Which PKI (Public Key Infrastructure) is the Right One?
[Panel]

Carlisle Adams
Entrust Technologies
750 Heron Rd.
Ottawa, Ontario K1V 1A7,
Canada
carlisle.adams@entrust.com

Mike Burmester
Information Security Group
Royal Holloway – Univ. of
London
Egham, Surrey, TW20 OEX, UK
mikeb@dcs.rhbnc.ac.uk

Yvo Desmedt, *Moderator*
Dept. of Computer Science
Florida State University
PO Box 4530
Tallahassee, FL 32306, USA
desmedt@cs.fsu.edu

Mike Reiter
Bell Laboratories, Lucent Technologies
600 Mountain Ave.
Room 2A-342
Murray Hill, NJ 07974, USA
reiter@research.bell-labs.com

Philip Zimmermann
PGP
Menlo Park, California, USA
prz@pgp.com

ABSTRACT

Several organizations are setting up Public Key Infrastructures, examples are:

- the Corporation for Research and Educational Networking (CREN),
- the Federal Government plans to fund 7 Public Key Infrastructure Models pilot programs at different federal agencies.

However, experts have quite different viewpoints on how to set up such Public Key Infrastructure (PKI). Indeed, X500 and X509 are hierarchically organized (i.e. vertical), but PGP (see also Rivest-Lampson) is horizontally organized. Variants of PGP (see Reiter-Stubblebine (CCCS, ACM) and Burmester-Desmedt-Kabatianski (DIMACS)) require a minimum connectivity, i.e., a minimum number of disjoint paths in order to deal with hackers breaking into certifying entities (authorities). Moreover, Ellison-Schneier have questioned the need for a Public Key Infrastructure (PKI).

Before one builds such an expensive infrastructure, experts should debate what method to use and whether a PKI is needed. While a hierarchical PKI may become the next target of computer hackers, a multiple-connected one seems much more expensive to build.

1. SURVEY BY THE MODERATOR

1.1 Introduction

After the discovery of public key it was observed quickly that there is an authenticity problem. Indeed, there are two problems:

in the case of digital signatures, if a hacker can convince the receiver that a fake key, one made by the hacker, is the public key of the sender, then the receiver will be fooled into accepting as authentic documents created by the hacker.

in the case of encryption, if a hacker can convince the sender that a fake key is the public key of the receiver, then the hacker will be able to eavesdrop on the communication intended for that receiver.

The authenticity of the public key of a user can be established by using *certificates*. Seeing the importance of e-commerce, it is no surprise that several organizations are setting up Public Key Infrastructures. For example:

- the Massachusetts Institute of Technology (MIT) and the Corporation for Research and Educational Networking (CREN) propose to set up a hierarchical authentication structure for secure resource-sharing among higher education institutions [3].
- as a reaction to the jamming of such WWW addresses as yahoo.com, cnn.com, a meeting was set up at the White House. A list of key initiatives was set up [4]. One of those is the:

  > Funding 7 Public Key Infrastructure Models pilot programs in FY2001 at different federal agencies. ($7 million)

Although X500/509 is one of the first Public Key Infrastructures proposed, others have been suggested. Seeing the future importance of a Public Key Infrastructure (PKI), it is natural to wonder which Public Key Infrastructure is the best. We briefly survey now some of these alternative structures that have been proposed.

1.2 Several Public Key Infrastructure

1.2.1 Hierarchical Structures

In the case of the X500/X509 certificates the infrastructure is hierarchical, spanned by a tree with a Root Certification Authority (RCA). In this case the trust is centered at the root, and is transferred hierarchically to all the users in the network via Certification Authorities. More specifically, the public key of the RCA is known (a priori) to all the users, and this knowledge is used to induce confidence in the public keys of the

CCS'00, Athens, Greece.
ACM 1-58113-203-4/00/0011.

other entities via trust-paths. Observe that there is no need for the users to certify the public key of the RCA, because it is assumed that this key is known to all entities.

For applications in which it would be unreasonable to expect all the entities to trust the same RCA, one may use hierarchical authentication structures with several RCAs. The users are partitioned in domains, each one under the control of a single RCA, and the RCAs cross-certify their keys.

1.2.2 Trust graph

To understand the other PKIs we briefly discuss the concept of trust graph.

