Merging and Extending the PGP and PEM Trust Models - The ICE-TEL Trust Model

Authors: D W Chadwick, A J Young, University of Salford, England, and
N Kapidzic Cicovic, COST Computer Security Technologies, Sweden

Abstract

The ICE-TEL project is a pan-European project that is building an Internet X.509 based certification infrastructure throughout Europe, plus several secure applications that will use it. This paper describes the trust model that is being implemented by the project. A trust model specifies the means by which a user may build trust in the assertion that a remote user is really who he purports to be (authentication) and that he does in fact have a right to access the service or information that he is requesting (authorization). The ICE-TEL trust model is based on a merging of and extensions to the existing Pretty Good Privacy (PGP) web of trust and Privacy Enhanced Mail (PEM) hierarchy of trust models, and is called a web of hierarchies trust model. The web of hierarchies model has significant advantages over both of the previous models, and these are highlighted here. The paper further describes the way that the trust model is enforced through some of the new extensions in the X.509 V3 certificates, and gives examples of its use in different scenarios.

Note. In this paper both *he* and *his* are used to denote the singular, in order to specify precise semantics, but do not imply any gender association.

Merging and Extending the PGP and PEM Trust Models - The ICE-TEL Trust Model

Introduction

Public key cryptography is widely recognized as being a fundamental technology on which several essential security services can be built, such as authentication, integrity, non-repudiation and confidentiality. But in order to give users the necessary level of assurance (or trust) that the technology is actually providing the security service that they think it is, much work still needs to be done in specifying and implementing the security policies, procedures and infrastructures that underlie the use of the technology.

The original use of public key certificates, as proposed in [15], was to securely bind a user's name to his public key, thus providing a means of authentication. This use was subsequently adopted by X.509 Version 1 certificates [1], and PGP certificates [5]. The application receiving such a certificate could answer the question "who is wanting to access the service" (i.e. authentication) but could not answer the supplementary question "which (level of) service is this user allowed to access" (i.e. authorisation or access control) without recourse to additional, usually internal, access control information. More recent work, by the IETF Simple Public Key Infrastructure (SPKI) working group [16], and the ISO/ITU-T X.509 Version 3 Certificates Draft

Amendment [10], allows information to be contained in certificates that allows both of these questions to be answered.

Some early public key infrastructures are already in the process of being established. For example, Netscape distribute a list of Certification Authority (CA) public keys with their browser software, that allow the browsers to validate the signatures on incoming messages from the server. Several organisations, e.g. VeriSign [2], act as trusted third parties and offer a certification service to users and organisations, that allow them to have their public keys certified. For users of PGP, PGP servers distribute PGP public keys, certified by other PGP users, to anyone who requests them. These servers are mirrored at various locations around the globe [13]. In the commercial world, Visa and Mastercard are defining the Secure Electronic Transactions (SET) [12] specifications, that allow credit card payments to be made securely over the Internet using tailored versions of the X.509 V3 certificates. We should see the deployment of this technology during 1997.

The EC IV Framework ICE-TEL project [3] is a 3.26Mecu (US$ 3.75M) two year project running from December 1995 to November 1997. The main aims of the project are to establish a scaleable public key certification infrastructure throughout Europe, and to build secure applications such as S/MIME and WWW that can utilize it. This will allow academics and researchers (primarily although not exclusively) to securely transfer information between themselves using public key cryptography as the fundamental security technology. By "securely transfer information between themselves" we mean such things as: the participants can have a reasonable level of confidence that the information that they have received did really come from the purported sender, or that the encrypted information that a researcher is about to send can only be read by the intended recipient, or these remote users can only have access to these Web pages etc. There is no intention at the moment to use this infrastructure for commercial transactions, and the SET and ICE-TEL infrastructures will remain independent in the foreseeable future.

