Modelling Trust Structures for Public Key Infrastructures*

Marie Henderson[1], Robert Coulter[2]**, Ed Dawson[1], and Eiji Okamoto[3]

[1] Information Security Research Centre, Queensland University of Technology
marie@serc.rmit.edu.au, e.dawson@qut.edu.au

[2] School of Computing and Mathematics, Deakin University
shrub@deakin.edu.au

[3] Institute of Information Sciences and Electronics, University of Tsukuba
okamoto@is.tsukuba.ac.jp

Abstract. The development of Public Key Infrastructures (PKIs) is highly desirable to support secure digital transactions and communications throughout existing networks. It is important to adopt a particular trust structure or PKI model at an early stage as this forms a basis for the PKI's development. Many PKI models have been proposed but use only natural language descriptions. We apply a simple formal approach to describe the essential factors of a PKI model. Rule sets for some PKI models are given and can be used to distinguish and classify the different PKI models. Advantages for this approach with conglomerate PKIs, those that are built from multiple distinct PKI models, are discussed.

1 Introduction

Public key cryptography has now matured to the point of being widely used in real world applications. The security services supported by public key cryptography include authentication, confidentiality, integrity and non-repudiation. Combining public key cryptography with other cryptographic mechanisms (such as symmetric cryptography) provides the most practical and efficient cryptographic solution for securing modern communications. This is the reason organisations are now applying public key cryptography.

With public key cryptography each entity has a public key, which is available to all entities, and a private key, which is protected from disclosure and whose use is limited to the owner. With symmetric cryptography the difficulty is to transport the keys while preserving confidentiality. With public key cryptography confidentiality is not required for the public key but rather a guarantee of authenticity (to protect against masquerade attacks etc.). Trusted authorities called Certification Authorities (CAs) provide such guarantees by issuing

* Research sponsored by the Telecommunications Advanced Organisation (TAO) of Japan.

** This author was funded by a QUT postdoctoral research fellowship.

L. Batten and J. Seberry (Eds.): ACISP 2002, LNCS 2384, pp. 56–70, 2002.

certificates which link the public key to other data in the certificate. The certificate data may be identity information (as with the X.509 standard [9]) or authorisations (as with SPKI [4]), and other information. An entity may prove their association with the public key of a certificate by using their private key to digitally sign a random challenge or other communication.

The term Public Key Infrastructure (PKI) is used to cover the management and distribution of public keys and associated data. A simple public key exchange could consist of two entities, Alice and Bob, who meet and exchange public key values. In a global setting, such as the Internet, it is impossible for all parties to exchange public keys in this way, motivating the use of intermediary CAs. In this case it will be necessary to use multiple CAs to service the large community of users. The structuring of the relationships between these CAs becomes an important issue for constructing a PKI. Basically, the CA structuring reflects how the CAs issue certificates. A number of generic structuring models have been proposed, some of which we will discuss. We will refer to these structuring models as *PKI models*. The term trust models is also used as the PKI model describes how trust is referenced within the PKI. In a PKI a *trust anchor* is any CA (or rather their certificate or public key) which is trusted without the trust being referenced through the PKI certificates. An example is in a hierarchy PKI model where the top most CA, sometimes called the *root CA*, is the trust anchor for the PKI. The public keys of trust anchors must be obtained out-of-band. Further references and background information on PKI and the related cryptographic security services can be found in [8].

In this article we give formal descriptions for PKI models, focusing on those that look to achieve broad coverage (i.e. they service a large user community or cross national or organisational boundaries). In [13] and [6] informal descriptions for some PKI models are given. However, the main focus of [6] is to outline weaknesses of existing PKI solutions and provide some currently feasible remedies. In this article we focus on the PKI models and so exclude such areas from consideration (the interested reader can refer to [6]). In Section 2 we provide a brief overview of some existing PKI models. We also include a PKI model from [13] as this incorporates an approach distinct from PKI models considered elsewhere. In Section 3 we give a formal description of PKI models from Section 2. We explain how this improves upon the natural language description and also discuss the new PKI models of [13]. In Section 4 we consider issues facing the development of a global PKI, the most challenging setting for establishing a PKI. We give a useful mechanism for joining multiple PKIs which is motivated by our analysis of PKI models in Section 3. Concluding remarks are made in Section 5.

2 Existing PKI Models

The consideration of PKI models has often been confused with certificate standards. For example, many regard the X.509 standard as being synonymous with a hierarchical PKI model. However this is not the case. In fact this scenario was deliberately avoided by the authors of X.509 as it would severely restrict the

flexibility of the standard to adapt to different requirements. The choice of PKI model is largely independent of the certificate structure, although it may place certain requirements for additional information to be carried by the certificate. Past and present attempts to develop workable (broad coverage) PKI models include, but are not limited too, the following PKI models. Our choice is motivated by the fact that this collection of PKI models represents a useful sample set for our development in Section 3 and discussion in Section 4.

