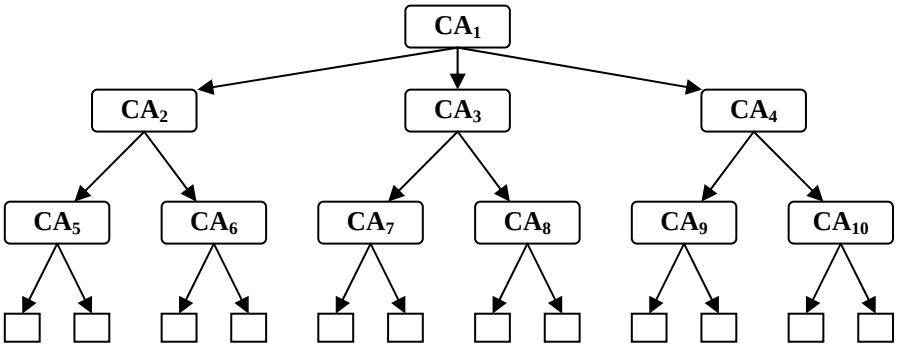
- A *version* number (identifying version 1, version 2, or version 3);
- A *serial number* (i.e., a unique integer value assigned by the issuer);
- An *object identifier* (OID) that specifies the signature algorithm that is used to sign the public key certificate;

- The distinguished name (DN) of the *issuer* (i.e., the name of the CA that actually signed the certificate);
- A *validity period* that specifies an interval in which the certificate is valid;
- The DN of the *subject* (i.e., the owner of the certificate);
- Information related to the subject's *public key* (i.e., the key and the OID of the algorithm);
- Some optional information used to uniquely identify an issuer in case of name re-use (defined for versions 2 and 3 only);
- Some optional information used to uniquely identify a subject in case of name re-use (defined for versions 2 and 3 only);
- The extensions field allows the addition of new fields to the structure without modification to the ASN.1 definition. An extension field consists of an extension identifier, a criticality flag, and an encoding of a data value of an ASN.1 type associated with the identified extension. (defined for version 3 only).

X.509 certificates are encoded using some specific encoding rules to produce a series of bits and bytes suitable for transmission. There are three standardized encoding rules, namely the *basic encoding rules* (BER), the *distinguished encoding rules* (DER), and the *packet encoding rules* (PER).

The trust model employed by X.509 is often referred to as being hierarchical.[4] This basically means that a user must define a number of root CAs and corresponding root certificates (i.e., certificates that are trusted by default) from which trust may extend (see Figure 1). Typically, a root certificate is self-signed, meaning that the root CA has issued its own certificate (i.e., the subject and issuer are identical). Note that from a theoretical point of view, self-signed certificates are not particularly useful. Anybody can claim something and issue a certificate for this claim. Consequently, a self-signed certificate basically says: "Here is my public key, trust me."

Having established a number of root CAs and corresponding root certificates, a user can try to find a certification path (or certification chain, respectively) that leads from a root certificate to a leaf certificate (i.e., a certificate that is issued for a user or system).



[4] Note, however, that ITU-T X.509 does not embody a hierarchic trust model. The existence of cross-certificates, as well as forward and reverse certificates, makes the X.509 model a mesh, analogous in some ways to PGP's web of trust, Zimmermann (1995). The X.509 model is often erroneously characterized as a hierarchic trust model because it is usually mapped to the *directory information tree* (DIT), which is hierarchic, more like name schemes.

Figure 1: Hierarchical trust model

Formally speaking, a certification path or chain is defined in a tree or wood of CAs (root CAs and intermediate CAs) and refers to a sequence of one or more certificates that lead from a root certificate to a leaf certificate. Each certificate certifies the public key of its successor. Finally, the leaf certificate is typically issued for a person or a system. A certification path can only be verified if all CAs on the path are trusted. In practice, certification paths are comparably short, and it is hardly understood how different certification paths can be combined in some meaningful way (Maurer 1996).

X.509 can be used in many ways. Consequently, every nontrivial group of users who want to work with X.509 certificates has to produce a profile which nails down the features which are left undefined in X.509. The difference between a specification (i.e., X.509) and a profile is that a specification does not generally set any limitations on what combinations can and cannot appear in various certificate types, whereas a profile sets various limitations, for example, by requiring that signing and confidentiality keys be different. Many standardization bodies work in the field of "profiling" X.509 for specific application environments. For example, the Internet Engineering Task Force (IETF) has chartered the PKI X.509 (PKIX[5]) working group (WG) to profile the use of X.509 on the Internet. The IETF PKIX WG is a dynamic and very active WG that has published many documents.

