Reflecting on X.509 and LDAP, or How separating identity and attributes could simplify a PKI

Jeroen van de Graaf

Osvaldo Carvalho

Laboratório de Computação Cientifica, UFMG
Avenida Antônio Carlos, 6627 – Belo Horizonte - MG - Brazil

jvdg|osvaldo@lcc.ufmg.br

Abstract. *X.509 certificates can be used to store attributes about its owner, and so can on-line directory systems such as LDAP. In this paper we explore the option of putting little or no data in the certificate itself, and all data in LDAP databases. We show how this approach completely changes the role of the Registration Authority, resulting in a more flexible PKI. In particular it leads to a way to implement Single Sign On, allowing hosting organizations to fully specify and modify access control lists, and for mechanisms in which the user can have some control over which information he shows to which organization.*

Resumo. *Certificados X.509 podem ser usados para armazenar atributos sobre seu dono; LDAP também serve para isto. Neste artigo exploramos a opção de colocar pouco ou nenhum dado no certificado, e todos os dados em bancos de dados LDAP. Mostramos como esta abordagem muda completamente o papel da Autoridade Registradora, proporcionando uma ICP mais flexível. Particularmente proporciona uma forma de implementar Single Sign On, permitindo organizações para especificar e modificar detalhadamente listas de controle de acesso, e mecanismos em que o usuário pode ter algum controle sobre quais informações ele mostra a qual organização.*

1. Introduction

X.509 certificates are rather ambiguous with respect to their validity. On the one hand they embrace the idea that a certificate has a value of its own, without an on-line verification mechanism to verify whether the certificate is really valid, and for this reason various attributes about the owner (or entity) can be included into the certificate. Version 3 of X.509 even allows ACs to extend the set of certificate attributes.

On the other hand, because certificates can be revoked (by their owner or by the CA) before their expiration date, a party who receives a certificate may feel the need to verify its validity by consulting the CA's data base in real time. When X.509 was specified this might have seemed unrealistic, but today, with the Internet, cell phones and PDAs, this has become a realistic alternative; for credit cards transaction on-line verification is the standard procedure. Two standards have been proposed for validating certificates: X.509 contains a specification for Certificate Revocation List (a blacklist containing the the serial numbers of revoked certificates), and RFC2560 specifies the Online Certificate Status Protocol. Apart from the difficulties associated to the correct implementation of these two protocols, a funny contradiction occurs: **given that you intend to consult the CA's database in real time, why bother putting attributes in a X.509 certificate?**

In this paper we defend the thesis that, given the fact that you intend to query a real-time database anyway, it is not worth to include many attributes in the certificate itself. Indeed, we will argue that it is better to put as little attributes in the certificate as possible, and put all the necessary information in an on-line database, for instance through the LDAP protocol. We explore how this impacts a conventional PKI when certificates are used for non-repudiation (and not for authentication).

This approach, when taken to the extreme, implies a dramatic change in the role of a certificate, which loses some of its intrinsic meaning (unless endorsed by some information from an on-line database) and in the role of the Registration Authority. There would be no need to have an RA vet the data related to some certificate owner *before* the certificate is issued. Instead, this vetting can take place *after* the certificate has been issued but before including additional attributes in the (on-line) database; in this process the owner must prove he knows the private key related to the certificate.

The ideas put forward in this paper are based on discussions with people [Bas04, Sie04] who are strongly involved in implementing the Grid Security Infrastructure [But+00], an authentication and privilege management system for fully distributed use of resources in high performance computing. A little-known part of X.509 does specify provisions for dealing with short-lived attributes, called the Privilege Management Infrastructure (PMI), which uses attribute certificates[ITU01]; we discuss this in Section 4. The Simple PKI proposal [RL96] eliminates identity certificates ("private keys are principals"). And the propriety PKI embedded in the Lotus Notes software for cooperation (used by many banks and, reputedly, by the CIA) also makes a very clear separation between authentication and access control.

The structure of this paper is as follows. In Section 2 we describe some of the current problems with X.509/PKI. Then we explain how authentication is implemented in the Grid Security Infrastructure. Section 4 describes attribute certificates, and Section 5 briefly describes LDAP. In Section 6 we generalize the Grid approach to general X.509-PKI, taking it to the extreme by discussing pseudonymous certificates in Section 7.

