International Journal of Mathematical Archive-7(4), 2016, 114-118
Available online through www.ijma.info ISSN 2229 – 5046

DISCUSSION OF DIFFERENT TYPES OF PUBLIC KEY INFRASTRUCTURE IN REAL LIFE

B. KUMARASWAMY ACHARY*
Research Scholar, Department of Mathematics, S. V. University, Tirupati, India.

Dr. V. VASU
Department of Mathematics, S. V. University, Tirupati, India.

(Received On: 16-03-16; Revised & Accepted On: 22-04-16)

ABSTRACT

Since public keys in asymmetric cryptosystems need not be kept secret, key management in those systems is simpler than in symmetric schemes. Private keys, however, must be kept secret. Also, public keys must be protected from falsification and abuse. Therefore, appropriate public-key infrastructures (PKI) must be set up. They are responsible for key distribution and management. In this paper, we describe how such public-key infrastructures work.

Key words: *Public Key Infrastructure (PKI), Certificate Authentic (CA), Certificate revocation list (CRLs).*

INTRODUCTION

A public key infrastructure (PKI) [1] supports the distribution and identification of public encryption keys, enabling users and computers to both securely exchange data over networks such as the Internet and verify the identity of the other party.

In this paper, we discussed different types of public keys used in our real life. The following steps we need to understand the public key infrastructure.

To summarise, any PKI must deal with the following issues:

- Key generation and management
- Certificate authentic? (CAs)
- Certificate revocation list (CRLs)

PUBLIC KEY INFRASTRUCTURE

A public key infrastructure, or PKI, is the sum total of everything required to securely use public key crypto. It’s surprisingly difficult and involved to assemble all of the necessary pieces of a Public Key Infrastructure or PKI into a working whole. Some significant PKI issues must be overcome before public key crypto is useful in most real world settings. For a discussion of some of the risks inherent in PKI.

A university certificate, or public key certificate, or simply a certificate contains a user’s name along with the user’s public key i.e. Hall ticket number. In most situations the corresponding university must be signed by a certificate authority, or CA, which acts as a trusted third party or TTP. By signing the certificate, the Certificate Authority or CA is confirming that the identity stated in the certificate is that of the holder of the corresponding private key. Note that the CA is not vouching for the identity of the holder of the certificate, since the certificate is public. Today, the largest commercial source for certificate is Verisign.

An important subtle point here is that verifying the signature does not verify the source of the certificate. Certificates are public knowledge, so, for example first person could send second person certificate to third person. Third person can’t assume he’s talking to second person just because he received second person valid certificate.

When you receive a certificate, you must verify the signature. If the certificate is signed by a CA [2] that you trust, then you would attempt to verify the signature using that CA’s public key. Anyone can create a certificate and claim to be anyone else only the verification of the signature can create trust in the validity of the certificate.

*Corresponding Author: B. Kumaraswamy Achary**
Research Scholar, Department of Mathematics, S. V. University, Tirupati, India.

A certificate could contain just about any other information that is deemed of value to the participants. However, the more information, the more likely the certificate will become invalid. For example, it might be tempting for a corporation to include the employee's department in his certificate. But then any reorganization will invalidate the certificate.

If a CA makes a mistake, the consequences can be dire. For example, a person once issued a signed certificate for students password to someone else; that is, a person gave the corresponding private key to someone other than students password. That someone else could then have acted (electronically, at least) as students password. This particular error was quickly detected, and the certificate was revoked apparently before any damage was done.

A experience in company certifical 'contains a user's name along with the user's identity number (public key). The certificate must be signed by company management (or) CA which acts as trusted this party.

This raises another PKI issue, namely certificate revocation. Certificates are usually issued with an expiration date. But if a private key is compromised, or it is discovered that a certificate was issued in error, the certificate must be revoked immediately. Most PKI schemes require regular certificate revocation lists, or CRLs, which are supposed to be used to filter compromised certificates. In some situations, this could place a significant burden on users, which means that it is likely to lead to mistakes and security flaws.

Next, we briefly discuss a few of the many trust models that are used today. Ultimately, you must rely on a digital signature to decide whether to trust a certificate. A basic issue in public key cryptography is determining whose signature you are willing to trust. There are several possible trust models that can be employed. We'll follow the terminology.