3. ANALYSIS OF THE REASONS FOR THE PKI FAILURE

In the last years, PKIs have experienced a hype and many companies and organizations have announced to provide certification services to the general public. Unfortunately, only a few of these companies and organizations have succeeded and actually provide certification services that can be taken seriously.

We have identified a number of reasons that may have led to the PKI failure that we have recently experienced. For a better understanding, we have classified those reasons into categories. In this sense, there are technical, economical, legal, and social reasons. Understanding these reasons is necessary to make sure that PKIs can be successfully deployed in the future.

3.1 TECHNICAL REASONS

The technical reasons for the PKI failure all have to do with the fact that the establishment and operation of a PKI is more involved than one might think at first sight. In this subsection we elaborate on the following technical issues: complexity, certificate management, global name space and cross-certification. We argue why those issues can be considered as reasons for the failure of PKIs wide deployment.

[5] http://www.ietf.org/html.charters/pkix-charter.html

3.1.1 Complexity

We believe that X.509 is a complex standard and X.509 certificates are complex data structures, as it can be assumed from the definition of the structure certificate in previous section. The complexity is due to the fact that X.509 certificates comprise many fields, but moreover, comprise many (critical and/or non-critical) extension fields. Additionally, X.509 certificates are specified with ASN.1 and must be encoded according to some encoding rules (see above). The resulting data structures are non-intuitive and not very meaningful for human readers. They are also comparably difficult to parse (by automated data processing processes). This is in contrast, for example, to PGP certificates. In the future, XML-based certificates may become an interesting alternative.

In general, a complex technology is not easy to deploy on a large-scale. Precisely, PKI was conceived to be used on a large scale basis. There are many things that can be interpreted and/or implemented differently when it comes to X.509 certificates. Consequently, X.509 products and services are not as interoperable as one would suggest at first sight, and they frequently suffer poor interoperability (partly due to the flexibility of the X.509 specification and the missing profiling). Furthermore, large complexity may also lead to poor usability (see below).

3.1.2 Certificate Management

Certificate management is a complex and very challenging task, and there are many things that can go wrong. For example, public key pairs must be generated in an efficient and secure way. This can be done in a centralized or decentralized way. In either case, one needs a cryptographically strong pseudo-random bit generator for the generation of the key pair, what is not always available to end-users. Alternatively, the CSP can provide that facility to end-users. However, under that assumption, any user may have the possibility to repudiate signatures performed using the corresponding private key, claiming an eventual disclose of such key at the CSP side (computer system) during the generation (and eventual following storage) of the key pair.

In relation with private keys, they must be securely stored in a personal security environment (PSE), and the certificates must be made publicly available in a certificate repository or directory service. In either case, it must be made sure that certificates and certificate chains are retrieved from there, and that they are verified accordingly. Hierarchical trust models are particularly dangerous, because they provide a single point of attack (Burmester et al. 2004), regardless the number of levels of CAs. Alternative models, like the mesh model (i.e., the PGP web of trust) are not a solution because they tend to become poorly scalable in time and in space.

From a more technical viewpoint, certificate revocation is probably the most challenging task of certificate management. A basic question is ”how can one make sure that status information about a certificate is distributed and used together with the certificate?”. The standard way to do certificate revocation is to have a CA periodically issue a *certification revocation list* (CRL), i.e., a list of certificates that have been revoked. As further addressed in Rivest (1998), the use of CRLs has several important limitations and shortcomings. The more direct one is that whenever a user wants to verify a signature, he/she will have to get the corresponding certificates in the certification chain and, additionally, will have to check that those certificates in the chain are not in the CRL.

There are a few alternative and/or complementary approaches to handle the certificate revocation problem (Oppliger 2002). However, and unfortunately, the certificate revocation issue is not properly solved (and sometimes not even addressed) by many applications in use today. The underlying reason for this fact is that a proper solution for the certificate revocation problem requires an online component; for instance, an *On-line Certificate Status Protocol* (OCSP) responder that is costly to establish and operate, RFC 2560 (1999).

3.1.3 Global Name Space

As mentioned above, X.509 certificates include X.500 DNs. These DNs, in turn, identify objects in a global (i.e., globally unique) name space. Unfortunately, the definition and maintenance of a global name space is not as simple in practice as theory suggests. In fact, there are only a few examples of global name spaces that work in practice (one example is the domain name system). Many practical problems related to X.509-based PKIs are related to the fact that these PKIs depend on the notion of X.500 DNs.