2. Problems with X.509 and PKIX

The security provided through digital certificates is usually blown to mythical proportions. The reasons for this are probably psychological: in the insecure cyberworld digital certificates seem the answer to almost all security problems. Buyers of security solutions would like to believe that PKI is the panacea to their security nightmares, and security companies have little incentive to help their clients out of their dream. The fact is that PKI is a very complicated technology which is understood by few people, and critically analyzed by even fewer. Let us briefly list some of the problems, for more details see [Gut, Gut02, Sie04].

- **The X.509 standard is extremely complex.** It is a part of X.500, an over-ambitious project for defining a world-wide directory structure. (The word "Light" of the Light Directory Access Protocol means: as opposed to X.500). As a consequence, X.509 is defined using ASN.1, a rather complicated way of defining data structures independently of any programming language. Specifying certificate extensions, for instance, is no trivial matter.

- **There is no unanimity on how *Distinguished Names* should be used in certificates.** DNs are the core of X.500. The idea was that, by using a hierarchical naming convention, every person would get a unique identifier, as follows: CommonName/OrganizationalUnit* / Organization/Country. For instance, a valid DN is Jeroen van de Graaf/LCC/ATI/Reitoria/UFMG/BR (more than one OU is allowed). When Netscape implemented SSL, it ignored the DN formalism, and instead used URLs to implement server authentication. And for end users, an email address seems to be a more logical choice than some contrived DN. To see the confusion, one simply has to look at the root certificates already present in a browser to observe that there is no uniformity.

- **X.509 is ambiguous with respect to its off-line validity.** As mentioned in the introduction, X.509 certificates were intended to have a validity on their own, that is, without any real-time verification. However, since certificates can be revoked for very good reasons, such real-time verification is necessary if transactions are involved in which the validity is a serious issue. For instance, when used for digital signatures one cannot allow that a user can bail out of a commitment afterwards, alleging that its key was compromised.

- **There is no good solution to certificate validation.** There exist two standards to verify the validity of a certificate: X.509 contains a specification for Certificate Revocation List (a blacklist containing the serial numbers of revoked certificates), and RFC2560 specifies the Online Certificate Status Protocol. Neither of them seems to work very well. See [Gut, Gut02] for further discussion and for other online verification protocol proposals.

- **X.509 certificates contain too many attributes.** In the off-line scenario it makes sense for the certificates to contain many attributes, but in an on-line scenario it doesn't. This aspect has various consequences:

  - **All attributes need to be known and defined before the certificate is issued, implying that every time an attribute changes, the old certificate needs be revoked and a new one needs to be issued.** This is a trivial consequence of the fact that once signed by the CA, any modifications will invalidate a certificate.

  - **X.509 blurs authentication and privilege management.** The previous point is particularly damaging when certificates are used for privilege management (access control), a security service which is often not addressed in the official PKI literature. In distributed organizations, the resource providers (host) will allow visitors (users) only if they can define their own access policy (which they can change anytime); they will refuse to cooperate if privilege management is defined in a centralized way. The obvious way out is to have users identify themselves through a digital certificate, and resolve all privilege issues locally. This is exactly how the Grid Security Infrastructure [But+00] works: a user identifies himself through a certificate containing a DN, his DN is mapped into a unix account (this relation is stored in the so-called mapfile), and the local unix administrator defines the rights of this user through in- and exclusion in unix groups. One can view this paper as an attempt to not restrict this philosophy to Grid Security only, but to apply it to the extreme, substituting the conventional PKI with a different infrastructure, providing the same security but (hopefully) more flexible and practical, and providing more privacy.