Certificates can be used to model the confidence of a network in its public keys by a directed *trust-graph* whose nodes correspond to the entities of the network (users and/or Certification Authorities) and edges to the certificates. Two nodes are joined by an edge if there is a certificate in which the entity of one node certifies (digitally signs) the public-key of the other. That is, if Alice certifies the public key of Bob then there is an edge from Alice to Bob (in some papers the arrow points the other way).

The confidence that an entity has in the public key of another entity (e.g., a CA) may be based on *direct* knowledge or, on *induced* knowledge. Certificates corroborate direct knowledge: an entity will only certify the public key of another entity if it believes that the key is authentic. Therefore the edges of the trust-graph reflect direct confidence. This confidence is established by non-cryptographic means (e.g., by checking personal details). Induced confidence is established via *trust-paths* that link nodes in the trust-graph. In a trust-path each node certifies the authenticity of the public key of the next node on the path. In this manner the trust is induced via the path. In particular, the first node in the path has induced confidence in the public key of the last node.

The trust-graph should be distinguished from the communication network because its edges correspond to trust relations, and are not necessarily communication paths. Furthermore, the nodes of the trust-graph may not be communication nodes. The following example illustrates this. Suppose that Bob is a good friend of Alice and that Alice knows the public key of Bob. Then Alice may be willing to certify Bob's key, even though Bob may be living far away, and there is no communication link between them.

1.2.3 PGP

Pretty Good Privacy (PGP) [10] is a freeware electronic mail system that uses an unstructured authentication framework. Users are free to decide whom they trust. PGP does not specify any specific structure for the trust-graph. A similar proposal was put forward by Rivest-Lampson [9].

1.2.4 Connectivity based

Reiter and Stubblebine [8] and independently Burmester-Desmedt-Kabatianski [1] proposed to use an approach similar as the one used in making networks reliable against Byzantine faults [5].
A trust-graph is ($2k$+1) *connected* if there are $2k$+1 node-disjoint trust-paths which connect any two nodes. Attacking such structures would require the penetration of more than k nodes.

If the trust-graph is known, then a public key is certified via $2k$+1 node-disjoint trust-paths [8, 1].

1.2.5 No PKI

Recently Ellison-Schneier argued that PKI have risks and so that it may be better not to have any. For more details see [6].

1.3 Security versus practicality

Several researchers have had serious security questions about X500/X509. The problem is that CA can be compromised at the organizational level or a hacker can penetrate those, as pointed out in [1, 8] and more recently in [6]. Indeed, the penetration of the Pentagon, the vandalizing of web pages such as the one of the FBI, the US Senate, etc., demonstrate that modern computers cannot be assumed to be secure.

Although PGP, addresses this issue to a certain extend, a user may have only asked one friend to certify his/her public key. This user is very vulnerable. In the connectivity based approach this problem is solved provided that:

- the graph is $2k$+1-connected,
- there are at most k nodes that can be hacked.

It seems however that requiring more connectivity reduces efficiency. Moreover, as was pointed out by Ellison-Schneier, many software packages cannot deal with the X500/X509 PKI, let alone a more complex one.

Certain figures seem to indicate that these PKIs will be expensive. Indeed, the Federal Government expects a cost of $1 million per federal agency to set those up [4]. It seems therefore logical to conclude that before one builds these PKIs, experts need to discuss whether these are needed and if so which one should be selected.

1.4 Other issues

A question is how to use the PKI and for what purposes. Ellison's concept of Simple-PKI focusses on the applications of PKI, in particular when a user is given an authorization, e.g., an authorization to edit a file. Ellison has argued that such authorization should be binded to the public key and not to the name of the individual. The argument is that a public key is more unique than the name of the individual associated with that public key, in particular when the name of the individual is very popular.

2. POSITION STATEMENT BY ADAMS

What does it mean to be the "right" PKI? It is possible to consider this from two perspectives: definition and application. According to the "definition" perspective, the "right" PKI must be the one that gives you the closest approximation to a true, or ideal, PKI. That is, not only must it utilize public-key technology, but it must also provide a real *infrastructure* (a pervasive substrate that provides a specific service to calling applications and devices through a simple, well-defined interface). This infrastructure must be understood by a large number of applications and environments, and must be entirely transparent (in its normal, day-to-day operation) to human users. Furthermore, the ideal PKI must provide full life-cycle management for its own primary tools (i.e., key pairs and corresponding certificates). Again, in a fashion transparent to the

human user, the infrastructure must incorporate mechanisms to handle key/certificate creation, use, and destruction, as well as techniques to allow certificates to be "turned off" when necessary, to be used appropriately beyond their validity period, and to provide extended trust — with suitable controls — beyond a specific defined domain.