In the version 3 of X.509, the ITU-T tried to solve this problem by creating the "Subject alternative name" extension, a field to contain one or more alternative names using a variety of name forms for the entity bounded by the CA to the certified public key. However, this has not solved the problems. The definition of the field is:

```
subjectAltName EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-subjectAltName }
```

where **GeneralNames** is:

```
GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
    otherName                   [0]     INSTANCE OF OTHER-NAME,
    rfc822Name                  [1]     IA5String,
    dNSName                     [2]     IA5String,
    x400Address                 [3]     ORAddress,
    directoryName               [4]     Name,
    ediPartyName                [5]     EDIPartyName,
    uniformResourceIdentifier   [6]     IA5String,
    iPAddress                   [7]     OCTET STRING,
    registeredID                [8]     OBJECT IDENTIFIER }
```

As other authors propose, instead of using a global name space, one may use local name spaces that are linked in one way or another. A corresponding set of certificate formats and protocols was developed in the late 1990s by the IETF *Simple Public Key Infrastructure* (SPKI) WG, RFC 2693 (1999). In parallel, a team at the MIT worked out the *Simple Distributed Security Infrastructure* (SDSI) (Rivest et al. (1996), which is in its basic concepts very similar to the SPKI idea.

The central aspect of the SPKI/ SDSI development is not to bind a key to a certain identity, but rather to a role or an authorization and to use only locally unique names instead of global names. Therefore, names are only exclusive within a certain context – the local name space.

However, SPKI/SDSI has not been adopted by the industry so far. Therefore, the problem remains open.

3.1.4 Cross-Certification

In the past, people have sometimes argued that CAs can cross-certify each other to form multi-CA PKIs. Cross-certification, however, requires that the corresponding CPSs are equal (or at least very comparable). Unfortunately, cross-certification does not work in practice. Certification Service Providers typically argue that the certification services they provide are better than the ones of the competitors, and hence that they are not able to cross-certify them.

There is hardly any incentive for cross-certification, and the authors are not aware of any non-trivial cross-certification used in practice. Federated identity management faces very similar problems, and hence it will be interesting to see if and how federated identity management solves these problems in practice.

3.1.5 Unproven Assumptions

Last but not least, we note that public key cryptography is based on the unproven assumption that (trapdoor) one-way functions exist (Oppliger 2005). If this assumption turned out to be false, then public key cryptography in general, and public key certification in particular would become useless. This point may be interesting from a theoretical viewpoint. It is, however, not really relevant for the failure of PKIs, and hence is not further addressed in this paper.

3.2 ECONOMICAL REASONS

The economical reasons for the PKI failure all have to do with the fact that the establishment and operation of a PKI is an expensive endeavour and that it is difficult to charge users for certificates (not only when the user gets his first certificate but, as some PKI providers pretend, also in subsequent issues due to revocation or expiration reasons). In this subsection we discuss on the following economical reasons: large investments, return of investment and business case.

3.2.1 Large Investments

The establishment and operation of a PKI requires large investments. For example, the PKI must be established and operated in a physically secure environment. In national digital signature laws, various guidelines exist to assure the security and safety of the facilities in which a CA is operating. This includes regulations on the thickness of the walls of the building, on doors and windows, even on entrance control systems and video monitoring. Certainly, fire protection concepts and those against penetrating water have to be installed, too.

Except the violent intrusion into the CA's facilities the operators have to consider the attacker using other approaches to undermine the system. One scenario is based on the compromising radiation emitted by the operating computers. Intercepting the

electromagnetic signals (e.g. from the processor, monitor, or the graphic accelerator) for instance could help the assailant to reproduce the keystrokes that were performed when the pass phrase for the CA's signing key was entered. Forgers might try to exploit other weaknesses of the used hardware involved in the certification process.

In order to encounter these, for example, the German digital signature law (SigG 2001), explicitly requires every single element involved in generating qualified certificates to be evaluated either by the Information Technology Security Evaluation Criteria (ITSEC) or the Common Criteria for Information Technology Security Evaluation (CC). Technical components and those involved into the secure generation of the signatures have to withstand an examination conform to the level EAL4 (CC) or E3/High (ITSEC). The security enhancing mechanisms and the evaluation of the hardware and software is extraordinary time consuming and expensive. Last but not least, there is also the need for the adequately eligible personnel, being able not only to implement the concepts into the processes and products, but also to install and maintain the necessary software.

