Global Information Assurance Certification Paper

PKI with a KISS
SANS Security Essentials, GSEC Practical Assignment, Version 1.3

Dale Hapeman
March 14, 2002

- Abstract

The concept of identifying users by issuing them public key certificates has been around for a years. The widespread implementation of applications that use public key certificates, and the Public Key Infrastructures (PKI) required to support them, have been slow to be embraced. We discuss a very common environment where an extremely simple PKI implementation will meet all requirements. In order to speed deployment in these specific environments, we are going to Keep It Simple, Stupid.

- Introduction

There are plenty of papers and products which tout the benefits of deploying a security solution based on public key technology. Whether it be for identification and authentication of users via digital signatures or key management via key negotiation protocols. What we haven't seen a lot of is companies who use this technology on a day to day basis.

The biggest obstacle to the deployment of a PKI is that it's hard. (Yogi Berra would have had a better quote.) The tools are unfamiliar, the cryptography is challenging, the standards are many and changing, and most of all -- the policies and procedures that need to exist to make a PKI trustworthy are extensive and rigid.

Another stumbling block is that before you can use a certificate to identify a person, you have to identify them so you can issue them a certificate. Often, this is being attempted using a scheme that requires identifying the requesting party over the internet.

The contention of this paper is that there are situations where the relationship between the concerned parties is such that the PKI implementations can become much simpler. The situation is one where all participants are somehow "associated" with each other in a formal. or informal "organization". Associated may mean covered by a contract, bound by employment, related by blood, etc. Organization may mean a company, a consortium, a company and it's clients, a clan, etc.

- Simple Solution

Our solution is a PKI that issues Identity Certificates to all of the "members" of a given "organization". It issues certificates for use in situations in which the party that relies on the certificate is also a member of the organization.

This will be a PKI that is implemented entirely within the "organization". This makes it a "standalone PKI", it does not rely on (or assume) the existence of any PKIs outside of the organization. The top level CA (the Root CA) is associated with the controlling organization. Management and ownership of the hierarchy may be implemented entirely within the organization or may be outsourced to any degree. This is often termed a "closed" PKI.

The PKI is hierarchical. This can be a single level or multi level hierarchy but every CA is subordinate to the self-signed Root CA (either directly or through a path that leads "up" to the Root CA).

The important characteristic of this PKI is that that the CAs in the hierarchy issue certificates to members and don't issue certificates to non-members.

- Target Situation

The requirements that need to be met for our solution to be valid are such that a level of trust can be established between the members of this organization and the entity that will be issuing certificates. This could mean that there is one controlling entity which everyone must trust.

All members must trust the PKI to issue certificates to members of the organization and only to members of the organization. The PKI must be trusted to always make sure that the member requesting and receiving the certificate is the member named in that certificate.

The PKI must trust the members of it's organization and/or it's own processes to a level such that it is willing to believe that a certificate request has come from that particular member. This trust is established via some method of identification the requestor.

Trust can also be established by threat of punishment.

These obscure references to "trust" and "members" and "organizations" can best be shown by a few examples.

- The "organization" is a company and the "members" are it's employees. The employees are bound by the policies of the company (whether written, assumed, or made up on the spot) and the ultimate punishment for violating those policies is termination.
- The "organization" is a company and the "members" are customers who have bought a particular service that includes an application that uses certificates. There is a contractual relationship between the company and the customer. The contract can be severed by either party if they violate policies. Even if the contract does not explicitly specify PKI related policies, failure of the company to provide the application or a misrepresentation on behalf of a member would be a contract violation.
- The "organization" is an extended family and the "members" are relatives. Violate the "trust" and you are the black sheep.

The job of the complete PKI is to create a trustworthy environment. The job of the CA (and the person who fills that role) is to:

- Associate an identity with a public key
- Make sure that the public key (and associated private key) really belong to the party represented by that identity
- attest to these facts by issuing a certificate

- Why Not Passwords

The question may be asked: why go through all this trouble when using user IDs and passwords have worked so well (for so long)? We'll ignore the obvious arguments -- cleartext passwords, password guessing, post-it notes on the monitor, etc. The case for public key certificate based authentication can be made on three fronts: management, scalability, and preparation for the future.