Pretty Good Privacy (PGP)
PGP [14] is an unregulated PKI where each entity controls which public keys they trust. It is used by individuals to encrypt and digitally sign electronic mail messages and files. We shall refer to PGP-like PKI models as *mesh* models (mesh models are also known as webs of trust or the user centric trust model). Figure 1 depicts a mesh model with five entities.

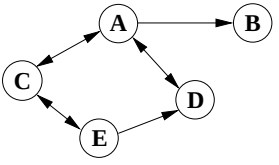

Fig. 1. A five entity incomplete mesh model

PGP is anarchic by design. This means that even though PGP may be used widely, it does not scale well or lend itself to large deployments.

Privacy Enhanced Mail (PEM)
PEM was developed by the Internet Engineering Task Force (IETF) to secure Internet email. The PKI model adopted consisted of a (global) *hierarchy* model with a single trust anchor (the Internet Policy Registration Authority), a lower layer of policy CAs, and lower level CAs. Figure 2 depicts a simple hierarchy with a total of four levels.

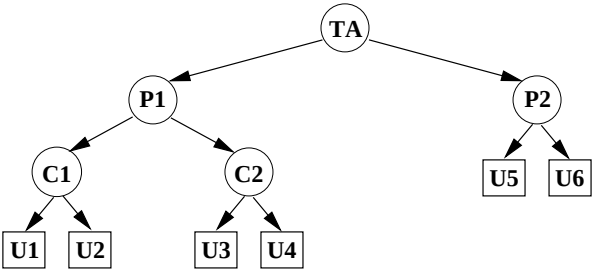

Fig. 2. A hierarchy model

Of the four associated documents, Part II specifies the supporting PKI model [10]. The certificate structure adopted was the X.509 version 1 certificate format. Each policy CA published and registered with the trust anchor a policy statement covering all lower certificates. All PEM CAs adhered to name subordination rules based on the X.500 naming scheme. For a variety of reasons, PEM was never widely adopted and has been replaced by the IETF with PKIX [7] which does not mandate a PKI model.

ICE-TEL Project

The now finished ICE-TEL project employed a PKI model where hierarchies were joined using X.509 cross certificates between the trust anchors[1] merged with PGP (as individual users control their own set of trusted public keys [2]). We shall refer to such PKI models as a *web of hierarchies* model. Each user keeps a set of trusted public keys of users and certificates of trust anchors. The hierarchies operate using X.509 version 3 certificates and each separate hierarchy is referred to as a *security domain*. The trust anchors can construct cross certificates between the hierarchies and X.509 extensions are used in the usual way to control the trust relationship extension into other security domains. Users may also place limits on the number of cross certificates they will accept in a certificate path. Security domains of higher trust may reside within the structure of a lower security domain but not conversely. The ICE-TEL PKI covered a number of European countries but was primarily focused on servicing the academic and research community. Note that the utilisation of the university based academic and research community follows the early evolution of the Internet. Figure 3 depicts a simple mesh of three hierarchies (with trust anchors T1, T2 and T3) one of which is a sub-hierarchy of another (T2 is below C2) and a single external user, X, trusted by U6.

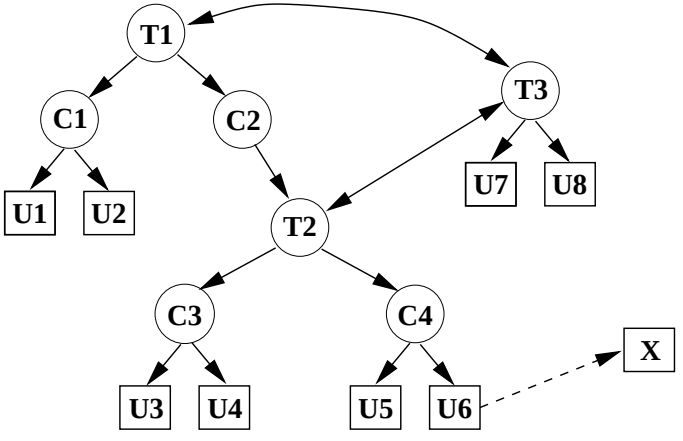

Fig. 3. An example ICE-TEL model

[1] the ICE-TEL project used the term trust points instead of trust anchors

Bridge CA
A Bridge CA (BCA) acts as a hub CA for a number of different PKIs. Each PKI joins to the BCA through their own Principal CA (PCA). The PCA may coincide with PKI trust anchors (for example, in a hierarchy the PCA is normally the trust anchor). Figure 4 depicts five PCAs joined through a BCA. The PCAs would be in turn joined to their respective PKIs.

