IDENTITY-BASED KEY INFRASTRUCTURES (IKI)

Yvo Desmedt,[1,2] and Mike Burmester,[2]

[1] *Department of Computer Science*
University College London, UK

[2] *Department of Computer Science*
Florida State University, USA
burmester@cs.fsu.edu

Abstract Kohnfelder realized in 1978 that public key schemes require a Public Key Infrastructure (PKI). X500/X509 were set up to standardize these ideas. PGP, proposed by Zimmermann is an alternative to the original PKI idea. Variants of the PGP based PKI were discussed independently by Reiter-Stubblebine and Burmester-Desmedt-Kabatianskii.

The goal of Shamir's 1984 idea of "identity-based" cryptography was to avoid a Public Key Infrastructure. Instead of having the users have their own public key, the identity of the user is the "public key," and a trusted center provides each party with a secret key. Several identity-based cryptosystems have been proposed, in particular recently.

We analyze Shamir's identity-based concept critically. We argue the need for at least a registration infrastructure, which we call a "basic Identity-based Key Infrastructure." Moreover, if secret keys of users can be stolen or lost, the infrastructure required to deal with this is as complex as the one of PKI. We make further comparisons between public key systems and identity-based ones.

Keywords: PKI, trust infrastructures, identity-based cryptosystems

INTRODUCTION

While in conventional cryptosystems the sender as well as the receiver's key must remain secret, in public key systems the privacy of one of the keys is not essential, and so can be made public. However, public keys must still be authenticated. Kohnfelder Kohnfelder, 1978 observed the need for some Public Key Infrastructure to authenticate the public key of a user. A lot of work has been done on Public Key Infrastructures

Y. Deswarte et al. (eds.), *Security and Protection in Information Processing Systems*

since then. X500 and X509 standards were developed (see e.g. Menezes et al., 1996; Schneier, 1996).

Public-key certificates have two parts: data and a signature. The data contains information about the identity of an entity, the public-key, the validity period and other relevant details. The signature is a digital signature on the data by a certifying entity. For the X509 certificates this is a Certification Authority (CA). With X509, the infrastructure is hierarchical, i.e. a rooted tree. The *Root* is a certification authority, called 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 can be used.

Alternatives to X509 have been proposed in PGP Zimmermann, 1995. Alternatives that make the infrastructure more robust to make it less vulnerable to attacks by outsiders, and maintain security when CAs are sloppy, were suggested Reiter and Stubblebine, 1997; Burmester et al., 1998 (see also Burmester and Desmedt,).

Several countries in South-East Asia are in the process of setting up national PKIs. A problem with PKI is the cost to run a proper PKI. To be of any use, the identity of the party must be properly verified. This cannot be done over the internet! The natural question is whether one can avoid using a public key and a PKI using alternatives.

From a web page of Microsoft Microsoft Security Bulletin MS01-017, 2003 we learn:

> 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 some question PKIs Ellison and Schneier, 2000.

In 1984 Shamir Shamir, 1985 suggested the concept of identity-based cryptosystem. In such system *the public key of the user is just the identity of the user.* To guarantee that this works, a trusted center will use a master (secret) key. Using this master key, the trusted center computes from the identity of the user u, the secret key of the user. It seems that this removes the need for PKI. The goal of this paper is to critically analyze this impression. Note that recently several identity-based cryptosystems have been proposed, e.g. by Boneh-Franklin Boneh and Franklin, 2001.

We will demonstrate in Section 2 that a basic Identity-based Key Infrastructure is at least required to verify the identity of the user before handing out the secret key. When keys can be stolen or lost, one needs to deal with revocation, which is discussed in Section 3. We conclude in Section 4 and also discuss robustness issues. We now start by reviewing the concept of identity-based cryptography.

1. BACKGROUND AND NOTATION

Shamir invented the notion of identity-based cryptography Shamir, 1985. The idea is that each individual could use his/her/its own identity as "public key." A Key Distribution Center would compute the secret key starting from the identity of the user, using an algorithm f and based on the Key Distribution Center's top secret master key, K. We denote the output of f as $f(\cdot,\cdot)$.