Perhaps the most obvious trust model is the monopoly model, where one universally trusted organization is the CA for the known universe. This approach is naturally favoured by whoever happens to be the biggest commercial CA at the time (currently, Verisign). Some have suggested that the government should play the role of the monopoly CA. however, many people don't trust the government.

One major drawback to the monopoly model is that it creates a very big target for attack. If the monopoly CA is ever compromised, the entire PKI system fails. Also, if you don't trust the monopoly CA, then the system is useless for you.

The oligarchy model is one step away from the monopoly model. In this model, there are multiple trusted CA. In fact, this is the approach that is used in Web browsers today a Web browser might be configured with 80 or more CA certificates. The security conscious user is free to decide which of the oligarchy CAs he is willing to trust and wluc.li lie is not.

At the opposite extreme from the monopoly model is the anarchy model. In this model, anyone can be a CA, and it's up to the users to decide which "CAs" they want to trust. In fact, (his approach is used in POP, where it goes by the name of "web of mist".

The anarchy model can place a significant burden on users. For example, suppose you in receive a certificate signed by first person and you don't know first person, but you do trust second person and second person says third person is trustworthy and third person vouches for first person. Should you then trust Flunk? This is clearly beyond the patience of the average user, who is likely to simply trust anybody or nobody in order avoid headaches like this.

There are many other PKI trust models, most of which try to provide reasonable flexibility while putting a minimal burden on the end users. The fact that there is no agreed upon trust model is itself one of the major problems with PKI.

DISCUSSED DIFFERENT TYPES OF PUBLIC KEYS IN REAL LIFE

Personal Security Environments

Importance

If a person wants to generate signatures or in bank cheque personal documents using a public-key system, then if a person needs a private key. Person must keep this key secret because everybody who knows the key can sign messages in person name or personal secret documents that were sent to person. Therefore, person needs a personal security environment (PSE) in which his private keys are securely stored. Since the private keys should not leave the PSE, it also does the signing or personal files.

Frequently, the PSE [3] also generates the private keys. If the private keys are generated elsewhere, then at least the generating institution knows person secret keys, which may corrupt the security of the system. On the other hand, secure key generation may require resources not present in the PSE. For example, for RSA keys random primes of a fixed bit length are required. In particular, the key generating environment must generate large, cryptographically secure, random numbers. If the random number generator of the PSE is weak, then the public-key system is insecure. It may therefore make sense to have the RSA keys generated by a trusted institution.

Implementation

For example, the more sensitive the documents that are signed or encrypted, the more secure the PSE must be. A simple PSE is a bank locker in banks that can be accessed only after entering a secret password. This password may, for example, be used to open the bank locker. The security of a company PSE relies on the security of the underlying company system, One may argue that company systems must be very secure anyway and that they are therefore able to protect the PSE. Company systems, for example, prevent unauthorized users from becoming company persons. On the other hand, it is well known that with sufficient effort the security of most company systems can be successfully attacked. Therefore, a company PSE is not adequate for applications that require high security.

It is more secure to put the PSE on a PAN (or) Aadhar-Person card. Person can carry his PAN (or) Aadhar card in his purse. If the card is in the PAN (or) Aadharreader, it only permits very limited access. Manipulating its hardware or software is very difficult (although successful attacks have been reported). Unfortunately, computations on PAN (or) Aadhar cards are still very slow. Therefore, it is impossible to decrypt large documents on a PAN (or) Aadhar card, so public keys encrypt session keys which, in turn, are used to encrypt the documents. The encrypted session key is appended to the encrypted document. The PAN (or) Aadhar card only decrypts the session key. The decryption of the document is then done on a fast PC or workstation.

Representation problem

Even if a first person uses a PAN (or) Aadhar for signing, there is still a Severe security problem. If a first person wants to sign a document, she starts program on her PC, which sends the document or its hash value, in the PAN or Aadhar or ATM card, where it is signed. With some effort, the attacker, Anuther person can manipulate the signing program on a first person's PC such that it sends a document to the PAN (or) Aadhar that is different from the. one that a first person intended to sign. Because the PAN (or) Aadhar has no display, a first person is unable to detect this fraud. It is therefore possible that a first person could sign documents that she never wanted to sign. This problem is called the representation problem for signatures. The more important documents are for which digital signatures are accepted, the more dramatic the representation problem becomes. The problem is solved if a first person sees what she signs. For this purpose, a first person's PSE needs a display. One possibility is to use a cellular phone as a PSE. But its display is very small. Hence, the documents that can be signed securely on it are rather short. It depends on the solution of the representation problem whether digital signatures can be used to replace handwritten signatures.

