IDENTITY-BASED KEY INFRASTRUCTURES

Yvo Desmedt and Mike Burmester
Department of Computer Science
Florida State University
Tallahassee, FL 32306-4530, USA
desmedt,burmester@cs.fsu.edu

Abstract

Public Key Infrastructures (PKIs) provide a means by which trust in public keys can be established and managed. The trust is supported by a directory of certificates and/or proofs. However dealing with trust issues in such directories can be highly complex. A different approach to trust management was proposed in 1984 by Adi Shamir. This is based on identity-based cryptography (IBC). With IBC the public key of an entity *is* the identity of the entity. Entities do not have to "select" a public key and then get it certified by a Certifying Authority; instead they use their own identity as public key. A trusted center provides each entity with a corresponding secret key. Several identity-based cryptosystems have been proposed, including recently an Identity Based Encryption scheme based on Weil pairing.

In this paper we analyze Shamir's identity-based concept as a tool for managing trust of public keys. We argue the there is still a need for a trust infrastructure, which we call an Identity-based Key Infrastructure (IKI). We compare this infrastructure to that of a PKI and show that, essentially, IKIs are as complex as PKIs.

Keywords: PKI, trust infrastructures, identity-based cryptosystems.

Introduction

The fundamental goal of a Public Key Infrastructure (PKI) is to provide a means by which trust in public keys can be established and managed within a system or across domain boundaries. The trust should be built on real world trust relationships and is usually based on a directory of certificates, or more generally proofs, or a combination of these. Solving directory management issues is the key to the security and interoperability of PKIs.

As early as 1978, Kohnfelder [10] observed the importance of secure key management for public key cryptosystems. Pioneering work in the mid 1980s led to the standarization of the X500/X509 certificates directory (see e.g. [13, 16]). This particular directory supports a hierarchical infrastructure. Several alternative infrastructures followed, varying from "anarchic" infrastructures to highly inter-connected infrastructures. In particular, Zimmerman proposed the PGP infrastructure [18] and later Reiter-Stubblebine and independently Burmester-Desmedt-Kabatianski [14, 4] (see also [3]) proposed variants in which the trust is established via multi-connected web relationships.

A different approach for managing the trust in public keys was proposed in 1984 by Adi Shamir. This is based on the concept identity-based cryptography (IBC) [17]. In IBC the public key of an entity is its identity. Entities do not have to "select" a public key, and then get it certified by a Certifying Authority; instead they use their identity as a public key. A trusted (registration) center will then provide each entity with a corresponding secret key. Several identity-based cryptosystems have been proposed, including recently an Identity Based Encryption scheme based on Weil pairing [2].

In this paper we analyze Shamir's identity-based concept as a tool for managing trust in public keys. We argue the there is still the need for a trust infrastructure, which we call an Identity-based Key Infrastructure (IKI). The IKI is required to establish trust in the public identity keys (the secret keys are only issued to entities whose identity has been properly checked) and to manage these (for key revocation). We then compare both infrastructures and show that IKIs are essentially as complex as PKIs, at least with regards to key revocation.

The paper is organized as follows. In Section 1 we discuss the background and our notation. In Section 2 we show that with identity-based cryptography we need a basic Identity-based Key Infrastructure (IKI), to check the identity of each entity before a secret key can be issued. In Section 3 we deal with key management issues, in particular with key revocation in IKIs (for lost/stolen keys or more generally keys that have to be revoked). In Section 4 we discuss robustness issues and conclude with general remarks. We now start by reviewing Public Key Infrastructures and the concept of identity-based cryptography.

1. Background and notation

While in a conventional (symmetric) cryptosystem the sender and receiver use the same key, which must be kept secret, in a public key cryptosystem there are two keys: a public key and a private key. The first

is made public, while the second must be kept secret. However, public keys must be authenticated: that is, there must be proof of, or at least trust in, a link between the public key and its owner. Kohnfelder [10] was the first to observe the need for some kind of trust infrastructure for public keys. A Public Key Infrastructure is a trust infrastructure which authenticate public keys (links public keys to their owners). PKIs are supported by a distributed directory of certificates, or proofs, or more generally a combination of these. The most popular PKIs are based on the X500/X509 standard [9]. Public-key certificates have two parts: data and a signature. The data contains: a serial number, the public key of the entity, the issue date, the expiration date and additional fields (for relevant details). The signature is a digital signature on the data by a certifying entity. In the case of the X500/X509 directory, the certificates are issued by a Certification Authority (CA). The X500/X509 directory has a hierarchical infrastructure, i.e. a rooted tree of certifying authorities. The *Root* is called a Root Certification Authority (RCA). The public key of the RCA is known *a priori* to all users, and this knowledge is used to induce confidence in the public keys of CAs who authenticate the public key of the user. More than one tree may be used.