Note that using secure distributed computation (see e.g. Goldreich et al., 1987; Ben-Or et al., 1988; Chaum et al., 1988) the need to trust a single Key Distributing Center can be relaxed.

Let ID_u be the identity of the user u. Evidently, this must be unique. So, it may have to contain other information besides what we call the *natural ID*, being the first name(s) and the last name of the user.

To deal with the case $f(ID_u, K)$ does not[1] exist, Shamir envisioned using a short binary string j_u so that $f(ID_u||j_u, K)$ does exist, where $||$ indicates concatenation. 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 Key Distribution Center.

If $f(ID_u, K)$ always exists, i.e. for each each string ID_u, then it seems that there is no need for j_u. However, in Section 3 we will argue that in a real world situation this is false.

When using encryption and digital signatures, the secret key SK_u plays the same role as the secret key in public key systems. $ID_u||j_u$ plays the same role as the public key does in public key systems. So, to privately send 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 $V_{ID_u||j_u}(M, sign)$.

2. THE NEED FOR A BASIC IDENTITY -BASED KEY INFRASTRUCTURE

The impact of lost or stolen secret keys on the concept of identity-based cryptography is only discussed in details from Section 3 on. So, in

this section only, we assume, hypothetically, that secret keys are never lost or stolen.

As well known by the information security community, it is essential in PKI, that the identity of the user be properly checked. The aforementioned example of VeriSign illustrates this. For the same reason is it required that the Key Distribution Center verifies the identity of the user requesting a secret key. Indeed, if a third party can fraudulently claim to be user u, then it can pretend being the user u, i.e. decrypt/sign as user u.

Since, the identity verification cannot be done remotely (so not over the internet), there is a need for what one could call a Local Registration Center. We now stipulate the steps that need to be taken at the stage of "registration." This will demonstrate the need for a true infrastructure.

1 The Local Registration Center needs to first of all verify the identity ID_u of the user in person, checking as much evidence as possible. The required evidence may be a birth certificate, a passport (and possibly old passports[2]), a driver's license, witnesses, etc.

   In the case the user is an organization, such as a company, and the user would like to obtain a secret key for the entity, one needs to verify that the people who represent this organization have the authority to do so and to obtain a secret key in the name of the organization. Local laws may differ from country to country on who can legally represent an organization.

   Note, if one does not trust the Local Registration Center, multiple could be used, in a similar way as pointed out in Burmester et al., 1998; Reiter and Stubblebine, 1997 (see also Burmester and Desmedt,).

2 One needs to verify that the proposed ID_u has not yet been assigned. Indeed, otherwise, (at least two) users may have the same secret key and the one can pretend to be the other. Note that since public keys are significantly longer and much more random in nature than identities, the probability two users have the same public key is almost zero, and so can be ignored. For identities, the situation is very different. The solution to this problem now depends on whether the format of ID_u is:

   variable length encoding. The user could append other relevant information, which may depend from country to country[3].

   fixed length encoding. (This is for example the case with the login name on many operating systems.) In this case if:

the user's name are sufficiently long the user 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 corporation, there is evidently no first name, etc.

the user's names are not sufficiently long we basically fall in a combination of the variable length encoding case with the fixed length encoding. The user needs to append to his name and/or compact the string.

Evidently, one could use j_u to make the identity unique, but, j_u will need to be used for other purposes, as we discuss 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 this person, or the network provider used, etc. However, this also implies that a new secret key will be required when the affiliation or network provider changes.

  Note that if a public key system is used, one also needs to uniquely identify the user to find the public key of that user. However, there is a major difference. If a user plays different roles in society, and is therefore known under different "affiliations," the different IDs in the PKI database[4] related to the same individual, could all point to the *same* public key. In identity-based cryptosystems, these different "affiliations" give rise to different secret keys. If the user relies on small handheld/handless devices, this may cause lack of memory problems.