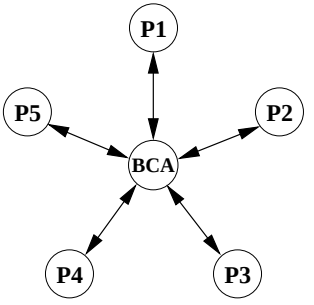

Fig. 4. A Bridge CA model joining five PKIs

Using BCAs (in place of bilateral arrangements between separate PKIs) can decrease the total number of cross certificates required to join the PKIs. The BCA does not become a trust anchor for any of the PKIs as it is not directly trusted by any of the PKI entities. Rather trust is referenced from internal PKI trust anchors. The United States federal PKI (FPKI) project is attempting to join together multiple PKIs set up under separate federal agency programs using bridge CAs. See [8, Chapter 16] for further information or the steering committee home page [5].

Up-Cross-Down PKI
This PKI model is discussed in [13]. It is distinct from the PKI models already presented as it has not even been considered for implementation. The PKI is based on X.509 version 3 certificates and makes use of the same name subordination rules as PEM. In other words, a hierarchical name space is used with name subordination rules supported by a suitable directory structure such as X.500 (this simplifies retrieval of stored certificates). Effectively, this imposes an association between each PKI entity and their level in the PKI. The level is determined from the length of the entities (X.500) name. There are three types of certificates:

Up: a child certifies a parent node.
Cross: any node certifies another node on a separate branch.
Down: a parent certifies a child node.

In the basic system any user will progress up through the name space until a least common ancestor or cross certificate to an ancestor of the target name is found. The certificate path contains either:

- the certificates up to the least common ancestor and the certificates down to the required certificate, or
- the certificates up to the cross certificate, the cross certificate, and then those down to the required certificate.

The name subordination rules disallow lower level entities from creating certificates for higher level entities unless for the unique entity that has created a certificate for them. Cross certificates at the same level are allowed. Figure 5 depicts a simple up-cross-down model with three levels.

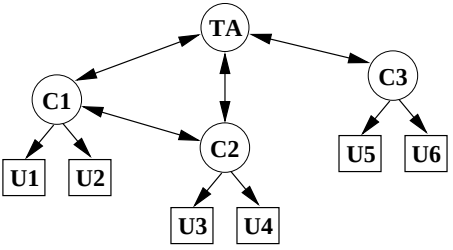

Fig. 5. An up-cross-down model with three levels

3 A Formalisation of PKI Models

Natural language is very flexible and expressive but lacks precision. Unambiguous descriptions are supported through the use of formal languages. The use of formal methods to support the clear specification of PKI processes has been used in [11,12,3]. In this section we shall take a formal approach to precisely describe some of the basic PKI models that have been proposed (specifically, we consider the PKI models presented in the previous section). This is not intended to replace the existing natural language descriptions but to help clarify the distinguishing features. This approach has limitations. For example, the scope of coverage for certificate contents must be restricted to enable a concise description. However, such limitations still affect natural language descriptions (descriptions of PKI models normally only include the certificate contents required for the description). Overall, our descriptions are comparable and improve clarity and analysis methods. We shall take the following approach:

> *A PKI model consists of a set of certificates which adhere to a given set of rules. It is not just the certificates that distinguish the PKI model but also the rules that govern the construction of certificate paths or, equivalently, the issuance of certificates.*

Indeed, it is possible to derive from a given set of certificates different PKI models as the certificates do not supply all information about how the PKI is structured: this must come also from rules describing acceptable certificate paths.

The most detailed definition of a certificate (and the best known) is given in the ITU-T X.509 standard [9]. X.509 gives an extensive description of a certificate's contents and processing rules. In [12] formal methods are applied to analyse these processing rules for the IETF PKIX standard [7]. In this article we do not require as detailed an examination of the contents of a certificate, rather we use a high level description of some generic certificate data fields, similar to that given in [11]:

- `issuer`: the issuer identity.
- `public key`: the public key value for this identifier.
- `identifier`: the subject identity or authorisation for the public key.
- `use-period`: the period over which the certificate is to be used.
- `data`: additional data fields.
- `signature`: the signature created by the issuer over all other certificate data.

In [11] the focus is to model the functions of a working PKI such as issuance of certificates, validation of certificate paths, etc., so a number of certificate fields were required. We can restrict the number of certificate fields even further as we are not concerned with these functions of the PKI but instead with static PKI models or structuring rules. Nonetheless, it is useful to have this basic description of the certificate data fields. The relationship between our approach and that of [11] is discussed again at the end of this section.

We begin our development with some high level definitions for fundamental sets (or types):

- `ENT`: represents the set of PKI entities (these are PKI members that are either people, machines or processes, etc.),
- `ID`: the set of all PKI identifiers (e.g. for authorisations or to simply denote a public key),
- `DATA`: the set of all additional certificate data, and
- `CERT`: the set of all PKI certificates.