The Importance of a Trust Model

The user has to trust that the components that provide a particular security service are reliable and trustworthy (up to a certain point, that is, since trust is rarely absolute). The components that provide a service such as authentication or confidentiality comprise not only the software, the hardware, and the network that enable the computers to communicate, but also the established security policies, procedures and infrastructures that are operated by the various parties. For example, a user might be using software and hardware with DoD Orange Book [4] class C1 assurance, running over a private network, but this in itself is not sufficient to guarantee the authenticity of a message received from a remote user. Suppose there is a remote certification authority associated with some remote users on this network, and that this CA has been entered into the local system as being trustworthy by the local administrator. If in fact, the remote CA is willing to certify the public keys of its users without asking them for any formal proof of their identities (for example a floppy disc containing a public key is sent through the internal mail with a typed memo) then these remote users may pose as other employees. A digitally signed message that a local user receives from 'remote researcher 1', will be verified by the trusted software as originating from 'remote researcher 1', but it could actually have been produced by anyone who contacted the remote certification authority claiming to be ‘remote researcher 1’ and was duly issued with a certificate. The local user would be none the wiser that a masquerade has taken place. It is the establishment of trust in the certification infrastructure, and through this, trust between users, that is the subject of this paper. The trust model is the mechanism used to describe how this trust may be established. The way that the trust may be implemented and enforced by the validation software in the user’s end system, is also described. This relies on some of the new V3 extensions to X.509 certificates.

Advantages and Deficiencies in the PGP trust model

PGP [5] is probably the most widely used public key cryptographic package today on the Internet. Although post-dating PEM [6] by a number of years, its deployment was much more rapid due to a number of factors. One factor was that the software was freely available to academics and researchers in the US from inception, and a non-copyright version soon became available to the rest of world [13]. Although a similar situation existed for PEM software, the one significant advantage that PGP had, was that no certification infrastructure was needed before anyone could use PGP in a secure manner. Two users can simply download a copy of the software, generate their own key pairs, exchange their public keys by some private means (for example at a meeting), and trustworthy security services (authentication, integrity and confidentiality) are there for immediate use. PGP’s trust model, that always starts from the user (*“now who do I trust?”*) naturally leads to free and unrestricted organic growth of the user base. The PGP trust model has variously been described as a public trust model [14], and a web of confidence [5.3], since any member of the public may set up their own trust chains to other PGP users in any way that they wish. This leads to a web of trust chains being set up between the various members of the global community of PGP users.

Ironically the success of PGP is in some ways its downfall. The method of key distribution (a central mirrored server), and the associated web of trust that users build for themselves, is not easily scaleable when hundreds of thousands of users are involved. It is also ironic that the PGP trust model allows for certification authorities, but does not call them that - the term “trusted introducers” is used instead. The problem here is how do you know if an introducer can be trusted or not, and if he can be trusted, how much can he be trusted. More recent proposals e.g. to SURFnet [7], have planned to build certification authorities and certification hierarchies into PGP to replace the trusted introducers. But because of the simplicity of the PGP certificate structure, for example certificates do not have a period of validity (they are infinitely valid), and they have to be revoked by the user not by the CA (which can be difficult if a user loses his private key!), awkward structures such a “year keys” had to be invented, and users are advised to create revocation certificates immediately after key generation. These factors are inhibiting the growth of the PGP community. Finally, a PGP key, whilst enabling authentication, gives a user no indication of the authorizations associated with key, or the uses to which the key might be put. This is left up to the discretion of the key user.

Advantages and Deficiencies in the PEM/X.509 trust model

By contrast, X.509 [1, 8] has a more rigorously thought out certificate structure than PGP, that does not suffer from the latter’s deficiencies, although the added complexity of the Version 3 certificate [10] creates interworking problems of its own (see later). X.509 was originally designed on the premise that trusted third parties, called Certification Authorities (CAs), would exist, and that certified public keys, or certificates for short, would be the normal mechanism used for distributing public keys. The intent was always to be able to cater for millions of users on a global basis. It was also envisaged that the global X.500 directory [9] would be the natural way of publishing and distributing a user's public key certificate, although the global X.500 directory has still to become a reality. The X.509 standard does not insist on any particular certification infrastructure, since as an International Standard it should be able to cater for all requirements. Both hierarchical and network models of CAs are catered for. This general approach to CA infrastructures is perfectly proper to have in an International Standard, but the disadvantage is that it is not specific enough to allow an operational CA infrastructure to be quickly or easily implemented by a community of users. Restrictions need to be introduced, that will allow both trust and certification paths to be more easily established between communicating parties. The PEM document [6.2] was the first attempt at a functional standard for X.509.

Unfortunately PEM was too long in the making, taking nearly 5 years to complete the IETF standardization process. It was overtaken be events, and so was never really deployed to any great extent. It is unlikely now that it ever will be, although there are some significant pockets of users that are still using this technology. One of the primary drawbacks to a quick wide scale deployment of the PEM certification infrastructure, is that, unlike PGP, it does not easily support organic growth. Two users cannot securely interchange messages after downloading PEM software, because they first need to have their public keys certified by their local CAs, and their CAs need to be certified by a Policy CA, which itself needs to be registered by the Internet Policy Registration Authority (IPRA). The latter was never really established by the IAB, and most organizations do not operate a CA. Hence the inertia in establishing a PEM based certification infrastructure.