On the user's side, providing each user with hardware devices, such as smart cards and card readers, for the secure storage of private key(s) is costly, especially in the case of a large user population. Consequently, people often go for the less secure alternative of using softtokens. In this case, the private keys and certificates are stored (and protected) only in software. This is arguably less secure, but reduces the deployment costs of PKIs considerably.

3.2.2 Return on Investment

When it comes to economical considerations, people often wonder about the return on investment (ROI). If the establishment and operation of a PKI requires a large investment (as mentioned above), then the ROI is particularly important. Unfortunately, the ROI of a PKI is very difficult to determine and quantify. Part of the problem is that – like many other infrastructure components – public key certificates do not provide a specific function that can be charged, but provides only the means to secure functions.

3.2.3 Business Case

Taken into account the large investments and the hard-to-determine ROI of a PKI, people have been looking into possibilities to come up with business cases for CSPs without success. In Switzerland, for example, a group of companies and organizations formed the IG tOP[6] (Trägerschaft öffentliche PKI) after the major CSP went out of business in 2001. The aim of the IG tOP was to specify a business case for a public PKI that could be implemented by any CSP. The IG tOP was not particularly successful, and the major result was that it is very difficult to make a living from the provision of certification services.

[6] http://www.igtop.ch

3.3 LEGAL REASONS

The legal reasons for the PKI failure have to do with the fact that there are many open questions with regard to the liability of digital signatures and certificates, and that non-repudiation may represent a dual-edged sword (from the user's viewpoint), meaning that non-repudiation is not always appreciated. In this subsection we elaborate on the liability and non-repudiation issues.

3.3.1 Liability

Liability is an important building block when it comes to digital signature legislation. In fact, if a digital signature is generated according to some digital signature law, then somebody must be made liable if something goes wrong. For example, the German legislation (SigG 2001), makes the certificate provider liable if the damage is caused by the failure of the technical components, the misbehaviour of the CSP (i.e. its employees) or other violations against the law. This may force the CSP to create reserve funds or to get an insurance in order to compensate eventual damage that may result from the misuse of the certificates, as it is the case of the Spanish legislation. The insurance protection, in turn, is expensive and must be taken into account in a business case.

3.3.2 Non-repudiation

The owner of a public key certificate cannot repudiate a signature that is generated with the appropriate signing key. From the certificate owner's viewpoint, this may be disadvantageous. In fact, it may lead to the situation in which a certificate owner may be held liable and accountable for statements that are digitally signed with the proper key, even if he or she does not know (and does not agree) with the statements. There are several limitations and shortcomings of digital signature schemes that must be considered with care, and that may limit the deployment and use of digital signatures in practice. The limitations and shortcomings are discussed in Oppliger et al. (2004) and Maurer (2003, 2004). In short, although many properties of digital signature systems can be proven in a mathematically strong sense, there are many open questions and challenges when it comes to an implementation and real-world deployment of the systems on a large scale. What happens, for example, if a binary string is a valid representation of two digital objects (e.g., a Word file and an image file)? To which object would a digital signature be attributed?

3.4 SOCIAL REASONS

Last but not least, there are also a couple of social reasons that may be partly responsible for the PKI failure. In this section, we will elaborate on the following issues: notion of trust, poor usability and lack of awareness.

3.4.1 Notion of Trust

People often argue that certificates and PKIs can be used to establish trust. The authors think that the notion of trust is sometimes badly understood, and that a clear distinction must be made between the notion of trust and what certificates can be used for. In the real world,

trusted relationships are based on bilateral relationships and experiences that have been made over time. Consequently, trust can only be established slowly, but it can be destroyed almost immediately.

However, these properties (of trust) can only insufficiently be modelled with certificates. It is important to note that, in certain way, certificates break the bilateral relationship by introducing a trusted third party (i.e., the CSP). Furthermore, it is often required that a party that has a certificate can be trusted without having made experiences. So the level of trust we get from certificates is often overestimated.

3.4.2 Poor Usability

The usage of public key cryptography in general, and public key certificates in particular, is less trivial than postulated by vendors. In fact, poor usability is a common feature of many products that employ (public key) cryptography (Whitten 2004). There is still a lot of room for research and development that focus on the end user in order to find cryptographic solutions that, being secure, are usable too. Examples include trusted devices, appropriate user interfaces, embedding the cryptographic solutions in the applications.