As pointed out earlier the cost of maintaining a secure PKI is a major issue particularly with regards to interoperability. To be of any use, the identities of the entities must be properly verified. Trust must be based on real world trust relationships, and cannot be established remotely (over the internet). The natural question to ask is whether one can use an infrastructure with weaker trust relationships. The following from a web page of Microsoft [8] tells us that there is no simple solution:

> In mid-March 2001, VeriSign, Inc., advised Microsoft that on January 29 and 30, 2001, it issued two VeriSign Class 3 code-signing digital certificates to an individual who fraudulently claimed to be a Microsoft employee. The common name assigned to both certificates is "Microsoft Corporation". The ability to sign executable content using keys that purport to belong to Microsoft would clearly be advantageous to an attacker who wished to convince users to allow the content to run.

So, it is no surprise that there are concerns in the security community regarding the weaknesses and the potential security risks which ill-defined PKIs can lead to [6].

In 1984 Adi Shamir [17] proposed a different approach, based on identity-based cryptosystems (IBC). With these systems the public key of an entity is simply the identity of the entity. The corresponding secret key is computed by a trusted center using an algorithm f with master (secret) key K. Several identity-based digital signatures systems have been proposed (see e.g., [13, 16]) and recently an identity-based encryption system based on Weil pairing has been proposed (see e.g., [2]).

Let ID_u be the identity of entity u. Evidently, this must be unique. So it may have to contain other information besides what we call the *natural ID*, that is, the first name(s) and the last name of the entity. The secret key of u will be $f(ID_u, K)$. At a first glance it may seems that this approach removes the need for a public key infrastructure. The goal of this paper is to show that this is not the case.

First observe that $f(ID_u, K)$ may not exist.[1] To deal with such cases Shamir proposed appending to ID_u a short binary string j_u, such that $f(ID_u||j_u, K)$ will exist ($||$ indicates concatenation). The string j_u could be computed using deterministic exhaustive search. So, the secret key of the user is $SK_u = f(ID_u||j_u, K)$, which is provided to the user by the trusted center. If $f(ID_u, K)$ always exists, i.e. for each string ID_u, then it seems that there is no need for j_u. However, in Section 3 we will argue that we still need j_u for key revocation.

The secret key SK_u of an identity-based cryptosystem corresponds to the secret key of a public key encryption or digital signature cryptosystem. The identity key $ID_u||j_u$ corresponds the public key. Indeed, an identity-based cryptosystem is a particular case of a public key cryptosystems. In particular, to send privately a message M to user u, the sender will send the ciphertext $C = E_{ID_i||j_u}(M)$ and to decrypt, the receiver u will compute $D_{SK_u}(C)$. To digitally sign the message M the user u will compute the signature $sign = Sign_{SK_u}(M)$ and to verify its correctness the receiver will compute the Boolean verification function $V_{ID_u||j_u}(M, sign)$.

We conclude this section by observing that it is not necessary to endow all the trust to one Key Distributing Center: by using secure distributed computation techniques (see e.g. [7, 1, 5]) the trust can be distributed.

2. Establishing trust in the identity keys

We shall discuss the *management* of trust in the public identity keys of an identity-based cryptosystem (key revocation) in Section 3. In this section we focus on the *establishment* of trust.

As pointed out earlier it is essential that the trust is based on real world relationships and that it is adequately checked. In particular identity verification cannot be done remotely (over the internet) and must be properly checked prior to a secret key being issued. Indeed, if a third party can fraudulently claim to be entity u, then it can impersonate u and sign or decrypt in the same way as u does. Since identity-verification

[1] For example, if the function f corresponds to computing the square root of ID_u modulo n and K is the prime factorization of n, then it will not exist if ID_u is a quadratic nonresidue modulo n.

cannot be done remotely, it has to be delegated to trusted Local Registration Centers. We now describe the steps that need to be taken at the stage of "registration". From these steps it will become clear that we need an adequately managed trust infrastructure.