Nevertheless, X.509 does define a standard, reasonably well thought out certificate structure, that allows for: validity times (the debate over its 2 digit years has recently been resolved), certificate serial numbers, algorithm identifiers and any number of extensions, that are all digitally signed by the CA. Whilst the role of the original certificate was only to aid authentication, the Version 3 certificate, in addition, defines a general certificate extension mechanism and a core set of extensions that allow authorizations and security policies to be bound into a certificate as well as authentication. Furthermore, the same basic certificate structure may be used by a myriad of different users, which means that generic software can be purchased and tailored by individual groups thereby reducing overall implementation costs. The downside of this of course is that each set of users will have to specify their own (different) functional standards, deciding which particular set of core extensions to use, and which domain specific extensions to create, and these implementations will not then interwork. Two well known groups that are currently producing their own independent X.509 functional standards are the IETF Internet Public Key Infrastructure (PKIX) working group with its Internet Draft [11], and the Visa/Mastercard consortium with its SET specifications [12]. These specifications are designed for different functional purposes i.e. generic Internet use and Internet credit card transactions, and therefore cannot be expected to interwork. The ICE-TEL project has also developed its own functional standard [17]. This specifies a subset of the PKIX extensions, and is therefore less costly to implement, but never the less it is anticipated that ICE-TEL and PKIX implementations will still be capable of interworking.

The X.509 V3 extensions allow for certificates to be revoked by the issuing (or other) CA, and for the distribution point of the certificate revocation lists (CRLs) to be published inside the certificate. Cross certification between CAs is also allowed. But the standard does not provide any guidelines on the use of CRLs, cross certificates or certification paths, so the users of X.509 have to provide these procedures themselves. The net result of this, is that it takes a lot longer to establish X.509 user communities than PGP user communities. This is not something that the ICE-TEL project wished to perpetuate.

Advantages and disadvantages of the SPKI trust model

SPKI is a relatively new IETF working group, formed in 1996. At the time of writing they had not published their first Internet Draft (let alone Internet Standard), although working documents are available on the Internet [16]. The aim of the SPKI group is to produce a certificate format and associated protocols that are simple to understand, implement and use. For this reason they do not use ASN.1 [18], the language used to specify X.509 certificates, which is often criticized for being difficult to understand, inefficient to implement, and costly to use (i.e. the ASN.1 compilers are not free for commercial use). It is true to say that ASN.1 is not the preferred language of choice for many Internet engineers, ASCII encoding remaining their firm favourite. Another primary aim of the SPKI group is to allow a certificate to provide for trusted authorization as well as trusted authentication. The primary disadvantage of this work from the ICE-TEL perspective is its immaturity - the work started after the ICE-TEL project - but the ability to include authorizations as well as authentication in SPKI certificates is clearly an advantage over PGP and X.509 V1 certificates. Fortunately the X.509 V3 certificate also has the ability to incorporate authorizations in its structure (either directly or indirectly as decided by the user community) and so this is the mechanism used by the ICE-TEL trust model.

Using the best from all trust models - the ICE-TEL trust model

Requirements of the ICE-TEL Trust Model

Based on the experience gained previously with the other trust models, the following requirements can be listed for the ICE-TEL trust model:

1. The trust model should be capable of operating without the use of certificates or CRLs, through the trusted exchange of public keys. Users should be able to determine who they trust, without the forced imposition of any certification infrastructure.
2. Where certificates are used, the trust model requires the use of the X.509 standard V3 certificates and V2 CRLs, since these are a standard and do not have the deficiencies of Version 1. Earlier versions of the certificate and CRL may be supported by implementations while they remain in widespread use, though these may not provide access to all the facilities of the trust model, in particular, an

indication of what the public key may be used for (i.e. authorization). Proprietary certificate and CRL formats will not be supported.