3.4.3 Lack of Awareness

The users of public key cryptography are often not aware of the vulnerabilities and pitfalls. If, for example, an SSL/TLS session is established to a secure Web server, then the user is expected to verify the authenticity of the certificate provided by the server. If the certificate is issued by a trusted CA, then the user must do nothing particular. If, however, the certificate is issued by some untrusted (or even unknown) CA, then the user must decide whether he or she accepts the certificate.

Unfortunately, it is possible and very likely that the user accepts the certificate by clicking the corresponding checkbox without thinking and without considering the further implications (Ellison et al. 2000). There is a long way to go until people are sufficiently aware of the vulnerabilities and pitfalls of public key cryptography.

4. LESSONS LEARNT AND CONCLUSIONS

After having argued that PKIs have failed and having identified a number of reasons for this failure, one may be tempted to conclude that digital certificates and PKIs are not particularly useful, and that they will slowly disappear. This conclusion, however, is too short-sighted.

Public key cryptography in general, and digital signatures and public key-based key establishment procedures are simply too valuable than not to be used in practice. In fact, there is hardly any alternative to the use of digital signatures to provide non-repudiation services on a large-scale. Whenever public key cryptography is employed, certified public keys must also be made available in one way or another. Consequently, there are definitively (many) useful applications for public key certificates and PKIs, and hence it is reasonable to expect that public key certificates and PKIs will be highly deployed in the future.

In fact, we do believe that the main reason why hardly any CSP is providing certification services to the general public is economical: it is very difficult to find a business case. The

underlying reason for this difficulty is that only a few people are willing to buy certificates. To make things worse, there is a chicken-and-egg-problem: Without applications, there is hardly any incentive to buy a certificate, and without certificates that are widely deployed, there is hardly any incentive to build applications that make use of certificates.

In some countries, politicians have argued that it is necessary that the state solves the problem by providing electronic ID cards that comprise certificates to citizens. This line of argumentation, however, has problems of its own (not addressed in this paper).

If one were able to solve the economical problem(s) and find a way to successfully market certificates, then the technical and social problems could be comparably simple to solve. With respect to the legal problems, the situation is more involved. Digital signature legislation is and continues to be very difficult to handle, and many people will attempt to minimize their liability. In the most extreme case, they will try not to use the technology in the first place. Consequently, the promoters of digital signatures will have to work with economical incentives. For example, a service provider can argue that a customer who digitally signs all of his or her transactions is authorized to get a discount.

Contrary to the common belief that public key certificates and certification services can be marketed independently from applications and application environments, we do not think this way. Instead, we think that applications and application environments will come along with their public key certificates and PKIs. Some of them will be very specific and even proprietary to some extent (this contradicts, for example, to the aforementioned vision of electronic ID cards).

Moreover, we think that the use of public key certificates and PKIs will be deeply intertwined with future operating systems. The Microsoft Windows 2003 PKI Server and the support and use of public key certificates in contemporary Windows operating systems point in that direction. Whenever a user is registered and provided with an account, it is fairly simple and straightforward to also equip him or her with a public key pair and a corresponding public key certificate.

These types of public key certificates will be different from the ones one usually has in mind when one talks about public key certificates that are issued in the context of digital signature acts. From the user's point of view, these certificates are not completely different from passwords. Note that a user typically also has to enter a password to unlock and make use of a private key. So the use and deployment of public key certificates and PKIs may go unnoticed and in a subliminal way.

It is not always necessary to make users aware of the full complexity a technology brings with itself. Against this background, public key certificates and PKIs may have a bright future and one may expect them to be omnipresent in future computing and networking environments; this does not necessarily mean that they will be visible.

REFERENCES

Adams et al. (2002). Adams, C., and S. Lloyd. *Understanding PKI: Concepts, Standards, and Deployment Considerations*, Addison-Wesley, 2002

Burmester et al. (2004). Burmester, M., and Y. Desmedt. *Is hierarchical public-key certification the next target for hackers?* Communications of the ACM, Vol. 47, No. 8, August 2004, pp. 68-74.

Davis (1995). Davis, D. *Kerberos Plus RSA for World Wide Web Security*. Proceedings of the 1st USENIX Workshop on Electronic Commerce, 1995, pp. 185-188.

Datamonitor (1999). Global PKI Markets 1999-2003. Datamonitor report, Nov. 1999.

Diffie et al. (1976). Diffie, W., and M. Hellman. *New Directions in Cryptography*. *IEEE Transactions on Information Theory*. IT-22, n. 6. 1976, pp. 644-654.