1 The Local Registration Center must first verify the identity ID_u of entity u in person, checking as much evidence as possible. The required evidence may be a birth certificate, a passport (and possibly old passports), a driver's license, witnesses, etc. If the entity is an organization, such as a company, the Local Registration Center must verify that all the representatives of this organization are authorized to represent it. The secret key will be computed using the name of the organization. Local laws may differ from country to country on who can legally represent an organization.

2 The Local Registration Center must verify that the submitted ID_u has not yet been assigned a secret key. Otherwise more than one entities will be assigned the same secret key and any one of these can impersonate the others. Note that since public keys are significantly longer than identities, and much more random in nature, the probability that two entities have the same public key is negligible, and so can be ignored. For identities however, the situation is very different. The solution to this problem now depends on whether the format of ID_u is:

Variable length. The entity can append other relevant information, which may depend from country to country[2].

Fixed length. (This is the case with the login name for many operating systems.) In this case if:

The entity's first/last name(s) are sufficiently long: the entity could combine parts of his/her first name(s) with parts of the last name to obtain a compact unique ID. In the case of a organization, there is evidently no first name.

The entity's names are not sufficiently long: we basically have a combination of fixed and variable length encodings. The entity needs to append to his/her/its name other relevant information and/or compact it.

[2]Although Shamir [17] actually suggested to add such information as social security numbers, in some countries such as the US this is a bad idea since a social security number is sometimes used as a password for credit transactions.

Evidently one could use a different j_u to make the identity unique, but we will need to use j_u for another purpose, as we shall see in Section 3.

The issue of checking the uniqueness of the ID causes several problems:

- The ID_u is no longer public information in the sense that any third party will *not* know a priori all the information appended or the compacted string. In the case of variable length encoding, one can evidently try to append known information. For example, one could append the affiliation of the entity, or the network provider, etc. However, this also implies that a new secret key will be required when the affiliation or the network provider changes.

  Note that with public key cryptosystems which are not identity-based, one also needs to uniquely identify the entity. However there is a major difference. If a entity has different roles in society, and is therefore known under different affiliations, the different IDs of an individual in the PKI database could all point to the *same* public key. In identity-based cryptosystems, these different affiliations will give rise to different secret keys. With small handheld/handless devices, this may cause memory problems.

- A trust infrastructure is needed. Indeed, before a secret key is issued, one must make certain that ID_u is globally unique.

  Note that if the identity contains a (work related) affiliation, it is much simpler to organize the checking of identities. First the Local Registration Center must make sure that all affiliations have a unique representation. Then it is up to the Local Registration Center to make certain that within this organization the name is unique. However, there is no guarantee that the natural identity of the entity can be used within this organization. Indeed, certain first names and last names are so popular that it is not uncommon to find two people in the same organization with the same first name and last name. Evidently, if one replaces the affiliation with entities such as the network provider, etc, the same applies.

  This check may imply a time delay. If ID_u is not unique, then the entity is contacted and a new ID_u is suggested. Evidently an alternative solution would be to provide several names in advance. The time delay can, for all practical purposes, be eliminated if the entity can reserve in advance a name,

e.g. over the internet. This gives the entity sufficient time to find appropriate strings to append or ways to compact his/her/its identity, before making the final choice. Note that even if such an identity is reserved, the entity will still need to demonstrate in person to the Local Registration Center the validity of his/her/its identity before receiving a secret key.

3 The entity will need to obtain the secret key SK_u privately and in an authenticated way. This introduces a *key distribution problem*, in particular if the entity does not wants the Local Registration Center to see the secret key SK_u. The solution is to provide a temporary key that a Key Distribution Center can use to encrypt the secret key SK_u. This center must possess the Master key K and will use it to compute SK_u. For simplicity, we focus on the case when SK_u is encrypted. In this case the entity u selects a temporary Public Key, Secret Key pair: (TPK_u, TSK_u), for the encryption of SK_u.

4 At this stage the Local Registration Center can forward to the Key Distribution Center a request for the entity with unique identity ID_u to obtain a secret key SK_u. The request, signed by the Local Registration Center, will contain at least:

   (a) the unique identity ID_u, and

   (b) the temporary (public) key TPK_u that will be used by the Key Distribution Center to encrypt the secret key SK_u for u.