3. The trust model should allow for the creation of security domains encompassing
   - single users
   - multiple users (small/simple organizations); and
   - arbitrarily complex organizations.
4. The trust model should allow for organic growth among the users and organizations; and should allow security domains to grow, shrink and re-organize at any time with minimum inconvenience to the users within the domain and to people communicating with those users.
5. The trust model should allow users and/or the administrator of a security domain to choose which other domains are to be trusted. There is no requirement that inter-domain trust is mutual and there are no points that any domain or user is required to trust. Inter-domain trust is not transitive, unless specifically allowed by the administrator.
6. The operation of the trust model as a whole should not depend on the existence and operation of any single part of the deployed infrastructure. In particular, there need not be a central top level registration authority like the PEM IPRA.

Components of the ICE-TEL trust model

CAs and Security Policies vs. Users

The primary difference between a user or a PGP introducer and a CA, is that a CA publishes a security policy that states the principles under which it operates. These principles should include statements about:

- how the CA performs user authentication before issuing a certificate
- how the CA protects its private key from disclosure
- when, under what circumstances and how a CA publishes its Certificate Revocation Lists (CRLs)
- what the assurances and liabilities the CA offers with the certificates it issues, and for what purposes they should be used.

Security policies are currently published as free form text documents. A number of them are available on the Internet e.g. [19],[20]. It is by understanding the security policy of a domain, that a security administrator (or sophisticated user) can determine the amount of trust to be placed in the certificates issued by a remote CA, and conversely, how much trust can be placed by remote administrators (and/or sophisticated users) in the certificates issued by the local CA. Security administrators (and/or sophisticated users) also need to determine for themselves how much trust they can place in the CA itself, and in the fact that the CA actually operates according to its published policy (but note that methods for checking this are beyond the scope of the current paper). The assumption of the ICE-TEL project is that security administrators (and/or sophisticated users) will by off line means determine which CAs and their policies can be trusted, and the identifiers of these policies will be configured into the local systems (the users' Personal Security Environments - see next section) along with the public keys and names of those CAs. In X.509 V3, policies are identified by globally unique numbers called object identifiers. Object identifiers are freely available to organizations[1], and can be assigned to the various security policies that they operate.

Personal Security Environment

Each user has a Personal Security Environment (PSE), in which are stored[2] the public keys and names of the users that he trusts, and the names and public keys with their associated policy ids of the CAs that he trusts. (A PSE is similar to a PGP user's public key ring.)

Policy ids form a new extension in the X.509 V3 certificates (the *certificate policies field*), so that each certificate may carry around with it an identifier of the security policy under which it was issued. When validation software is checking the validity of a certificate, it will check that the policy id in the certificate is the same as the policy id in the user's PSE. In this way the validation software enforces the trust that is configured into the user's PSE[3], since certificates containing unknown policy ids can be rejected because the user (or his administrator) has not said that he trusts this unknown policy[4].

Figure 1 shows a user (User 1) who trusts his organization's CA (OrgCA), and has added the public key and policy id of this CA into his PSE. Figure 2 shows a user who trusts a pan-organization CA (PCA), and has added the public key and policy id of this CA into his PSE.

[1] It does not matter which object identifier is used by an organization as the root of its number space, since it is only a number with no internal meaning attached to it. For example, an organization that obtains an IANA number of n, automatically has the object identifier of 1.3.6.1.4.1.n allocated to it, by virtue of the way object identifiers are assigned. It may then assign object identifier components below this point to security policies (and other objects) as it pleases.

[2] It is a local policy issue who is allowed to update the PSEs of the users. In some organizations, only the security administrator will be allowed to update the users' PSEs. In other organizations, sophisticated users will be allowed to manage their own PSEs.

[3] The SET specifications have gone one step further than that proposed by the ICE-TEL project, in that their certificates also contain a cryptographic hash of the policy. This stops a policy being changed after a certificate has been issued. When a policy is retrieved it can be hashed to ensure that the policy identified in the certificate is same as the policy just retrieved.

[4] The policy id is not strictly necessary for a security domain that only ever supports one security policy, since this can be implicitly assumed from the CA's public key. Policy ids are essential when more than one policy is used by a CA.