- The need to check the uniqueness of ID_u, implies the need for an infrastructure. Indeed, before a secret key is issued, one must make certain it is globally unique.

  Note that if the identity contains a (work related) affiliation, the check for identity is much simpler to organize. First 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, this is no guarantee that the natural identity of the user can be used within this organization. Indeed, certain first names and last

names are so popular it is not uncommon to find two people in the same organization with the same (first name, last name). Evidently, if one replaces the affiliation with such entities as network provider, etc. the same comment applies.

This check may imply a time delay. If it is not unique, the user is contacted and a new ID_u is suggested. Evidently an alternative solution is to provide several names at once. The time delay can for all practical purposes be eliminated if the user can in advance "reserve" a name, e.g. over the internet. This gives the user plenty of time to think about good strings to append or how to compact the identity, before making the final choice. *Note that if such an identity is reserved the user 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 user will need to obtain the secret key SK_u privately and in an authenticated fashion. This introduces a *key distribution problem*, in particular if the user wants to avoid that the Local Registration Center learns the secret key SK_u. The solution is to provide a temporary key that the Key Distribution Center will use to encrypt the secret key SK_u. For simplicity, we focus on the case the user does not like that the Local Registration Center knows SK_u. In that case the user could make a Temporary Public Key TPK_u, secret key pair, it will use to receive its secret key SK_u.

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

   (a) the unique identity ID_u, and

   (b) the temporary (public) key TPK_u that will be used by the Key Distribution Center to provide securely the key to the user.

5 The Key Distribution Center can now compute the key and send it signed. So the user receives:

$$(E'_{TPK_u}(SK_u), Sign_{SK_{KDC}}(E'_{TPK_u}(SK_u)))$$

where E' is a public key encryption algorithm.

6 The Key Distribution Center informs the hierarchy about the identity key: (ID_u, j_u).

3. THE IMPACT OF REVOCATION AND NON-UNIQUENESS OF THE NATURAL ID

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

From our discussions until now, it seems that a major difference between a Public Key Infrastructure and a Identity-based Key Infrastructure is that in the last case one only needs the infrastructure for the registration phase. However, in Public Key Infrastructures, when a third party (e.g. Bob) wants to learn the public key of Alice, he/she/it needs to consult the infrastructure or have obtained the public key from Alice[5]. Such an interaction does not seem necessary for Identity-based Key Infrastructures. There are two problems with above reasoning:

1 If ID_A is different from the natural identity of Alice, then this means that there are at least two people who have the same natural identity. So if Bob wants to use ID_A, e.g. to send an encrypted message, then Bob needs to find out what the correct ID_A is. If that is not natural, then Bob needs to consult an infrastructure and evidently, the reply given by this infrastructure must be authenticated, so 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 to be authenticated.

2 If keys are stolen or lost, then continuing to use the old ID_A and j_A of Alice, clearly undermines the security. In the case of a public key system, the user could just provide in person (and his/her/its identity having been checked) the Public Key Infrastructure a new public key. However, providing the Identity-based Key Infrastructure a new ID_A may be undesirable, in particular, when ID_A is equal to the natural ID of the Alice. So, a solution to this issue, is that a new j_A is used.

   So, to deal with revocation, one needs a similar solution as in the case of Public Key Infrastructures.

So, to deal with non-uniqueness of natural identities and the use of j_u, a similar structure as used for Public Key Infrastructures is required, which we call Identity-based Key Infrastructure. In this context, one could call the Local Registration Centers: Certifying Authorities. Although their duties in an Identity-based Key Infrastructure are different,

these are sufficiently equivalent to call them Certifying Authorities. The Identity-based Key Infrastructure 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 as in the case of Public Key Infrastructures can be used, this means the typical off-line or the more recent on-line approach Rivest, 1998 (see also McDaniel and Rubin, 2000).

Note that Boneh-Franklin Boneh and Franklin, 2001 already mentioned the revocation problem. However, they assume an underlying PKI. We argue that to deal with revocation in identity based cryptosystems, we need a PKI like structure.