We assume that a certificate is the tuple (x, y, z) where $x \in \texttt{ENT}$, $y \in \texttt{ENT} \cup \texttt{ID}$, and $z \in \texttt{DATA}$. The set $\texttt{ENT} \cap \texttt{ID}$ is assumed to be empty. This allows certificates to exist in the PKI that connect public keys to identities *or authorisations*. We use the set `DATA` to cover all additional certificate data not relevant to the immediate discussion (such as public key value, period of certificate use, policies, signature etc.). Then, as in [12], the certificate contents can be later extended, as desired. This gives the scope to cover further PKI operations or processing if desired. Simply, we have

$$\texttt{CERT} \subseteq \texttt{ENT} \times (\texttt{ENT} \cup \texttt{ID}) \times \texttt{DATA}. \tag{1}$$

As in [11], we use the following functions to obtain the component values from a certificate $C \in \texttt{CERT}$ given by $C = (x, y, z)$:

$$I(C) = x,\ S(C) = y, \text{ and } D(C) = z.$$

Loosely, these functions may be interpreted to return the *issuer* of the certificate, $I(C)$, the *subject* of the certificate, $S(C)$, and the remaining *data* of the certificate, $D(C)$, respectively. From a set of PKI certificates the set of all certificate paths can be constructed. Let seq CERT be the set of all sequences of certificates from CERT. For example, if $\mathtt{CERT} = \{(a, b, c), (x, y, z)\}$ then

$$\operatorname{seq}\mathtt{CERT} = \{\langle\rangle, \langle(a, b, c)\rangle, \langle(x, y, z)\rangle, \langle(a, b, c), (x, y, z)\rangle, \langle(x, y, z), (a, b, c)\rangle\}$$

where $\langle\rangle$ is the empty sequence and $\langle C_1, \ldots, C_n\rangle = \{(1, C_1), \ldots, (n, C_n)\}$. We shall also use the sequence join function:

$$\langle X_1, \ldots, X_m\rangle^\frown\langle Y_1, \ldots, Y_n\rangle = \langle X_1, \ldots, X_m, Y_1, \ldots, Y_n\rangle.$$

Let $\#(A)$ denote the number of elements in the set A. Next we construct the set of all certificate paths, CERT_PATH, of a PKI from the set seq CERT:

$$\begin{aligned}\mathtt{CERT\_PATH} =& \{(1, cert) \mid cert \in \mathtt{CERT}\} \cup \\ & \{s \in \operatorname{seq}\mathtt{CERT} \mid \#(s) \geq 2 \wedge S(s(i)) = I(s(i+1)), 1 \leq i \leq \#(s) - 1\}\end{aligned}$$

Thus the set CERT_PATH contains all single certificate sequences and all longer sequences of certificates from CERT for which the issuer of each certificate in the sequence is the same as the subject of the previous certificate. There are, of course, equivalent definitions. We call $p \in \mathtt{CERT\_PATH}$ a *loop* of certificates if $S(p(\#(p)) = I(p(1))$, i.e. if the subject of the last certificate is also the issuer of the first certificate. Note that for a loop of certificates $\langle C_1, C_2, \ldots, C_k\rangle \in \mathtt{CERT\_PATH}$, there will be k certificates paths in CERT_PATH given by $\langle C_i, C_{i+1}, \ldots, C_{i+k}\rangle$ for $i = 1, \ldots, k$ that are also loops.

Let $\mathbb{P}(X)$ represent the power set of a set X (i.e. the set containing all subsets of X). We shall use the functions $ancestor : \mathtt{ENT} \to \mathbb{P}(\mathtt{ENT})$ given by:

$$ancestor(x) = \{y \in \mathtt{ENT} \mid \exists\, cert \in \mathtt{CERT} \bullet S(cert) = x \wedge I(cert) = y\}$$

and $des\_set : \mathtt{ENT} \to \mathbb{P}(\mathtt{ENT} \cup \mathtt{ID})$ given by:

$$\begin{aligned}des\_set(x) = \{z \in \mathtt{ENT} \cup \mathtt{ID} \mid \exists\, path \in \mathtt{CERT\_PATH}, (i, k \in \{1, \ldots, \#(path)) \bullet \\ I(path(i)) = x \wedge S(path(k)) = z \wedge k \geq i\}\end{aligned}$$

The first function, *ancestor*, returns the set of entities who have issued a certificate to the given entity, $x \in \mathtt{ENT}$. The second function, *des_set*, returns the set of all entities or identifiers that the given entity's, $x \in \mathtt{ENT}$, certificate's have been used to certify in any certificate path. Note that the second function will only count once any descendants originating from certificate loops.