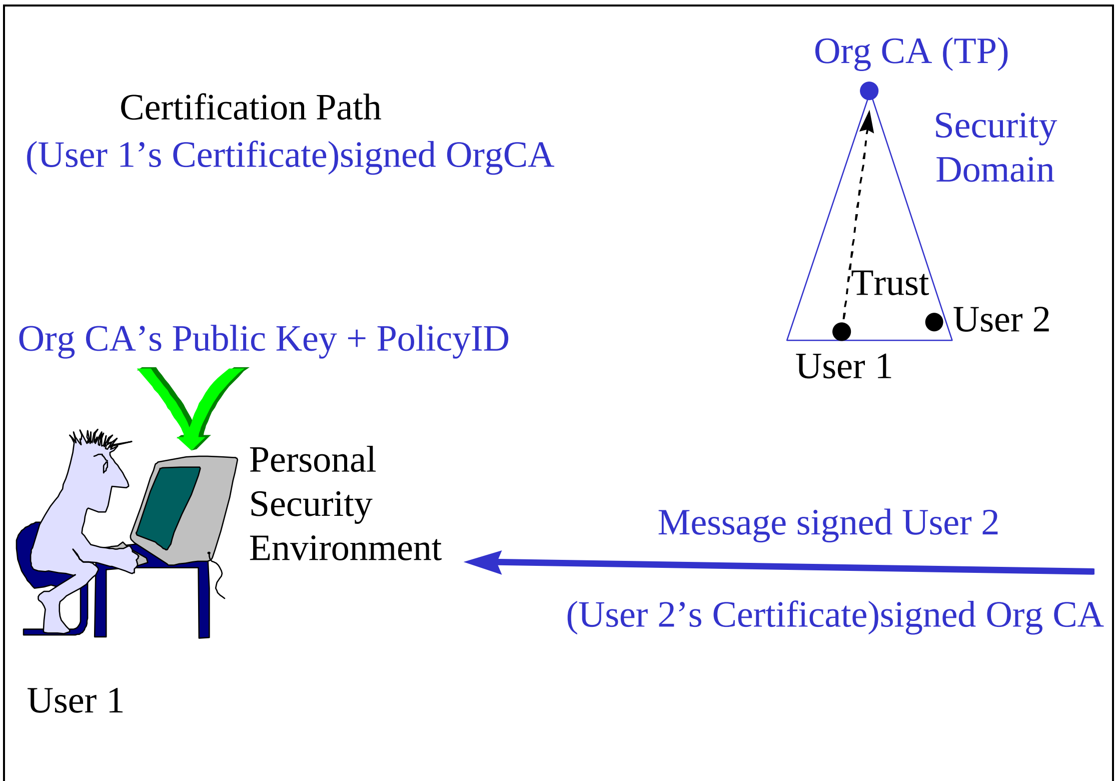

Figure 1 Organisation Based Trust

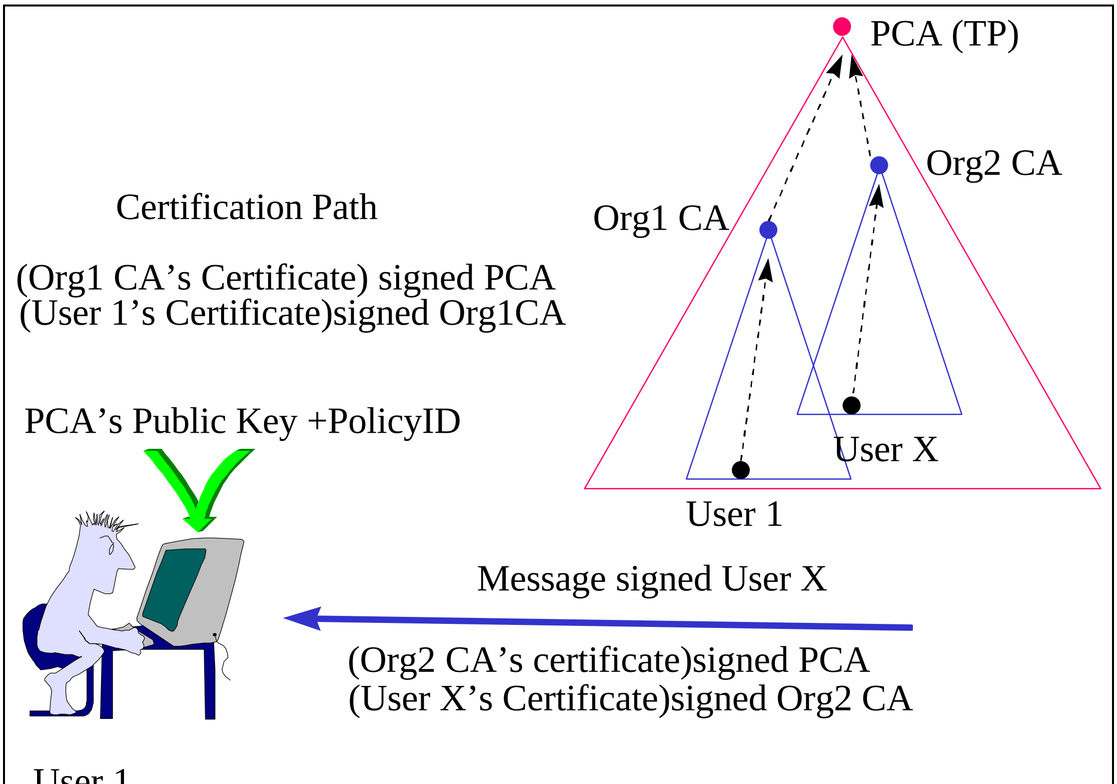
PCA (TP)
Org2 CA
Certification Path
Org1 CA
(Org1 CA's Certificate) signed PCA
(User 1's Certificate)signed Org1CA
PCA's Public Key +PolicyID
User X
User 1
Message signed User X
(Org2 CA's certificate)signed PCA
(User X's Certificate)signed Org2 CA
User 1