Password management issues can become extremely burdensome (to both the user and the administrator), especially when efforts are taken to make them more secure (longer, frequently changing, complex). Possession of a public/private key pair, with a lifetime that is measured in years, is less "administration intensive" on a day-to-day basis. This does imply that the initial roll-out and user learning curve have been overcome.

Scalability is the most often mentioned benefit to a public key enabled identification solution. As the number of users and applications grow, the ability to centrally manage identity information becomes more and more attractive. Add in the concept of a common database of certificates accessible with standard protocols (e.g., LDAP) and PKI solutions scale to millions of users -- beyond the realm of administrators who reset passwords.

As the use of certificates for identification broadens, and more applications become certificate aware, deployment of the appropriate technologies now becomes an investment for the future. Even if a "simple PKI" has to be replaced with a more sophisticated implementation in the future, the lessons learned and processes developed will be reusable.

If we assume the use of standard passwords, any attempts to fix the insecurities that are inherent in their use becomes an effort that rivals the deployment of a PKI in it's complexity. The use of secure protocols to hide cleartext passwords involves solutions such as VPNs or SSL (both of which can or do make use of certificates to some degree). These solutions have their own roll-out and integration stumbling blocks. The imposition of policies that mandate frequently changing, "good" passwords, has both a technology requirement (can all of my applications enforce the same password rules?) and a human component (how am I going to remember THAT?).

There are also other "password based " technologies to consider -- such as one-time passwords.

Again, integration of these schemes with individual applications becomes an issue.

If deployment of a "secure" password based solution is so painful, why not deploy a simple (albeit not painless) PKI based solution?

- Why Not Outsource

There is no shortage of companies that will offer to build and/or manage a PKI for you. For example, see "PKI's Are Still Tough To Deploy" in Internetweek.com. The questions to ask include how much control you want to maintain, how much you are willing to spend up front, and how much each certificate is going to cost you.

If depending on someone else to build, operate, and charge you for their version of a PKI is not very attractive, why not deploy your own simple PKI?

- The Benefits of Simplicity

A "simple" PKI is simple to deploy, simple to apply policies and procedures to, and solves some of the perceived shortcomings of a "PKI solves all" mentality.

Minimal Bells and Whistles

When environments and applications are constrained properly, some of the added features of a PKI become unnecessary. If you manage users elsewhere (such as an HR database or e-mail address book or password file) the need for revoking certificates (at all, much less in a timely manner) becomes moot. This means no CRLs or OCSP servers. If the section on Extensions below convinces you that you don't need extensions -- issuing just got easier. If your PKI product has a built-in database and no one else needs to have access to the contents, you have no need for a global LDAP directory.

Simplified CPS

A Certification Practices Statement (CPS) is a requirement of any PKI. It documents the policies and practices used by a PKI to insure it's "trustworthiness". In our simple environment, not only is the PKI simpler to implement but the document is simpler to write. The rest of this section is the outline (`in courier text`) for a CPS as specified in RFC 2527, Certificate Policy and Certification Practices Framework. It has been annotated with comments that indicate how the document becomes simplified.

```
                        KISS CPS

   1.    INTRODUCTION
```

Provides an overview. Defines the relevant community and applicability of the document, lists contacts. These are all well defined in the environment we are discussing.

```
   2.  GENERAL PROVISIONS
```

Section 2 contains the information that is most simplified by the "simple" PKI we are proposing. We'll look at it in the most detail.

2.1 Obligations

This section defines the obligations of the CAs, RAs, data repositories, subscribers, and relying parties. While these obligations need to be stated, they are not complicated in a closed environment.

2.2 Liability

Liability and financial responsibility are two areas where the assumptions we have made about our "organization" are very valuable. By definition, our organization has existing relationships (e.g., contracts or agreements) which cover these areas. So rather than creating reams of legalese, this section invokes existing documentation.

2.3 Financial responsibility

See Liability above.