5 The Key Distribution Center can now compute the string j_u and the secret key SK_u, and send to entity u the digitally signed encryption:

$$(\mathcal{E}_{TPK_u}(SK_u), ID_u||j_u;\ Sign_{SK_{KDC}}(\mathcal{E}_{TPK_u}(SK_u), ID_u||j_u))$$

   where $\mathcal{E}$ is a public key encryption algorithm.

6 The Key Distribution Center notifies the other centers of the infrastructure about the identity key: $ID_u||j_u$.

3. Managing trust in the identity keys

The registration phase already introduces the need for an infrastructure. In certain circumstances this infrastructure can be kept relatively simple.

From our earlier discussion, it seems that a major difference between a Public Key Infrastructure and a Identity-based Key Infrastructure is

that with IKIs one only needs the infrastructure for the registration phase. In a PKI if a third party, say Bob, wants to find the public key of Alice, he has to get it from the infrastructure or directly from Alice.[3] Such an interaction does not seem necessary with IKIs. There are two problems with this reasoning:

1 If ID_A is different from the natural identity of Alice, then there will be at least two entities with the same natural identity. So if Bob wants to use ID_A, e.g. to send an encrypted message, then Bob must first find out the correct identity of Alice. If this is not natural, then Bob must consult an infrastructure and evidently, the reply given must be authenticated (signed).

   However, it would be incorrect to conclude that Bob only needs to consult the infrastructure when ID_A is not unique, because Bob may not know this. Consulting a WWW page about this, also requires that this WWW page is authenticated.

2 If keys are stolen or lost, then continuing to use the old $ID_A||j_A$ of Alice, clearly undermines the security. In the case of a public key cryptosystem, the entity has to just provide the Public Key Infrastructure with a new public key, in person (and of course provide adequate evidence of his/her/its identity). However, providing a new ID_A with Identity-based Key Infrastructures may be undesirable, particularly when ID_A is equal to the natural ID of the Alice. So, in this case, a solution would be to use a new j_A. Consequently, to deal with revocation, one has to use a similar approach to that for Public Key Infrastructures.

In conclusion we remark that, to deal with non-uniqueness of natural identities ID_u and the use of the string j_u, we require a structure that is similar to that of Public Key Infrastructures. We have call this, an Identity-based Key Infrastructure (IKI). In this infrastructure, one could regard the Local Registration Centers as the Certifying Authorities of a Public Key Infrastructure. Although their duties in an IKI are different, they are very similar. For example, the IKI also needs to be consulted by a third party before it will use (for the first time) an identity $ID_A||j_A$. Moreover, to deal with lost and stolen secret keys, one needs a revocation mechanism. Similar mechanisms to those for Public Key Infrastructures can be used, such as the typical off-line revocation lists or the more recent on-line approach [15] (see also [11]).

[3] We view PGP as a Public Key Infrastructure.

4. Conclusion

In this paper we argue that to deploy identity-based cryptography in a secure way, one needs to use an infrastructure that may be as complex as that of a Public Key Infrastructure. This is primarily a consequence of the fact that the natural identity (first name, last name) of an entity may not be unique. Revocation aggravates the problem.

If the identity of entities is not properly verified by some CAs, as in the case of VeriSign mentioned earlier, then these CAs are untrustworthy. Moreover, if on-line revocation is used, there is the risk of hackers can break into (the computer systems) the root CA [11], making the whole infrastructure untrustworthy. To deal with this problem, PKIs with a more robust infrastructure have been proposed [18, 14, 4, 3]. With these, multiple vertex disjoint trust-paths are used to certify public keys.[4] Robustness for IKIs can be achieved using the same approach as in the case of PKIs.

To conclude, we have compared the public key infrastructures of public key cryptosystems and identity-based cryptosystems. Except for the case when one only wants a low security level or, when one can guarantee that secret keys will not be lost or stolen and that the natural identities uniquely identify the entity, there is a need for an Identity-based Key Infrastructure. Its role is very similar to that of a Public Key Infrastructure. However, one has to deal with a key distribution problem, i.e. how can the Key Distribution Center give the secret key SK_A to Alice in a secure way. This problem can be solved using a public key encryption system with a temporary public key. This roughly doubles the hardware/software needs. With a handheld/handless device, the secret key may have to be uploaded securely from a PC that runs public key software.