Ellison et al. (2000). Ellison, C., and B. Schneier. *Ten Risks of PKI: What You' re not Being Told about Public Key Infrastructure*. Computer Security Journal, Vol. XVI, No. 1, 2000.

Ford et al. (2000). Ford, W., and M. Baum. *Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption. Second Edition*. Prentice Hall PTR,2000

Ganesan (1995). Ganesan, R. Yaksha. *Augmenting Kerberos with Public Key Cryptography*. Internet Society Symposium on Network and Distributed Systems Security. IEEE Press, 1995, pp. 132-143.

Kaliski (1993). Kaliski, B. *A Layman's Guide to a Subset of ASN.1, BER, and DER*, RSA Laboratories Technical Note, November 1993.

Kohl et al. (1993). Kohl, J., and B.C. Neuman. *The Kerberos Network Authentication Service (V5)*. 1993.

Kohl (1989). Kohl, J. *The Use of Encryption in Kerberos for Network Authentication*. Proceedings of CRYPTO '89. Springer-Verlag, LNCS 435, 1989, pp. 35-43.

Lopez et al. (2004). Lopez, J., Oppliger, R., and G. Pernul. *Authentication and authorization infrastructures (AAIs): a comparative survey*. Computers & Security, Vol. 23, No. 7, October 2004, pp. 578-590.

Lopez et al. (2005). Lopez, J., Oppliger, R., and G. Pernul. *Classifying public key certificates*. Proc. 2nd European PKI Workshop. The University of Kent, England, June 2005.

Maurer (1996). Maurer, U.M. *Modelling a Public-Key Infrastructure*. Proceedings of the European Symposium on Research in Computer Security (ESORICS' 96), Springer-Verlag, LNCS 1146, 1996, pp. 325-350.

Maurer (2003). Maurer, U. *Intrinsic Limitations of Digital Signatures and How to Cope With Them*. Proceedings of the 6th Information Security Conference (ISC '03), Springer-Verlag, LNCS 2851, pp. 180-192.

Maurer (2004). Maurer, U. *New Approaches to Digital Evidence*. *Proceedings of the IEEE*, Vol. 92, No. 6, June 2004, pp. pp. 933-947.

Oppliger (2002). Oppliger, R. *Security Technologies for the World Wide Web. Second Edition*. Artech House, Norwood, MA, 2002.

Oppliger (2005). Oppliger, R. *Contemporary Cryptography*. Artech House, Norwood, MA, 2005.

Oppliger et al. (2004). Oppliger, R., and R. Rytz. *Digital Evidence: Dream and Reality*. *IEEE Security & Privacy*, Vol. 1, No. 5, September/October 2003, pp. 44-48.

RFC 2560 (1999). M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams. *X.509 Internet Public Key Infrastructure. Online Certificate Status Protocol – OCSP*. Request for Comments 2560, Network Working Group, IETF, June 1999.

RFC 2693 (1999). Ellison, C., et al. *SPKI Certificate Theory*. RFC 2693, September 1999.

RFC 2828 (2000). Shirey R. *Internet Security Glossary*. Request for Comments 2828, Network Working Group, IETF, May 2000.

RFC 3647 (2003). Chokhani, S., et al., *Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*, RFC 3647, November 2003

Rivest et al. (1996). Rivest, R. and Lampson, B. *SDSI - A Simple Distributed Security Infrastructure*, 1996.

Rivest (1998). R. Rivest. *Can we eliminate Certificate Revocation Lists?* Proceedings of Financial Cryptography, 1998.

Schiller et al. (1995). Schiller, J., and D. Atkins. *Scaling the Web of Trust: Combining Kerberos and PGP to Provide Large Scale Authentication*. Proceedings of the USENIX Technical Conference, 1995.

SigG (2001). *Gesetz über Rahmenbedingungen für elektronische Signaturen und zur Änderung weiterer Vorschriften*. Bundesgesetzblatt Jahrgang 2001 Teil I Nr. 22, Bonn am 21. Mai 2001.

Whitten (2004). Whitten, A. *Making Security Usable*. Ph.D. Thesis, Carnegie Mellon University, 2004.

Zhou (2001). Zhou, J. *Non-repudiation in Electronic Commerce*. Artech House, Norwood, MA, 2001.

Zimmerman (1995). P. Zimmerman. *The Official PGP User's Guide*. MIT Press, 1995.