2.4 Interpretation and Enforcement

Again, the laws and clauses that would be cited here either already exist or don't apply to a closed environment.

2.5 Fees

Fees and refunds for PKI services either don't apply or are covered in other formal documentation (e.g., service agreements).

2.6 Publication and Repository

This section contains a description of the data that a CA publishes, where certificate and policy information are stored, and who is allowed to access that information. Within our environment, we don't have to deal with public disclosure or limiting access from the general public.

2.7 Compliance audit

Auditing presumably already takes place at some level in our environment. Auditing of the PKI is one more dimension.

2.8 Confidentiality

The control of user information is already addressed by the existing organizational relationship

2.9 Intellectual Property Rights

Intellectual property is always addressed in a company or a company/client relationship. Other organizations are likely to either address these issues elsewhere or not care (e.g., the extended family example).

3. IDENTIFICATION AND AUTHENTICATION

Identification of the members of an organization is another area where a simple PKI takes advantage of existing relationships. During initial registration, rekey, or revocation, the CA needs to be sure they are acting on behalf of the right person. This section of the CPS describes the processes used for identifying users but most likely does not have to invent a new process.

Methods used to assure name uniqueness is also covered in this section. An existing organization is going to have taken care of this issue previously.

4. OPERATIONAL REQUIREMENTS

The operational requirements section discusses the processes and requirements for all phases in a certificate lifecycle. Since identification of certificate users has been simplified (as above), the

processes become more straightforward. The processes still need to be described in detail since they are the foundation for the trust that users have in the PKI. Often, the methods by which a certificate is requested or handled are dictated by the application that uses those certificates.

5. PHYSICAL, PROCEDURAL, AND PERSONNEL SECURITY CONTROLS

Physical, procedural, and personnel security are all addressed by any company's security policy and it can be referred to in this section. If the organization is broader than a single company, this section may quote from a security policy or may require an original discussion. In most cases, sharing this information with members should not be an issue (whereas it might be an issue for a service provider).

6. TECHNICAL SECURITY CONTROLS

Technical security is provided by the products chosen to implement the PKI and by architectural decisions made during implementation. This section must be completed in any environment (we don't get to claim that this section is easier in our "simple" scenario).

7. CERTIFICATE AND CRL PROFILES

Certificates and CRLs most be profiled. Their (mandatory and optional) components are described here. A simple PKI has simple certificates but they still need to be profiled.

8. SPECIFICATION ADMINISTRATION

Describes the lifecycle of the CPS. A company will have policies and procedures covering documentation. A customer and client will be covered by a contract that should discuss the CPS and what happens in the event of changes.

Fewer "Problems with PKI"

Carl Ellison and Bruce Schneier have written a paper exposing what they perceive are the problems with the use of public key technology supported by a PKI -- titled "Ten Risks of PKI: What You're not Being Told about Public Key Infrastructure". The organization we have been discussing, one that is constrained by a number of "simplifying" assumptions, directly addresses some of the authors' concerns. Below, each "problem" is listed (`in courier font`) and the benefits of keeping it simple are discussed.

Ten Risks of PKI

`Risk #1: "Who do we trust, and for what?"`

The authors ask the question, "who made the CA trusted". The answer in this environment is -- "someone in charge". There is a controlling "organization" that says so.

`Risk #2: "Who is using my key?"`

This risk is based on the standard "you can't trust your computer to keep your private key secure" argument. Well, your company can keep your computer as secure as it wants it to be. The authors say "You almost certainly don't own a secure computing system with physical access controls ... is your computer in a locked room, with video surveillance, so that you know

no one but you ever uses it? In a closed community, this may very well be the case -- if need be. If not, someone recognizes that fact and adjusts the confidence level placed on that private key.

Risk #3: "How secure is the verifying computer?"
Same argument, same answer.

Risk #4: "Which John Robinson is he?"
"...how many John Robinsons are in the New York City phone book..." Yes, but if more than one John Robinson is a customer of Acme Widgets, Acme Widgets knows this and assigns the Common Name in a certificate appropriately.

Risk #5: "Is the CA an authority?"
Yes, the CA in our environment knows that EVERY piece of information asserted by a certificate it issues is true and accurate.