  - **X.509 does not protect privacy.** It is commonly believed that the CA needs to publish all the certificates it issued by making them available in some repository. This is obviously a big violation of the privacy of individuals in case many personal data are included as certificate attributes. On closer inspection it seems that we can do better. The repository is needed for two purposes: *1)* to look for a certificate, in case one wants to send an encrypted email to some person using the public key of that person—in this case simple identifying information (equivalent to what is found in common directories) is sufficient, no need to include social security number, name of the parents, etc. in the certificate. *2)* One needs a repository to verify the status of a certificate, for instance when verifying a signature. That is, the verifier already has the certificate and he can already see all the attributes and check that its genuine (because it has been signed by a certificate authority), but he needs to know whether it revoked or not. However, in this case the message digest or serial number of the certificate would do the job.

Some of the points mentioned here are addressed by the X.509 Privilege Management Infrastructure, which uses attribute certificates. See Section 4.

- **X.509/PKI has no provisions for end users acting as a small CA.** Technically there is no impediment to an end user creating another public key/private key pair, and signing the first using his “principal” private key, i.e. creating a certificate as if he were a CA. Such certificates can be useful for the purpose of delegation. The GSI uses such certificates, which it calls “proxy certificates” [NTW01]. Created with a short life time of several hours or days, the subordinate private key can be stored on less reliable servers and act as a proxy when it comes to authentication. If it gets compromised, possible abuse can only last for a few hours or days. Another question not addressed by X.509/PKI is: suppose a professor allows a student to make a 100 photocopies. How should this be resolved?

As a final criticism, which applies to any PKI and not just X.509, is the problem of **where and how to store the private key**. When it comes to non-repudiation in the juridical sense, the user is held responsible for anything that has been signed with his private key. So a basic assumption is that the user securely generates and stores his private key. (For an interesting paper explaining why one cannot outsource the generation of a public key, see [CS03].) However, this assumption only makes sense if the technology exists which assure that a private key cannot be exported, or at least that it cannot be used without the owner's consent.

Even though smart cards and smart tokens are widely sold as the final solution to this problem, their security is largely based on a myth, because if these tokens are connected to a computer that has been invaded, the non-repudiation assumption gets severely compromised. We can distinguish three kinds of smart cards (tokens):

- memory cards: these cards come with no protection whatsoever so they offer no security. Unfortunately they are sometimes called smart cards too, undeservedly.
- protection through a PIN: these cards allow memory access after a valid PIN has been entered. The problem is that the PIN is not entered directly in the smart card, but in the computer. This computer could contain some malicious software that 1) records the PIN entered by the user, and 2) grabs the data stored on the smart card. If done cleverly, the card could be cloned.
- protection through a cryptographic processor: the idea of such cards is that they create their own private key, which never leaves the card. Though making cloning impossible, it seems conceivable that a malicious software (malware) gets a signature on a document that the owner never intended to sign. For instance, suppose that the software sandwiches a false signing request to the signing of a legitimate document. The owner thinks his smart card is being used for signing one document, but the malware is actually signing another document (or various other documents) as well, without the owner being aware of this fact. When later confronted with this signed document, the user might have a very hard time proving his innocence. The smart card maintains no log of the documents it has signed (reason why smart smart cards should be developed [Cus03]), and the malware may have removed itself from the computer, erasing all its traces.

Technologically speaking, the bottom line is that when you stick a smart card in an equipment you cannot trust, non-repudiation is very hard to achieve. And biometrics is not going to help you, for the same reasons. A way out would be to develop hardware trusted by the owner, that sits between the computer and the smart card and which has a small keyboard for a PIN or password. But as long as such alternatives do not exist, a government-imposed directive for using digital certificates as a substitute for signatures seems undesirable.

3. Example: the Grid Security Infrastructure

The idea of Grid computing is to pool computational resources. There exist several projects (many of them in physics) in which hundreds of scientists from tens of institutions share resources located at a few centers for high performance computing. Dealing with authentication and privilege management can soon become a project or system administrator's nightmare.

The Grid Security Infrastructure (GSI) provides an elegant solution to this problem [But+00]. Candidate users of the Grid first need to get a certificate from a CA run by their institution. The only information contained in this certificate is the email address and name of the user, put as the DN (thus disobeying X.500). For instance, a certificate would contain "`DN=jvdg@lcc.ufmg.br/Jeroen van de Graaf`". Since this CA is issuing a certificate with identity information about the user, we will call it an **Identity Authority** (IA).