Finally, we note that $ID_u||j_u$ may be shorter than the pair: (identity information, public key). This may be the only real advantage of identity-based cryptography when used in secure environments that, as we argued, need an Identity-based Key Infrastructure (scalability, particularly with revocation lists can be a problem, see [12]). Evidently, a well known disadvantage is that the Key Distribution Center knows each entity's secret key.

[4] In the case of PGP these are not necessarily "Authorities," but could be just friends. The VeriSign example suggest that one could allow for "self proclaimed authorities."

Acknowledgments

Part of this research was inspired by Cisco CIAG Research Wishlist, in particular by the topic: "The Internet Without PKI, What are the Alternatives"? The first author thanks Tanja Lange (Ruhr Universität Bochum, Germany) and Roberto Avanzi (University of Duisburg-Essen, Germany) for some discussions related to the topic of identity based cryptography.

References

[1] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non cryptographic fault-tolerant distributed computation. In *Proceedings of the twentieth annual ACM Symp. Theory of Computing, STOC*, pp. 1–10, May 2–4, 1988.

[2] D. Boneh and M. Franklin. Identity based encryption from the Weil pairing. In *Advances in cryptology – Crypto '2001*, volume 2139 of *Lect. Notes Comput. Sci.*, pp. 213–229. Springer, 2001.

[3] M. Burmester and Y. Desmedt. Hierarchical public-key certification: The next target for hackers? Submitted October 2001 to Communications of the ACM, accepted February 21, 2003.

[4] M. Burmester, Y. Desmedt, and G. Kabatianskii. Trust and security: A new look at the Byzantine generals problem. In R. N. Wright and P. G. Neumann, editors, *Network Threats, DIMACS, Series in Discrete Mathematics and Theoretical Computer Science, December 2–4, 1996, vol. 38*. AMS, 1998.

[5] D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols. In *Proceedings of the twentieth annual ACM Symp. Theory of Computing, STOC*, pp. 11–19, May 2–4, 1988.

[6] C. Ellison and B. Schneier. Ten risks of PKI: What you're not being told about Public Key Infrastructure. *Computer Security Journal*, 16(1), pp. 1–7, 2000. See also `http://www.counterpane.com/pki-risks.html`.

[7] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proceedings of the Nineteenth annual ACM Symp. Theory of Computing, STOC*, pp. 218–229, May 25–27, 1987.

[8] Microsoft Security Bulletin MS01-017 (version 2.0) – Erroneous VeriSign-Issued Digital Certificates Pose Spoofing Hazard. http://www.auscert.org.au/render.html?it=1238&cid=1

[9] ITU-T Rec. X.509 (Revised), "The Directory - Authentication Framework", International Telecommunication Union, Geneva, Switzerland, 1992 (equivalent to ISO/IEC 9594-8:1995).

[10] L. M. Kohnfelder. Toward a practical public-key cryptosystem, BSC-thesis, MIT Department of Electronical Engineering, 1978.

[11] P. McDaniel and A. Rubin. A response to "can we eliminate certificate revocations lists?". In Frankel Y, editor, *Financial Cryptography, 4th International Conference, Proceedings (Lecture Notes in Computer Science 1962)*, pp. 245–258. Springer-Verlag, 2000. Anguilla, British West Indies, February 20–24.

[12] S. Micali. Novomodo. *Proceedings, 1st Annual PKI Workshop*, Gaithesburg, Maryland, pp. 15–26, 2002.

[13] A. Menezes, P. van Oorschot, and S. Vanstone. *Applied Cryptography*. CRC, Boca Raton, 1996.

[14] M. K. Reiter and S. G. Stubblebine. Path independence for authentication in large scale systems. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pp. 57–66, April 1997. Zurich.

[15] R. L. Rivest. Can we eliminate certificate revocations lists? In R. Hirschfeld, editor, *Financial Cryptography, 2nd International Conference, Proceedings (Lecture Notes in Computer Science 1465)*, pp. 178–183. Springer-Verlag, 1998. Anguilla, British West Indies, February 23–25.

[16] B. Schneier. *Applied Cryptography*. J. Wiley, New York, second edition, 1996.

[17] A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology. Proc. of Crypto 84 (Lecture Notes in Computer Science 196)*, pp. 47–53. Springer-Verlag, 1985. Santa Barbara, California, U.S.A., August 19–22.

[18] P. R. Zimmermann. *The Official PGP User's Guide*. MIT Press, Cambridge, Massachussets, 1995.