Further set definitions that we shall use in our descriptions are as follows:

- `CA` a subset of `ENT` of special authorities, known as Certification Authorities (CAs), who manage certificates.
- `TA` a subset of `CA`, known as Trust Anchors.
- `BCA` a element of `CA` known as a Bridge CA.
- `PCA` a special subset of `CA` known as Principal CAs.

We are now ready to begin our description of the PKI models given in Section 2. We note that the rules listed do not form a unique set of rules for each model, i.e. many equivalent rule sets can be given.

A: The Mesh Model (PGP)
The mesh model follows PGP as it is unregulated (see Section 2) but allows the incorporation of CAs (PGP has also moved towards utilising CAs in later versions). The mesh provides a limiting case for our analysis, in one sense, as there are no rules applied to the construction of certificate paths, i.e. all certificate paths are acceptable.

B: The Hierarchical Model (PEM)
We consider a hierarchical model with a single trust anchor, as proposed with the PEM project.

B1: $\forall\, cert \in \texttt{CERT} \bullet I(cert) \in \texttt{CA}$

B2: $\#(\texttt{TA}) = 1$

B3: $\forall\, p \in \texttt{CERT\_PATH} \bullet I(p(1)) \in \texttt{TA}$

B4: $\forall cert \in \texttt{CERT} \bullet \#(ancestor(S(cert))) = 1$

Rule B1 states that all certificates are issued by CAs. Rule B2 states that there is a single trust anchor. Rule B3 states that all certificate paths begin with a certificate issued by the trust anchor. Rule B4 ensures that for each certificate subject there is a unique issuer (we have assumed that trust anchors have issued self signed certificates - a common PKI practice). Again, the hierarchy is another limiting case in the sense that each entity has only a single certification path. In this regard, the hierarchy is the most regulated PKI model.

C: The Web of Hierarchies Model (ICE-TEL)
The hierarchical model can be extended to cover a collection of cross certifying trust anchors and individual users building limited PGP-like relationships as implemented with the ICE-TEL project. Basically the trust anchors are not limited in regards to the certificate paths constructed between them, i.e. the certificate paths between trust anchors are constructed along the lines of the mesh model. The rules of the hierarchy model shall be enforced for descendants of a trust anchor whom are not themselves trust anchors[2] unless they are single

[2] In the terminology of [2] trust anchors define separate *security domains* so that each security domain employs a hierarchical PKI model. However, as each security domain is controlled by a single trust anchor, there is a one-to-one correspondence between these sets. In this case there is no need for us to make a distinction.

end user public keys trusted by individual users. To keep things simple we shall use the certificate structure to model this exchange of public keys. The issuer of such certificates shall be the entity who has chosen to trust them and they shall use as the subject of the certificate the special identifier $\texttt{pub\_key} \in \texttt{ID}$. User limits on the number of acceptable cross certificates in a certificate path does not affect the set of constructible certificate paths (except from their perspective). Therefore this does not affect the PKI model so we do not consider it further. These requirements are reflected in the following rules:

C1: $\forall\, cert \in \texttt{CERT} \bullet (I(cert) \in \texttt{CA}) \lor (I(cert) \notin \texttt{CA} \land S(cert) = \texttt{pub\_key})$

C2: $\texttt{TA} \neq \emptyset$

C3: $\forall\, p \in \texttt{CERT\_PATH} \bullet I(p(1)) \in \texttt{TA} \lor (I(p(1)) \notin \texttt{CA} \land \#(p) = 1)$

C4: $\forall\, cert \in \texttt{CERT} \bullet S(cert) \notin \texttt{TA} \Rightarrow \#(ancestor(cert)) = 1$

The first rule, C1, states that only CAs issue certificates with the exception of certificates identifying public keys and in this case such certificates are not issued by CAs. Rule C2 ensures that the set of trust anchors, TA, is non-empty (distinguishing this model from the mesh model). Rule C3 states that all certificate paths begin with a certificate issued by a trust anchor or are a single certificate path. Rule C4 guarantees that for each subject of a certificate, who is not a trust anchor, there is a unique issuer. These rules can be modified to exclude public keys. However, we have included this aspect to completely describe the ICE-TEL model as given in [2].