Those coordinating the project, known as the Virtual Organization (VO), will include the user's DN in a public directory, for instance LDAP. Observe that the VO can add any information it likes, such as to which groups the user belongs, thus defining privileges.

When the user wants to use a resource, he proves his identity by showing his certificate and showing he knows the corresponding private key. Once convinced, the Resource Provider (RP) accesses the (LDAP) database of the VO to find the attributes associated to this particular user, and grants access as a function of the information obtained, and its local (security and priority) policy.

From the RP's point of view, the user can be trusted if the RP can trust the policy of the CA run by the organization to which the user belong. That is, as long as the RP can trust that this CA is not issuing certificates capriciously to just anybody, but really verifies the identities of its users (by procedures to be agreed on), the RP can believe in the identity of the user. And if the RP can rely on the information present in the VO's database, it can trust the privileges granted to the user. Since the RP is in control of its own computers, it can impose restrictions that won't allow a guest user to go out of bounds.

Observe that GSI implements a neat separation between identification and privilege management, as does Lotus Notes, as well as PMI presented next

4. Attribute certificates and Privilege Management Infrastructure

The problems with respect to attributes in X.509 certificates mentioned in Section 2 were recognized, and resulted in a little-known extension to the X.509 standard [ITU01]. It is called the Privilege Management Infrastructure and uses attribute certificates. We give a brief summary here, see for instance [Ali00] and [ZM03] for a good introduction.

An **attribute certificate** (AC) is simply a document digitally signed by an Attribute Authority, containing (among others) a reference to an existing X.509 identity certificate, as well as attributes pertaining to the owner of the identity certificate. So an attribute certificate does *not* contain the public key of the owner, but it links some attributes pertaining to him in a trusted manner, through the digital signature of the AA. (Only in this section we use the term "identity certificate" to mean conventional certificates containing a public key signed by a CA, in order to make a distinction with attribute certificates. In the other sections "certificate" refers to an identity certificate, unless the qualifier "attribute" is used.)

Attribute certificates establish a clean separation between authentication (i.e. identification) and authorization (i.e. access control, privilege management). Before issuing a certificate, the Identity Authority convinces himself of the identity and data presented to him by the user (the fact that he is indeed the owner of the private key is implicit in the protocol in which the user requests a certificate to be signed). This is identical to the role of a PKIX Registration Authority; however, now the IA does not need to be concerned about attributes not related to identity. This has become the responsibility of the Attribute Authority, who is concerned with associating attributes to a person in the following way: “If this person has proved his identity to you, then the following attributes apply to him: x;y;z.” So an AA does not concern himself with the identity of a person, but only about inferences that can be made about a person. Observe that in general, one IA can serve many AAs. In the context of a scientist involved in various projects, the same certificate could be used with various AAs. In the context of a citizen, the AAs could correspond with election authorities, revenue service, drivers license, etc.

The separation between authentication and authorization has dramatic advantages:

- It makes interoperability easier, allowing for a distributed privilege management.
- It separates jurisdiction, since the attribute certificates are issued by the authority that “owns” the attributes, thus avoiding delegation of power to the Identity Authority.
- Attribute certificates can have a much shorter life-time than identity certificates, and can be revoked separately.

However, despite these clear advantages, attribute certificates have not been widely accepted (to put it mildly). Few pilot implementation of attribute certificates exist, among them PERMIS. One can speculate about the reasons why attribute certificates are not being used more extensively. Presumably, one major problem is that they inherently inherit most of the shortcomings of X.509 identity certificates outlined earlier. Another problem is that current browsers do not support attribute certificates. We therefore believe that from a practical perspective the alternative strategy of putting the attributes directly in LDAP is worth investigating.

5. LDAP

The Light Directory Access Protocol is a simplified version of the X.500 Directory Access Protocol, which has so many advanced features that many find it too expensive to implement it completely. LDAP is a service which provides basic information about persons or web services, like name, email address, organization, phone number, digital certificate, password hashes, group membership, etc. LDAP is both a database and a protocol, similar to DNS (which, however, contains fewer data). LDAP is optimized for searching and supports replication of the same information to other servers.