Certification Authorities

If a first person uses a public-key system, it is not sufficient for first person to keep first person own private keys secret.

If second person uses the public key of another person, must be sure, that it is really second person's key. If the attacker, third person is able to substitute his own public key for Person's public key, then third person can decrypt secret messages to second person and third person can sign documents in second person name.

One solution of this problem is to establish trusted authorities. Each user is associated with such a certification authority (CA). The user trusts his CA. With its signature,, the, C A certifies the correctness and validity of the public keys of its users, The users know their CA's public key. Therefore, they can verify the signatures of their CA.

Registration

If a person becomes a new user of the public-key system, then he/she is registered by his CA. He/she tells the CA his/her name and other relevant personal data. The CA verifies Person's information. A Person can, for example go to the CA in person and present some, identification. The CA issues a user name for a person that is different from the user name of all other users in the system. Person will use this name for example, if he/she signs documents. If a person wants to keep his/ her name secret, then he/she may use a pseudonym. Then, only the CA knows Person’s real name.

Key generation

A Person's public and private keys are generated either in his PSE [4] or by his CA. It is recommended that Person not know his private keys, because then he cannot inform others about those keys. The private keys are stored in Person's PSE. The public keys are stored in a directory of the CA. Clearly, the keys must be protected while they are communicated between Person and his CA.

For each purpose (for example, signing, encryption, and identification), a separate key pair is required. Otherwise, the system may become insecure. This is illustrated in the next example.

Example 1: If a first person uses the same key pair for signatures and challenge response authentication, then an attack can be mounted as follows. Other persons pretends that he wants to check A first person's identity. As a challenge, he sends the hash value h(m) of a document m. A first person signs this hash value, assuming that it is a random challenge. But in fact a first person has H document, which was chosen by second person without noticing.

Certification

The CA generates a certificate, which establishes a verifiable connection between Person and his public keys. This certificate is a string, which is signed by the CA and contains at least the following information:

1. the user name or the pseudonym of Person,
2. Person's public keys,
3. the names of the algorithms in which the public keys are used,
4. the serial number of the certificate,
5. the beginning and end of the validity of the certificate,
6. the name of the CA,
7. restrictions that apply to the use of the certificate.

The certificate is stored, together with the user name, in a directory. Only the CA is allowed to write in this directory, but all users of the CA can read the information in the directory.

Archive

Depending on their use, keys in public-key systems must be stored even after they expire. Public signature keys must be stored as long as signatures generated with those keys must be verified. The CA stores certificates for public signature keys. Private decryption keys must be stored as long as documents were encrypted using those keys must be readable. Those keys are stored in the PSEs of the users. Authentication keys, private signature keys, and public encryption keys need not be put in archives: They must be stored only as long as they are used for authentication, generating signatures, or encrypting documents.

Initialization of the PSE

After a Person has been registered and his keys have been generated and certified, the CA transmits private keys to his CA, if they have been generated by the CA. The CA may also write its own public key and a Person's certificate to the PSE.

Directory service

The CA maintains a directory of all certificates together with the name of the owner of each certificate. If a first person wants to know Person's public keys, she asks her CA whether second Person is one of its users. If Person is registered with A first person's CA, then a first person obtains Person's certificate from her CA's directory. Using the public key of her CA, a first person verifies that the certificate was in fact generated by her CA. She obtains the certified public keys of Person. If Person is not a user of a first person's CA, then A first person can obtain his public keys from another CA. This is explained below.

A first person may keep in her CA certificates that she frequently uses. However, she must check regularly whether those certificates are still valid.

If a CA has many users, access to its directory may become very slow. It is then possible to keep several copies of the directory and to associate each user with exactly one copy.

Example 2: An King Fisher company wants to introduce a PKI for its 1 million employees in 10 countries. The company only wants to maintain one CA. In order to make access to its directory more efficient, the CA distributes of its directory to the 10 countries. Those copies are updated five a day.