According to the "application" perspective, it is important to consider the intended application or use of the PKI (i.e., "Which PKI is the right one for this environment?"; "Which is the right one for that purpose?"). For analysis, it is helpful to separate the possible fields of use into the categories that are commonly discussed today: business-to-business (B2B); single business, or enterprise (B); business-to-customer (B2C); and individual (I). It can then be determined whether each category has its own unique "right" PKI, or whether a single PKI can be the "right" choice for multiple categories simultaneously.

A PKI based upon the International Standard X.509 (and the associated standards defined in the IETF PKIX Working Group) is arguably the "right" one from both of the perspectives given above. It most closely fits the definition of an ideal PKI because of the peer-reviewed, formalized, and interoperability-tested mechanisms that have been specified to provide all aspects of key and certificate life-cycle management. (PKIs based upon other certificate formats tend to be lacking in one or more aspects of life-cycle management.) Furthermore, for a variety of reasons, including

- the importance of standards compliance to avoid vendor "lock-in" and to promote interoperability,
- the variety and flexibility of trust models supported, and
- the possibility of central control over EE trust decisions and behaviour,

it can be shown that an X.509-based PKI is ideally suited to B2B or B environments. The prevalence of browser-centric on-line business transactions (both in the wired and in the emerging wireless world) has led to the wide-spread recognition and adoption of the 3rd-party certifier model (a consequence of the X.509 certificate format) for the B2C environment because of the need for trusted "introducers" between otherwise unknown entities. Finally, for the "individual" environment, an X.509-based PKI may not be the natural choice; however, one may validly ask whether this environment requires a public-key *infrastructure* at all (i.e., whether simpler solutions involving public-key technology are more appropriate).

Advancements in the X.509 Standard and in the PKIX Profile over the past few years, including explicit provisions such that

- an X.509-based PKI need not be associated with an X.500-based Directory,
- an X.509-based PKI need not make use of X.500-based Distinguished Names, and
- an X.509-based PKI can be cryptographically bound, in a formal way, to a full-featured Privilege Management ("advanced authorization") Infra-structure,

only strengthen the argument that an X.509-based PKI is the "right" one from virtually any perspective.

3. POSITION STATEMENT BY BURMESTER

Several PKIs based on X.509 certificates have been proposed recently. However as pointed out in the literature, these are inherently weak, both in their design and in the structure of their cerificates (see e.g., [6, 7]). Obviously, by using appropriate security policies some of these weaknesses may be checked, but this approach will not thwart *malicious* attacks. Indeed, there seems to be no practical way to defeat a determined hacker. So the only way to guarantee security is to design a system which will survive a limited number of malicious attacks.

There are essentially three approaches that may be used to support the security of a PKI:

1. a *stochastic* approach,
2. a *security policy management* approach, and
3. a *structural* approach.

Each of these will control certain types of attacks. The stochastic approach uses a probabilistic model with a trust metric. Entities are assigned trust profiles and assurance levels. This model is based on statistical evidence and predifined criteria, and is ergodic. If used properly it will certainly control reliability faults, but it will not control malicious faults.

A security policy management framework that supports the trust relationships between the entities in a PKI may be used to prevent most types of attack, but not all. A determined hacker will always find ways to bypass *whichever* security measures are adopted.

The structural approach uses appropriate trust topologies. Hierarchical PKIs are very efficient, but trade functionality for security. This is because there is only one trust-path which connects any two entities. So public keys may be compromised with only one penetration. In particular all the public keys of the descendents of a penetrated CA (Certifying Authority) in a hierarchical structure may be compromised.

It has been shown that *horizontal* structures will survive a bounded number of penetrations by having a sufficient number of vertex-disjoint trust paths which connect any two entities $2k+1$ for k penetrations [2]. These structures can therefore be used to deal with malicious attacks. However in this case more certificates are needed to authenticate a public key.

Since faults will always occur in any system, however well it is designed, measures must be taken to prevent those faults that can be checked. The others may then be controlled by using appropriate trust topologies. A secure PKI must therefore combine all three aspects. The first two for reliability and to restrict the number of low level penetrations, while the last to thwart the most obnoxious attacks.

4. POSITION STATEMENT BY REITER

No position statement was received in time to be included in the proceedings.