The importance of LDAP does not come from the ability to look up a user's phone number, but stems from the fact that it can store all kinds of access control information, by storing attributes about its user. In other words, LDAP can serve as an alternative privilege management infrastructure by doing away with attribute certificates, making the data directly available through LDAP. For instance, the hash of the user's password can be stored on a LDAP server. When a user want to log in to the system, the system, instead of consulting some local password file, connects securely to the LDAP server, who verifies the password and returns an authorization to the system.

LDAP uses both SSL and SASL as security mechanism. The first, the Secure Sockets Layer, is well-known: it can run with any protocol that uses TCP/IP. The combination of LDAP and SSL is called LDAPS. SSL provides confidentiality, integrity, server authentication and client authentication (which is not often used), and is made to be transparent to the application under which it is running, meaning that the application is oblivious of the presence of SSL. This has advantages but also disadvantages. For instance, it is not possible to create digital signatures with SSL, because the application would need to have access to the private key and certificate and require a conscious action of the user, contradicting its transparency. In order to deal with some aspects of LDAP authentication or if one wants to change from anonymous access to authenticated access during an LDAP session, then SSL is no good.

LDAP allows for three types of authentication:

1. anonymous (i.e. none);
2. simple, that is through conventional password, that could be captured by a sniffers;
3. with SASL.

The Simple Authentication and Security Layer is a protocol for general authentication, capable of running on top of TCP/IP or SSL. In principle, SASL can work with protocols other than LDAP, though no other example is known. SASL permits various authentication mechanisms because it is a wrapper protocol, allowing a specific authentication mechanism to be plugged in:

- Anonymous (RFC 2245)
- CRAM-MD5 (RFC 2195)
- Digest-MD5 (RFC 2831)
- External (RFC 2222)
- Kerberos V4 (RFC 2222)
- Kerberos V5 (RFC 2222)
- SecurID (RFC 2808) (token)
- Secure Remote Password (draft-burdis-cat-srp-sasl-06.txt)
- S/Key (RFC 2222) (one-time-password)
- X.509 (draft-ietf-ldapext-X.509-sasl-03.txt)

Of these options, only a few are usually implemented: “Of the mechanisms on the previous list, popular LDAP servers (such as those from Sun, OpenLDAP, and Microsoft) support External, Digest-MD5, and Kerberos V5. RFC 2829 proposes the use of Digest-MD5 as the mandatory default mechanism for LDAP v3 servers.” [Sun03]. So we have that LDAP runs optionally with SASL, which runs on top of SSL (if present) or directly on TCP/IP (if SSL is absent).

It seems overkill to always use SSL and SASL at the same time. For instance, if SSL is used, there is no problem in using simple authentication (i.e. without SASL), because SSL protects the secrecy of the password. Note that in this case the whole session will be encrypted, thus protecting also the data in the query. Alternatively, if the secrecy of the data queried through LDAP is not a big concern, one could opt for getting rid of SSL and use SASL/Digest-MD5 as an authentication mechanism. This will probably be somewhat faster, since the remainder of the session is not encrypted.

6. Generalizing the GSI approach

The basic idea of this paper is to explore how a fully functional PKI would look by expanding the GSI approach explained in Section 3. We propose the use of identity certificates for authentication, while the Attribute Authorities maintain on-line LDAP databases to provide attribute data dynamically. This is very similar to the PMI presented in Section 4, except that we do not use attribute certificates, we propose to store the attributes directly in LDAP. Even though PMI may be a more sophisticated solution from an esthetic point of view, we believe that the LDAP alternative is more pragmatic, and worth exploring. In this section we discuss which modifications would be required to provide functionality similar to the one provided by conventional X.509/PKI.

6.1. Non-repudiation and time-stamping

We can identify three possible uses of a PKI:

1. Non-repudiation, combined with authenticy and integrity, through digital signatures.

2. Authentication and privilege management (which most textbooks overlook).

3. Confidentiality, through the use of public key encryption.

It may seem intuitively clear that the approach presented here will work well for the last two uses but not for the first. A big difference is the role that time plays in the three cases. In authentication and in encryption, everything takes place in a relatively short time span of minutes, maybe days. But in a digital signature, the time span between signing a document and verifying the signature can be very long, up to decades.