Key update

All keys in a public-key system have a certain period of validity. Before a key expires, it must be replaced by a new, valid key. This new key is exchanged between the CA and the users in such a way that it does not become insecure even if the old, invalid key becomes known.

The following key update method is insecure. Shortly before Person's key pair becomes insecure, Person's CA generates a new key pair. It encrypts the new private key using Person's old public key and sends it to Person. Person decrypts that key using his old private key and replaces the old private key with the new one. If the attacker, a person, linds tin-old private key of Person, then he can decrypt the message of the CA to Person that contains the new private key. Thus, hr can find person’s new private key if he knows the old one. The security of the new private key depends on the security of the old one. This makes no sense. Instead, variants of the Diffie-Hellman key-exchange protocol can be used that avoid the man-in-the-middle attack.

Revocation of certificates
Under certain conditions, a certificate must be invalidated although it is not yet expired.

Example 3: On a train trip, Person has lost his ATM (or) Aadhar card. It contains Person's private signature key, which he can no longer use for signatures since this private key is nowhere but on the ATM (or) Aadhar card. Therefore, Person's certificate is no longer valid since it contains the corresponding verification key. The CA must invalidate this certificate.

The CA collects the invalid certificates in the certificate revocation list (CRL). It is part of the directory of the CA. An entry in the CRL contains the serial number of the certificate, the date when the certificate was invalidated, and possibly further information, such as the reason for the invalidation. This entry is signed by the CA.

Access to expired keys
Expired keys are kept in the CA's archive and can be provided by the CA upon request.

Example 4: The CA changes the signature keys of its users each month. Person orders a new vehicle from A first person and signs this order. But three months later, Person denies that he ordered the vehicle, A first person wants to prove that the order was actually signed by Person. He requests Person's old public verification key from the CA. This key is kept in the archive since it is out of date.

Certificate Chains
If Person and A first person do not belong to the same CA, then A first person cannot obtain the public key of Person[5] from the directory of her own CA but can obtain Person's public key indirectly.

Example 5: A first person is registered with a CA in India. Person is registered with a CA in the America. Hence, A first person knows the public key of her Indian CA but not the public key of Person’s AC. Now A first person obtains a certificate for the public key of Person’s CA from her own CA. she also obtains Person’s certificate either directly from Person or from his CA. using the public key of Person’s CA, which, in turn, is certified by her own CA, A first person can verify that she obtained a valid certificate for Person.

As described in Example 5, A first person can use a certificate chain to obtain Person’s authentic public key, even if Person and A first person belong to different CAs. Formally, such a chain can be described as follows. For a certification authority CA and a name U, denote by CA {U} the certificate that certifies the public key of U. Here, U can either be the name of a user or the name of another certification authority. A certificate chain that for A first person certifies the public key of Person is a sequence.

$$CA_1\{CA_2\}, CA_2\{CA_3\},\ldots\ldots\ldots, CA_{k-1}\{CA_k\}, CA_k\{\text{Person}\}$$

In this sequence, CA_1 is the CA where A first person is registered. A first person uses the public key of CA_1 to verify the public key of CA_2, she uses the public key of CA_2 to obtain the authentic public key of CA_3, and so on until she finally uses the public key of CA_k to verify the certificate of Person.

This method only works if trust is transitive (i.e., if U_1 trusts U_2 and U_2 trusts U_3, then U_1 trusts U_3).

REFERENCES

1. Claude E. Shannon, Communication theory of secrecy systems. Bell systems Technical Journal, 28(4):656-715, 1949.
2. J.Buchmann. Faktorisierung gober zahlen. Spektrum der Wissenschaften, 9:80-88, 1996.
3. J.Buchmann and H.C.Williams. Quadratic fields and cryptography. In J.H.Loxton, editor, number theory and cryptography, volume 154 of London mathematical society lecture note series, pages 9-25. Cambridge university press, Cambridge, England, 1990.
4. J.Buchmann and S.Paulus. A one way function based on ideal arithmetic in number fields. In B.Kaliski, editor, Advances in Cryptology- Crypto 97, volume 1294 of lecture notes in computer science, pages 385-394, Berlin 1997. Springer-Verlag.
5. Probability & statics, S. Chand company ltd. http//: info@schand group.com.

Source of support: Nil, Conflict of interest: None Declared