D: The Bridge CA Model (FPKI)

The Bridge CA model is not a stand alone PKI model but, as described in Section 2, is used to join a number of other PKIs using a single BCA and PCAs for each of the joined PKIs. Therefore, no restrictions are placed on the joined PKIs in regards to the models they employ, except that they have a single PCA. We need to define a set of PKIs $\{\texttt{PKI}_1, \ldots, \texttt{PKI}_k\}$ which are joined via the BCA. For $\texttt{PKI}_i$, $1 \leq i \leq k$, the full set of certificate paths are given by $\texttt{CERT\_PATH}_i$, the set of acceptable certificate paths is given by $\texttt{ACC\_PATH}_i$ (i.e. those that satisfy the rules for the $\texttt{PKI}_i$ model), the set of entities is given by $\texttt{ENT}_i$, the set of trust anchors is given by $\texttt{TA}_i$, and the PCAs are given by $\texttt{PCA}_i$. Also, $data_{\texttt{iBCA}}$ represents the additional data in the certificate issued by $\texttt{PCA}_i$ to BCA whereas $data_{\texttt{BCAi}}$ represents the additional data in the certificate issued by BCA to $\texttt{PCA}_i$. The certificate paths of the entire PKI are described by the following rules:

D1: $\texttt{BCA} \notin \bigcup_{i=1}^{k} \texttt{ENT}_i$

D2: $\texttt{ENT} = \{\texttt{BCA}\} \bigcup \left(\cup_{i=1}^{k} \texttt{ENT}_i\right)$

D3: $\forall i \in \{1, \ldots, k\} \bullet \texttt{PCA}_i \in \texttt{CA}_i$

D4:
$$\texttt{ACC\_PATH} = \left(\bigcup_{i=0}^{k} \texttt{ACC\_PATH}_i\right)$$
$$\bigcup_{\left(\substack{i,j=0 \\ i \neq j}\right)}^{k} \Big(\{p_i \in \texttt{ACC\_PATH}_\texttt{i} \mid S(p_i(\#(p_i))) = \texttt{PCA}_i\}$$
$$\frown \{(\#(p_i)+1, \texttt{PCA}_i, \texttt{BCA}, data_{\texttt{iBCA}}), (\#(p_i)+2, \texttt{BCA}, \texttt{PCA}_j, data_{\texttt{BCA}j})\}$$
$$\frown \{p_j \in \texttt{CERT\_PATH}_\texttt{j} \mid I(p_j(1)) = \texttt{PCA}_\texttt{j}\} \Big)$$

The first rule states that the BCA is not an entity in any of the joined PKIs (so trivially can not be a trust anchor for any PKI). Rule D2 determines the entity set. Rule D3 ensures that the PCA for each PKI is a CA of the PKI. The rule D4 gives all certificate paths: those from the joined PKIs and the new certificate paths which pass through the BCA from a PCA of one PKI to the PCA of another PKI.

E: The Up-Cross-Down Model

The only restriction placed on certificate paths by this PKI model is that of the name subordination rules from the X.500 naming scheme, see [8]. We may represent this by assigning a level to each CA in `CA` and requiring that a lower level CA can only issue a single certificate to a higher level CA. Let $level : CA \to \{1, \ldots, n\}$ be the function mapping each CA of the PKI to their level within the PKI (that is to say their depth in the name subordination hierarchy). Here n is the maximum length of any CA name. It is not clearly stated in [13] that only certificates issued by CAs are acceptable. However, we shall assume that this is the case here.

E1: $\forall\, cert \in \texttt{CERT} \bullet (level(I(cert)) - level(S(cert))) \in \{0, \pm 1\}$

E2: $\forall x \in \texttt{ENT} \bullet \#\{cert \in \texttt{CERT} | S(cert) = x \wedge level(I(cert)) > level(x)\} = 1$

E3: $\forall x \in \texttt{ENT} \bullet (\exists\, c_1 \in \texttt{CERT} \bullet I(c_1) = x \wedge level(S(c_1)) > level(x)) \Rightarrow$
$(\exists\, c_2 \in \texttt{CERT} \bullet I(c_2) = S(c_1) \wedge S(c_2) = I(c_1))$

E4: $\forall\, cert \in \texttt{CERT} \bullet (I(cert) \wedge S(cert)) \notin \texttt{ENT}$

The first rule, E1, states that for any certificate the maximum distance between the level of the issuer and the level of the subject is one. Rule E2 ensures that for any entity there is exactly one certificate issued from a higher level with that entity as the subject. The rule E3 states that for any entity there is at most one certificate with a subject from a higher level which has been issued by this entity, and that subject is the unique issuer from E2. Finally E4 ensures that entities whom are not CAs can not issue certificates to each other. These rules are implied by the name subordination rules.

Discussion

The first two PKI models are simple to describe and are generally well understood. However, our descriptions make their distinctions clear: the mesh model has no rules restricting the creation of certificate paths while the hierarchical model has the most restrictive set of certification path rules. The models following these are more difficult to describe. For example, it is not a simple matter to determine the essential features of the ICE-TEL model from [2] and [13] does not clearly outline all of the up-cross-down model features. This is not a specific criticism of these articles but a general criticism of the natural language description method for PKI models. Our method provides an unambiguous description, requiring only minimal additional technical knowledge of the reader to understand the language used. As the audience must already be exposed to the general (technical) PKI area we do not think that this is an unrealistic expectation.