So, whereas time-stamping does not play a role of importance for authentication and confidentiality, it is crucial in non-repudiation. In conventional PKIs, in order to retroactively verify the validity of a certificate at a certain moment in time, it is necessary that all the CA's actions have been time-stamped, in particular the act of issuing and of revoking a certificate. Here, we explore a different direction.

We propose that, in cases were non-repudiation is a serious concern, the person who signs obtains a guarantee that the data queried from the AA is valid at that particular time, i.e. he obtains a time-stamped signature of those authorities who provide the attributes. In other words, instead of having the verifier retroactively verify the validity of the attributes (something that could potentially happen several years later), we have the signer collect all the necessary attributes and have it time-stamped before it is delivered to the verifier.

The difference between the two approaches is this. In a conventional PKI, all actions of a CA need to be logged and time-stamped for decades to make sure that the actual situation at any moment in the past can be reconstructed. In the case of certificates with many attributes this can be a painful problem, especially if several institutions are involved. In the approach presented here this wouldn't be necessary. The time-stamped proof that the signer collects from the AA acts as a snap-shop of the situation at the moment he signs.

So note that the role of the time-stamping authority has changed. In the traditional approach, every time a CA makes a modification on some attribute he requires a new time stamp. In the new approach, an AA can change attributes at will, but when a query takes place a time-stamp is required.

6.2. Advantages of this approach

The big advantage of this approach is its larger flexibility:

- Since the certificate contains almost no attributes, getting one becomes easier. In other words, the threshold for users to start using a certificate is lower. See [CCMS03] for a similar idea.

- Also, revocations will be rare. The only need is when the DN changes, or when the owner suspects that the private key is compromised.

- Attributes can be changed without penalty: if a Attribute Authority feels the need to change an attribute it can do so right away
- More privacy, since attributes are only accessible to who needs it. LDAP has various mechanism for access control, which could shield private information from prying eyes. We could even consider a more sophisticated approach like the one taken in Shibboleth, an new authentication and access control protocol similar to GSI but without digital certificates [Shi04]. In Shibboleth an individual gives to each AA a privacy policy, specifying which attributes the AA may reveal to which third party enquiring about him. Such a mechanism would give an individual some control over information about himself, provided that he can trust that the AAs act faithfully, according to the individual's policy. Also, the DN acts as a unique identifier, so if the Attribute Authorities conspire, they will have no difficulties combining the attributes belonging to the same individual and creating his profile.
- One of the big problems of a wide-scale PKI is that agreement is needed about the various Certificate Policies. This is mainly a consequence of the unfortunate link between establishing the identity and establishing the attributes of a person, and the fact that these questions need to be resolved before the certificate can be issued. The PKI presented here simplifies this problem because it lets one entity be responsible for establishing the identity of a person, whereas the other entities can concentrate on the attributes associated to that individual.

6.3. Disadvantages of this approach

On the downside, we can mention the following points:

- The system proposed here requires that Attribute Authorities have their LDAP service on-line 24 hours per day, with negligible down-time. Though not simple nor cheap, it is a well-known problem already tackled by phone companies and credit card companies, for instance.
- It also results in higher network traffic and possible bottlenecks. These need to be studied in detail.
- As an example, in the case of validating a drivers license, it requires a policeman to have immediate on-line access to a database. This might not be realistic, though one can wonder whether any certificate solution is realistic in this case. More precisely, how realistic is it to assume that a policeman does have equipment to verify the CAs signature on a certificate, but does *not* have communication equipment at his disposal?

Other issues are of an economic nature: PKIs have developed a business model in which CAs and Time Stamping Authorities make profits, and any new proposal should take these factors into consideration if it claims to have a real chance of being adopted.

7. Pseudonymous certificates and privacy

We can take the approach presented in Section 4 one step further by using an pseudonymous DN. Instead of putting the name and/or email of the owner, we could put a random number there. Siebenlist[Sie04] proposes to use the base 64 encoding of the SHA1 message digest of the private key.