4. CONCLUSION

We argued that to securely deploy identity-based cryptography, one needs a structure as complex as Public Key Infrastructures. This is primarily a consequence of the fact that the natural identity (first name, last name) may not be unique. Revocation just aggravates the problem. So, in many circumstances, there is a need for an Identity-based Key Infrastructure.

The problem with not properly verifying the identity of users, as in the aforementioned case of VeriSign, implies that CA's may be untrustworthy. Moreover, if on-line revocation is used, there is the risk of hackers breaking into the CA McDaniel and Rubin, 2000, making CA also untrustworthy. To deal with this problem, more robust PKIs have been suggested Zimmermann, 1995; Reiter and Stubblebine, 1997; Burmester et al., 1998; Burmester and Desmedt, . In all of these Bob (potentially) uses more than one CA[6] to obtain the public key. Adding robustness can also be done for Identity-based Key Infrastructures.

To conclude, we compare public key systems with identity-based ones. Except in the case one only wants low security or, one can guarantee that secret keys will not be lost or stolen and that the natural identities uniquely identify the user, there is a need for an Identity-based Key Infrastructure. Its role is very similar to the one of a Public Key Infrastructure. However, one needs to deal with a key distribution problem, i.e. how can the Key Distribution Center securely provide the secret key SK_A to Alice. This can be solved using a public key system with a temporary public key, roughly doubling the hardware/software needs. In the case identity-based public key is used in a handheld/handless device, its 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. Evidently, a well known disadvantage is that the Key Distribution Center knows each user's secret key.

Acknowledgments

Part of this research was funded by NSF CCR-0209092 and was inspired by Cisco CIAG Research Wishlist 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.

Notes

1. For example, if f corresponds to computing the square root of ID_u modulo n and K is the prime factorization of n, then this may not always exist, since not all numbers modulo n are quadratic residues.

2. Certain countries return to their citizens old passports. To avoid that one attempts to reuse these, holes are punched through the passport or corners are cut off, etc.

3. Although Shamir Shamir, 1985 actually suggested to add such information as social security number, in some countries, such as the US, this is a bad idea, since a social security number is viewed as a global password! For example it allows an individual to get a loan.

4. If we were to use database terminology, this ID would be called the "key". However, in the case of a public key system, from a cryptographic viewpoint, this is not a key at all. Therefore, we avoided this confusing terminology.

5. We also view PGP as a Public Key Infrastructure.

6. In the case of PGP these are actually not "Authorities," but could be just friends, so one could object against the word CA. However, the VeriSign example questions whether such organizations are "Authorities," or just "self proclaimed authorities."

References

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

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

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

Burmester, M., Desmedt, Y., and Kabatianskii, G. (1998). Trust and security: A new look at the Byzantine generals problem. In Wright, R. N. and Neumann,

P. G., editors, *Network Threats, DIMACS, Series in Discrete Mathematics and Theoretical Computer Science, December 2–4, 1996, vol. 38.* AMS.

Chaum, D., Crépeau, C., and Damgård, I. (1988). Multiparty unconditionally secure protocols. In *Proceedings of the twentieth annual ACM Symp. Theory of Computing, STOC*, pages 11–19.

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

Goldreich, O., Micali, S., and Wigderson, A. (1987). How to play any mental game. In *Proceedings of the Nineteenth annual ACM Symp. Theory of Computing, STOC*, pages 218–229.

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

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

Menezes, A., van Oorschot, P., and Vanstone, S. (1996). *Applied Cryptography.* CRC, Boca Raton.

Microsoft Security Bulletin MS01-017 (March 22, 2001, updated: June 22, 2003). Microsoft security bulletin ms01-017, erroneous verisign-issued digital certificates pose spoofing hazard. http://www.microsoft.com/technet/security/bulletin/MS01-017.mspx.

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

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

Schneier, B. (1996). *Applied Cryptography.* J. Wiley, New York, second edition.

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

Zimmermann, P. R. (1995). *The Official PGP User's Guide.* MIT Press, Cambridge, Massachussets.