5. POSITION STATEMENT BY ZIMMERMANN

In the minds of many people, the phrase "Public Key Infrastructure" has become synonymous with "Certificate Authority." This is because in the X509 world, the only PKI that is possible is one built on a centralized CA. Matt Blaze made the cogent observation at the RSA conference in January 2000 that: "A commercial CA protects you from anyone whose money it refuses to take." These CAs are "baked into" the major browsers, with no decisions by the users to trust them.

There is indeed a PGP Public Key Infrastructure. But what we call a PKI in the PGP world is actually an emergent property of the sum total of all the keys in the user population, all the signatures on all those keys, the individual opinions of each PGP user as to who they choose as trusted introducers, all the PGP client software which runs the PGP trust model and performs trust calculations for each client user, and the key servers which fluidly disseminate this collective knowledge.

PGP has flourished for many years without the need to establish a centralized CA. This is because PGP uses a decentralized system of trusted introducers, which are the same as a CA. PGP allows anyone to sign anyone else's public key. When Alice signs Bob's key, she is introducing Bob's key to anyone who trusts Alice. If someone trusts Alice to introduce keys, then Alice is a trusted introducer in the mind of that observer.

If I get a key signed by several introducers, and one of these introducers is Alice, and I trust Alice, then the key is certified by a trusted introducer. It may also be signed by other introducers, but they are not trusted by me, so they are not trusted introducers from my point of view. It is enough that Alice signed the key, because I trust Alice.

It would be even better if the several introducers of that key includes two or more people that I trust. If the key is signed by two trusted introducers, then I can be more confident of the key's certification, because it is less likely that an attacker could trick two introducers that I trust into signing a bogus key. People can make mistakes, and sign the wrong key occasionally. PGP has a fault tolerant architecture that allows me to require a key to be signed by two trusted introducers to be regarded as a valid key. This allows a higher level of confidence that the key truly belongs to the person named on the key.

Of course, a clever attacker could trick two or more unsophisticated introducers into signing a bogus public key. But that does not matter in the PGP trust model, because I don't trust unsophisticated introducers that can be so easily fooled. No one should. You should only trust honest and sophisticated introducers that understand what it means to sign a key, and will exercise due diligence in ascertaining the identity of the keyholder before signing the key in question.

If only untrusted introducers sign a bogus key, no one will be fooled in the PGP trust model. You must tell PGP which introducers you trust, and PGP uses that knowledge to calculate if a key is properly certified by an introducer that you trust by looking for signatures from one of the trusted introducers. If the key lacks any signatures from introducers that you've told PGP that you trust, PGP does not regard they key as certified, and won't let you use it (or at least will strongly urge you not to use it). Everyone gets to choose who they trust as introducers. Different PGP users will have different sets of trusted introducers. In many cases, there will be overlap, because some introducers become widely trusted. They may even sign a great many keys, on a full time basis. Such people are called CAs in the X509 world.

There is nothing wrong with having CAs in the PGP world. If many people choose to trust the same CA to act as an introducer, and they all configure their own copies of PGP to trust that CA, then PGP's trust model acts like the X509 trust model. In fact, the PGP trust model is a proper superset of the centralized trust model we most often see in the X509 world. There is no situation in the X509 trust model that cannot be handled exactly the same way in the PGP trust model. But PGP can do so much more, and with a fault tolerant architecture, and more user control of his view of the PGP PKI.

6. REFERENCES

[1] 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.

[2] M. Burmester and Y. Desmedt. Secure communication in an unknown network using certificates. In K. Y. Lam, E. Okamoto, and C. Xing, editors, *Advances in Cryptology—Asiacrypt '99, Proceedings (Lecture Notes in Computer Science 1716),* pages 274–287. Springer-Verlag, November 14–18, 1999, Singapore.

[3] Cren's certificate authority service. http://www.cren.net/ca/index.html.

[4] Cyber security budget initiatives. http://www.whitehouse.gov/WH/New/html/20000215_1.html, February 15, 2000.

[5] D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly secure message transmission. *Journal of the ACM*, 40(1):17–47, January 1993.

[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):1–7, 2000. See also http://www.counterpane.com/pki-risks.html.

[7] D. Goodenough. A Heretic's view of Certificates. www.DGA.co.uk/Cred

[8] 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,* pages 57–66, April 1997, Zurich.

[9] R. L. Rivest and B. Lampson. SDSI—a simple distributed security infrastructure. http://theory.lcs.mit.edu/~cis/sdsi.html.

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