It is interesting to reflect on what this does. In X.509-PKI parlance, pseudonymous certificates completely separate the role of the CA with its Certification Practices Policy (explaining how it issues certificates) from the RA with its Certificate Policy (explaining to whom it issues certificates). Here, the CA's business is to sign any certificate it sees, without asking questions (though maybe some protection is needed to avoid abuse or flooding the CA).

If considered necessary, identity could be treated as an attribute. That is, an Attribute

Authority could play the role of an Identity Authority, similar to the PKIX RA, but *after* the certificate has been issued, by establishing the identity of the owner whose DN equals some (random) number. This identifying information could be put in an LDAP, or the authority could make it known that it *has* the identifying information. This later alternative could be useful to contain abuse from pseudonymous certificate owners: the Identity Authority's Policy could state that it will reveal this information to other Authorities or Resource Providers if abuse has been observed.

A pseudonymous certificate provides the owner with a virtual identity or pseudonym. That is, in various interactions (sessions, transactions) the owner can show to the entities that he is the same person, however, without showing who he is. Note that in many situations this is sufficient: like with browser cookies, entities are not interested in who you are, they want to know you are the same person they have talked to before, thus maintaining a history between sessions.

Pseudonymous certificates can also provide some level of privacy. For instance, a student might be able get an attribute saying he belongs to some group allowing him access to some digital library. The attribute effectively says: "The owner of this certificate has access to this kind of information" without stating *who* the owner is. This is useful when this student feels embarrassed about the fact that he is looking for some kind of information; in such cases a pseudonymous certificate is more desirable than one containing identifying information. However, true pseudonimity is only guaranteed if neither the CA nor the AA can trace the true identity of the student. In practice this may be complicated.

Canada's Government On-Line Initiative also proposes pseudonymous certificates [Jus03], using a Meaningless But Unique Number as DN. In addition, it advocates that AAs (which correspond to various government programs) use a different number (called Program Identifier) as their data base key, thus allowing individuals to have different identity certificates with different AAs.

In some sense pseudonymous certificates are similar to free email accounts, like @hotmail.com. They give you some pseudonymity (and if you use many different pseudonyms, anonymity) but if your internet provider really wants, they can trace your IP connection and see who you are. However, since the threshold of getting one is very low, a disadvantage might be that the owner attributes less importance to his certificate, and easily shares it with others.

8. Conclusions

Though the basic idea presented here is simple, its implications are profound. It fundamentally changes parts of the X.509/PKI philosophy. The basic difference is that in this new-style PKI only one IA is necessary, issuing a certificate with (almost) no identifying information. Furthermore there would exist various AAs, who provide the credentials associated to this individual in the form of attributes.

We believe that this (infra)structure is simpler than the conventional X.509/PKI and PMI. In particular, there is no need to have all AAs agree beforehand on a Certification Policy to get things up and running; AAs can opt in or out very easily. There also seems to be less need to have the CA issuing various root certificates with different classes, basically creating several parallel structures.

We propose that this mechanism be adopted in the project ICP-EDU, a pilot project of RNP for implementing a Brazilian PKI for academia. The approach presented in this paper should also be studied for ICP-Brasil, the initiative of the Brazilian federal government to implement a nation-wide PKI. There exist a large variety of entities responsible for issuing identity cards: the Federal Police, the Secretaries of Public Security of each state, the OAB, and others. These could act as Identification Authorities. Then there would be various Attribute

Authorities: revenue service (CPF), election authorities (título de eleitor), SUS, INSS, etc. What is currently happening is that many Attribute Authorities are currently acting as Identification Authorities, which seems to lead to a hopelessly inefficient system.

The main purpose of this paper is to provoke a discussion about X.509/PKI, its shortcomings, and its alternatives for the future, especially with respect to the two projects mentioned in the previous paragraph. We hope that this paper contributes to this aim, and leads to viable alternatives.

9. Acknowledgements

In the context of the two Brazilian RNP projects *Directories* and *PKI-EDU,* we were struggling with the question: Where to put the user attributes, in the certificate, or in LDAP? At the same time we were studying the authentication mechanism used in Grid computing, and teaching a course on PKI. The very stimulating discussions JvdG had with Jim Basney during the 1st Workshop on Grid Computing in Petrópolis put many things in place. These ideas are very similar to those presented by Frank Siebenlist [Sie04], who communicated to us that his talk was inspired on ideas from others. We thank them both.