Risk #6: "Is the user part of the security design?"
This risk applies more to the e-commerce world than the environment we are describing.

Risk #7: "Was it one CA or a CA plus a Registration Authority?"
"The RA+CA model allows some entity (the CA) that is not an authority on the contents to forge a certificate with that contents." Forging certificates is not allowed, just like forging certificate requests is not allowed, just like stealing office supplies is not allowed.

Risk #8: "How did the CA identify the certificate holder?"
This is a fundamental assumption of our simplifications -- the CA has a relationship with every member such that they can be easily, efficiently, and absolutely identified.

Risk #9: "How secure are the certificate practices?"
Again, as secure as the controlling interests want it to be.

Risk #10: "Why are we using the CA process, anyway?"
In this risk, the authors discusses the problems with Single Sign On. This is not applicable to our discussion.

- Benefits of Having a Contractual Relationship

In describing our "simple" organization, we make special note of relationships mandated by a contract. In this situation, "Trustworthy" is really mandated by the contract and there is a special relationship between the contract (and it's attachments) and the CPS.

The American Bar Association, Information Security Committee has published "Guidelines To Help Assess And Facilitate Interoperable Trustworthy Public Key Infrastructures, PKI Assessment Guidelines". Otherwise known as the PAG. This document is written for parties that are going to asses the "quality" of a PKI. It makes specific note of the importance of any contracts that might apply to an implementation. The PAG states:

> Assessors should review a PKI's documentation to determine what requirements a PKI places on certificate usage and limitations on usage. They should determine whether any externally-imposed requirements apply to the PKI, including those imposed by operation of law or contract. Finally, they should determine whether relying parties participants are, in fact, adhering to any usage requirements.

- Extensions

We claim that simplified certificates might not need to carry extensions. This section looks at some of the more common extensions and describes why they might or might not be required in our environment.

Authority Information Access
This extension holds pointers to data that might be useful for validating the certificate in which it resides. In a closed PKI, the ability to validate a certificate and discover it's certification path is assumed.

Authority Key Identifier
Another aid to certificate path construction. Again, unneeded in a closed PKI.

Basic Constraints
Identifies CAs and can limit the length of certification paths. This is useful in situations of multiple cross-certifications -- an unlikely occurrence in tightly defined communities.

Certificate Policies
Indicates the policies under which a certificate was issued. In our "organization", everyone knows the policy -- and there is only one.

CRL Distribution Points
Valuable if you need to tell people where to find CRLs. In our environment we either don't use them or people have them or know where to get them (e.g., off the intranet server).

Extended Key Usage
Can become useful if many different applications are used within the organization. Then certain certificates would only be used for certain applications. We're not sure why.

Issuer Alternative Names
Help for identifying the issuer of a certificate. We know who issued our certificates.

Key Usage
If you use different key pairs for different security services using different algorithms, this extension helps you (and everyone else) keep track of the multiple certificates needed. In the name of KISS, we'd like to limit the occurrence of this situation.

Name Constraints
Used for limiting cross certification. Again, a requirement for cross certification is unlikely in the environments we are describing.

Subject Alternative Names
This extension can carry additional names used by the subject of a certificate. Useful if you are not storing this information in a directory or database someplace.

Subject Directory Attributes
This extension can carry any additional information associated with the subject of a certificate. Also useful if you are not storing this information in a directory or database someplace.

Subject Key Identifier
An identifier for the public key in a certificate. Useful when you have the same public key in multiple certificates. This is complicated -- not simple.

Policy Constraints
Constrains the certification paths so that some are not valid. A closed PKI should not need to impose these types of constraints -- it has control over all paths.

Policy Mappings
Used where a certificate is used across different domains that use different policies. Not true in our simple case.

RFC 2459: Internet X.509 Public Key Infrastructure, Certificate and CRL Profile describes recommended extensions in Section 4.2, Standard Certificate Extensions. Adherence to this specification results in certificate which are "PKIX-compliant". This profile dictates that many of these extensions be mandatory. While adopting this format is probably the most interoperable and extensible option, it may not be the simplest (which is what we are discussing here). Although, PKI products are being delivered with built-in PKIX-compliant profiles which makes issuing compliant certificates that much easier.