In [13], another PKI model is presented, called the flexible-bottom-up PKI. This is the advocated PKI model from [13] and is developed from the up-cross-down PKI model. It employs the name constraints extension from X.509 within the up-cross-down model framework to add flexibility by circumventing the tight name subordination rules. Effectively, the rules E1, E2 and E3 would no longer apply and we are left with the mesh model[3]. When restrictive name constraints are imposed as in [13], it is similar to "pruning" in the mesh model and eventually the up-cross-down model is obtained. Finally, we note that both the up-cross-down and flexible-bottom-up models seem to necessarily impose that a single certificate policy is enforced throughout the entire PKI, especially if policy constraints are applied. The situation in regards to other X.509 extensions needs to be clarified given that the name constraints extension is being used.

As mentioned above, a formal description for the hierarchy model is already given in [11]. However, the focus of [11] is more on the operations of the PKI (for example, joining new members, revocation, etc.). In [11] the hierarchy rules enforced certain restrictions on the operations. Our results can be used to extend to other PKI models: we take a static view of the acceptable PKI certificate paths and define the PKI by the rules that are applied but this can be used to state that at all times the structuring rules of PKI must hold no matter how the certificate sets are varied (and thus extend the scope of [11]).

We explain further: to determine a legitimate certificate path there are two processes to be performed, path discovery and path processing. The path discovery algorithm returns all the possible paths, perhaps subject to some limits set by the certificate requester. The path processing algorithm determines that validity of each given path in regards to signature validation, policies, path restrictions imposed by CAs, etc. The models given here do not cover these operations of a PKI but rather the structuring rules. They can be used to determine whether a given alteration (addition of a new certificate or deletion of a certificate from the PKI) is acceptable. Also, if certificates carry PKI model identifiers then this would enable the optimal path discovery algorithm to be used. This gives the optimal solution to the path discovery problem. This is particularly advantageous when a single PKI is made up of a number of distinct PKI models all joined together as in the Bridge CA model.

4 Prospectives for a Global PKI

The example PKI models we introduced in Section 2 were selected as they provide a good sample set for our development in Section 3. However, they are all intended to be broad coverage PKIs or, in other words, service large user communities. In this section we consider implications for the development of a global PKI, based on previous experience. We propose adoption of a classification system for PKI models, based on formal descriptions (or rule sets). Firstly,

[3] It is noted in [13] the flexible-bottom-up model is, in the limiting case, no different from the mesh model.

we consider the past attempts to develop a PKI model suitable for the global situation, focusing on those models from Section 2.

Existing Experience
Attempts to construct a global PKI have so far not progressed to an actual realisation of a workable global PKI. Many problems have been identified, which are usually operational (i.e. certificate management etc.). For example, operational issues have convinced the PKI community that the mesh model is not an acceptable basis for a global PKI while the hierarchy model has problems with placing significant control and dependence on a single root CA. The ICE-TEL project met with some success (at least compared to previous projects) but was limited to a community of users comfortable with electronic communications technology and collaborations spanning institutional and national boundaries. An important aspect was the acceptance that users can control the public keys they trust (another capability existing in this community). The FPKI project seems to have been developed as the pragmatic solution to join together a wide number of U.S. federal agencies with existing commitments to pilot PKI projects. It is now being offered as a PKI solution for e-commerce security. This is important as the global PKI is itself likely to be built from the ground up in an unplanned (and unstructured) way (much like the development of the internet). The PKI models from [13] do not seem that distant from the mesh model when viewed in regards to the rules imposed on certificate path construction. However, they may be a possible next step from the bridge CA model to join a number of BCAs in a more manageable way then using a mesh model.

Desired Characteristics
At the highest level, we believe the wish list of characteristics for a global PKI would include the following:

1. the PKI should be connected,
2. the PKI should support multiple trust requirements,
3. the PKI should be flexible, to incorporate different user community needs.

Although a user will not have the need to securely communicate, with all other users it is not possible *a priori* to determine the subset of users they will ever communicate with motivating the first selection. The reasons for choosing two and three are similar to each other (flexibility) but as the implementation mechanics may be different, we differentiate between them. It is becoming widely accepted that trust identifiers should form part of a certificate (indicating the strength of management procedures employed). International codification of trust levels would make such certificates useful to a broader number of applications.

Future Perspectives
As PKI implementers have already experienced, development of technology outpaces the development of policy so the pragmatic solution will succeed (this seems to be the case with the FPKI). In this case it is unlikely that a connected PKI will ever be achieved. The best solution seems to incorporate a Bridge CA model with the ICE-TEL acknowledgement that users can decide to accept

certificates (or public keys). This gives a flexible solution where trust can be managed by end users. In this case the Global Internet Trust Register [1] provides an important end user service. The Bridge CA model allows joined PKIs to select their own internal PKI model. As the global PKI develops it may be necessary to further expand the BCA into a structured mesh model, along the lines of the up-cross-down or flexible-bottom-up models. These add flexibility while maintaining some structure.