We would like to thank Wilton Caldas for many fruitful discussions about LDAP, privilege management and Grid security. JvdG thanks Ricardo Felipe Custódio for his help and support, and his students for their patience when they saw him struggling with understanding PKI. The referees also provided valuable criticism. Especially the pointer to attribute certificates has improved the quality of this paper.

References

[Ali00] Alifrangis, S. Attribute Certificates. Presentation to the Council on Technology Services, August 2000. Transparencies available from http://www.cots.state.va.us/minutes/ds081000/ACS.ppt.

[Bas04] Basney, J. Private communication, Februari 2004.

[Bra00] Brands, S. *Rethinking Public Key Infrastructures and DigitalCertificates.* MIT Press, ISBN 0262024918, 2000. The for this article most relevant parts of Brands' book are available at http://www.credentica.com/technology/book.html.

[But+00] Butler, R., Engert, D., Foster, I., Kesselman, C., Tuecke, S., Volmer, J., Welch, V. *A National-Scale Authentication Infrastructure.* IEEE Computer pg. 60, 2000. Article available from http://www.globus.org/documentation/incoming/butler.pdf.

[CCMS03] Carnut, M.A., Curvelo Hora, E., Mattos, C.L., da Silva, F.Q.B., *FreeICP.ORG: Free Trusted Certificates by Combining the X.509 Hierarchy and the PGP Web of Trust Through a Collaborative Trust Scoring System.* Presented at the 2nd Annual PKI Research Workshop held at NIST, Gaithersburg (MD), USA, April 2003. Article available from http://middleware.internet2.edu/pki03/PKI03-proceedings.html.

[CS03] Crépeau, C. & Slakmon, A. *Simple backdoors for RSA key generation.* RSA Conference 2003.

[Cus2003] Custódio, R.F. Research project for a smart smart card that remembers what the smart card signs. Personal commumication, 2003.

[Gut] Gutmann, P. *PKI Technology Survey and Blueprint.* Article available from http://www.cs.auckland.ac.nz/~pgut001/pubs/pkitech.pdf.

[Gut02] Gutmann, P. *PKI: It's Not Dead, Just Resting.* In IEEE Computer, August 2002.

[ITU01] ITU-T Recommendation X.509 ISO/IEC 9594-8. The Directory: Public Key and Attribute Certificate Frameworks. May 2001.

[Jus03] Just, M. *An overview of Public Key Certificate Support for Canada's Government On-Line (GOL) Initiative.* Presented at the 2nd Annual PKI Research Workshop held at NIST, Gaithersburg (MD), USA, April 2003. Article available from http://middleware.internet2.edu/pki03/PKI03-proceedings.html.

[NTW01] Novotny, J., Tuecke, S., and Welch, V. *An Online Credential Repository for the Grid: MyProxy.* Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press, August 2001. Article available from http://www.globus.org/research/papers/myproxy.pdf.

[RL96] Rivest, R. L. & Lampson, B. *SDSI – A Simple Distributed Security Infrastructure.* Presented at the Crypto96 Rumpsession. Article available from http://theory.lcs.mit.edu/~rivest/publications.html.

[Shi04] Shibboleth Project. http://shibboleth.internet2.edu.

[Sie04] Siebenlist, F. *Is there life after X.509?* Presentation given at the Security Workshop of the Globus World 2004 Conference, January 2004. Article available from http://grid.ncsa.uiuc.edu/gw04-security/GW04-SecWkshp-life-after-X509.ppt.

[Sun03] SUN. The JNDI Totorial. Available from http://java.sun.com/products/jndi/tutorial/ldap/security/sasl.html.

[ZM03] Zhou, W., Meinel, C. *Implement Role-Based Access Control with Attribute Certificates.* Forschungsbericht 03-5, Institut für Telematik, Universität Trier, 2003. Report available from http://www.informatik.uni-trier.de/~meinel/papers/zhou-alles.pdf.