Of course, individual commercial applications which are certificate aware might have their own requirements for certificate extensions. Hopefully, in the future, a PKIX-compliant certificate will be a standard standard.

- Profile

RFC 2459 describes the PKIX certificate profile. RFC 2527 asks you to discusses the profile for your certificates in a CPS using the following outline:

```
7.1  Certificate Profile
  7.1.1  Version number(s)
```

```
  7.1.2  Certificate extensions
  7.1.3  Algorithm object identifiers
  7.1.4  Name forms
  7.1.5  Name constraints
  7.1.6  Certificate policy Object Identifier
  7.1.7  Usage of Policy Constraints extension
  7.1.8  Policy qualifiers syntax and semantics
  7.1.9  Processing semantics for the critical certificate policy
extension
```

Here is an example of a VERY simple certificate for the PKI deployed at a company that issues certificates to it's customers.

Field	**Value or *Required Field***
Version	3
Serial Number	*Serial Number*
Issuer DN	*DN of CA*
Subject DN	UID = *UserID*
	E = *E-mail address of Subscriber*
	CN = *Name of Subscriber*
	O = *Name of Company*
Validity Interval	1 year
Subject Public Key	rsaEncryption (1024 bit)
	Algorithm Identifier: {1 2 840 113549 1 1 1}
Signature	sha-1WithRSAEncryption
	Algorithm Identifier: {1 2 840 113549 1 1 5}

Serial Number is assigned by the PKI product
UserID is coordinated with Customer Service
E-mail address of Subscriber is coordinated with Customer Service
Name of Subscriber is coordinated with Customer Service
Company Name is coordinated with Customer Service

- Background

This paper talks about simplicity of the PKI. An article by James Kobielus in Network World, "Simplification, not XML, is the key to PKI success" reiterates the need for simplicity:

> "Most important, we need radical simplicity of PKI and PMI configuration at the client level. This stuff has to be cheap and easy to set up and manage on the desktop, laptop and palmtop. Otherwise, it won't succeed in the mass market. We've seen too many 1990s visions stumble on the doorstep to the new millennium."

- References

American Bar Association, Section of Science & Technology Law, Electronic Commerce Division, Information Security Committee. "Guidelines To Help Assess And Facilitate Interoperable Trustworthy Public Key Infrastructures, PKI Assessment Guidelines", version 0.30, Public Draft for Comment, June 18, 2001.
http://www.abanet.org/scitech/ec/isc/pag/pag.html

Chokhani, S. and Ford, W. "Internet X.509 Public Key Infrastructure, Certificate Policy and Certification Practices Framework, Request for Comments: 2527", March 1999.
http://www.ietf.org/rfc/rfc2527.txt

Depompa Reimers, Barbara. "PKI's Are Still Tough To Deploy", Internetweek.com, April 9, 2001. **http://www.networkcomputing.com/1218/1218f3.html**

Ellison, C. and Schneier, B. "Ten Risks of PKI: What You're Not Being Told About Public Key Infrastructure" Computer Security Journal, v 16, n 1, 2000. **http://www.counterpane.com/pki-risks.html**

Housley, R., Ford, W., Polk, W., Solo, D. "Internet X.509 Public Key Infrastructure, Certificate and CRL Profile, Request for Comments: 2459", January 1999.
http://www.ietf.org/rfc/rfc2459.txt

Hously, Russ and Polk, Tim. Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructure. New York: John Wiley & Sons, Inc., 2001.

Kobielus, James. "Simplification, not XML, is the key to PKI success", Network World, May 7, 2001. **http://www.nwfusion.com/columnists/2001/0507kobielus.html**

RSA Security. RSA Keon Certificate Authority 6.0, Administrators Guide. Bedford, Massachusetts: RSA Security, Inc., 2001.

Xcert International. A Practical Guide to Public Key Infrastructure. Walnut Creek, California: Xcert International Inc., 1999.