In this case the global PKI will be a complex interconnection of multiple PKIs employing distinct PKI models to match their business needs. We refer to such PKIs as *conglomerate* PKIs. Given this view of the global PKI future, it is useful to place identifiers in certificates which state the PKI model that is used for the descendants of a given certificate. This would allow further refinement of the set of acceptable certificate paths. Suppose $\texttt{MODEL}_m$, $m \in \{1, \ldots, n\}$, is the PKI model identifier used for $\texttt{PKI}_i$ for $i \in \{1, \ldots, k\}$.

D6 $\forall cert \in \texttt{CERT} \bullet I(cert) \in des\_set(\texttt{PCA}_i) \Rightarrow \texttt{MODEL}_m$

The PKI model identifier can map to the set of rules (from Section 3 or rule sets developed elsewhere) which describe the acceptable certificate paths. The PKI model identifier would also allow the selection of the most efficient certificate path discovery algorithm in a straight forward way leading to an optimal path discovery algorithm.

5 Conclusions

We have used a very simple, general description to accommodate the different PKI models based on the restrictions placed on acceptable certificate paths (concerned with structuring). Our description does not rely on natural language alone but is supported by the use of formal language. The methods employed are easily accessible to a general audience with some technical skill but no more than would be expected in the general PKI area. In this way, ambiguities or omissions in the descriptions are avoided and clarity achieved. Using this basis, we have considered several existing PKIs. Our PKI model development has led to the idea of using PKI model identifiers in certificates when joining multiple PKIs employing different PKI models into a conglomerate PKI. This provides a way to achieve optimal path discovery algorithms across conglomerate PKIs. Combining our work with the results of [11] gives further benefits of extending results for PKI operations to the models described here. Finally, we motivate our direction for stricter definitions (and classification) of PKI models by arguing that the conglomerate PKI structure is most likely for the global situation.

References

1. R. Anderson, B. Crispo, J. Lee, C. Manifavas, V. Matyas, F. Petitcolas, *The Global Internet Trust Register*, MIT Press, 1999.

2. D. W. Chadwick, A. J. Young, and N. K. Cicovic, *Merging and extending the PGP and PEM trust models: the ICE-TEL trust model*, IEEE Network, **11(3)**, 16–24, 1997.
3. Defense Information Systems Agency, *State Analysis of Certification Path Processing Procedures*, June 2000.
`http://www-pki.itsi.disa.mil/certpathproc.htm`
4. C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen, *SPKI Certificate Theory*, Request for Comment 2693, September 1999.
`ftp://ftp.isi.edu/in-notes/rfc2693.txt`
5. Federal Public Key Infrastructure Steering Committee
`http://www.cio.gov/fpkisc/`
6. M. Henderson, M. Burmester, E. Dawson, and E. Okamoto, *Weaknesses in Public Key Infrastructures*, Proceedings of the First Workshop on Information Security Applications (WISA 2000), November 2000, 53–66.
7. R. Housley, W. Ford, T. Polk, and D. Solo, *Internet X.509 Public Key Infrastructure Certificate and CRL Profile* Request for Comment 2459, 1999.
`http://www.ietf.org/html.charters/pkix-charter.html`.
8. R. Housley and T. Polk, *Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructure*, John Wiley and Sons, 2001.
9. ITU-T Recommendation X.509, *Information Technology - Open Systems Interconnection - The Directory: Authentication Framework*, June 1997 (equivalent to ISO/IEC 9594-8, 1997).
`http://www.imc.org/ietf-pkix/mail-archive/msg04337.html`)
10. S. Kent, *Privacy Enhancement for Internet Electronic Mail, Part II: Certificate-Based Key Management*, Request for Comment 1422, February 1993.
`http://www.ietf.org/rfc/rfc1422.txt?number=1422`
11. C. Liu, M. Ozols, M. Henderson, and T. Cant, *A State-Based Model for Certificate Management Systems*, Public Key Cryptography: Third International Workshop on Practice and Theory in Public Key Cryptography (PKC 2000), Lecture Notes in Computer Science, **1751**, 2000, 75–92.
12. M. Ozols, M. Henderson, C. Liu, and T. Cant, *The PKI Specification Dilemma: A Formal Solution*, Proceedings of the 5th Australasian Conference on Information Security and Privacy (ACISP 2000), Lecture Notes in Computer Science, **1841**, 2000, 206–219.
13. R. Perlman, *An Overview of PKI Trust Models*, IEEE Network, **13(6)**, 38–43, 1999.
14. P. R. Zimmermann, *The Official PGP User's Guide*, MIT Press, Cambridge, Massachussets, 1995.