Figure 2 Multi-Organisation Based Trust

If a user operates in an environment without CAs, then he simply adds the public keys of the users that he trusts into his PSE, and no policy id are allowed, since the remote users are by definition not a CA, and therefore cannot issue certificates. The user's security validation software can ensure that it does not trust certificates issued by a public key in the PSE that does not have an associated policy id, since this must be a user signed certificate. This is one of the fundamental differences between the ICE-TEL trust mechanism and the PGP trust mechanism, in that CAs are clearly distinguished from users, and must publish their security policies before they can be trusted. In this way, a user cannot masquerade as a CA and issue certificates, without being caught. Furthermore, the X.509 V3 certificate carries an indication in each certificate (the *basic constraints field*) as to whether the certified subject is a user or another CA. This stops a user who has been certified by a CA from posing as a subordinate CA within that domain.

Trusted Point

A security domain encapsulates a collection of objects that all abide by the rules defined in its security policy. In this context, objects are users and CAs. A security domain can be as small as a single user ("I determine my own policy") or as large as a collection of many CAs and their users e.g. the whole European academic community serviced by ICE-TEL. Each security domain has at its apex a single CA (or user, in the case of a single user domain), called the trusted point. This trusted point, if a CA, may certify both users and subordinate CAs within the domain confirming that they all abide by the same overall security policy. The name and public key of the trusted point is known to all the objects (CAs and users) in the domain. The public key will initially be distributed by some out of band proprietary means, and the ICE-TEL model does not mandate any specific mechanism for this. This situation may change in the future, as the PKIX group is currently defining a standard data structure for the export of a CA's public key [21]. Needless to say, when the public key of a trusted point, and its associated policy id, are configured into a user's PSE, this is effectively making the statement that the user trusts every object in this domain for the purposes identified in the policy. If perversely a user does not trust the CA that certified his own public key, then he would not add the CA's key to his PSE. This would result in him not trusting other users certified by his CA, but the other users who do trust the CA would still trust his public key certificate.

In Figure 1, the CA 'OrgCA' is the trusted point, and in Figure 2, the CA 'PCA' is the trusted point.

Large domains may have a requirement for many CAs, and the model encompasses this. The only requirement is that all the CAs comply with the terms of the security policy specified at the trusted point, and that all the users who trust this point, as a minimum:

i) know the public key of the trusted point CA and
ii) have a certification path to this CA (see next section).

One can see that the PEM model of a Policy CA maps into the ICE-TEL model of a trusted point CA. In Figure 2, a PEM Policy CA (PCA) spans many organization based CAs, who all conform to the same security policy. Users hold the public key and policy id of the Policy CA in their PSEs.

In the situation where a subordinate CA actually implements a stronger security policy than that of the trusted point, this is catered for through the concept of an embedded high security domain (see below). In the ICE-TEL model it is not possible for a subordinate CA to implement a weaker security policy than that of the trusted point.

Certification Path

A certification path is a mechanism to allow remote users to establish trust in your public key, even though it is not stored in their PSE, and vice versa. A certification path comprises a chain of certificates, starting with a certificate signed by a CA that you trust, and ending with a certificate of the remote user[5]. In the ICE-TEL model, the first certificate is signed by the CA at the user's trusted point, and ends with the user's certificate. The shortest certification path that can be created consists of a single certificate, which is a user's public key signed by his CA, with the CA being the trusted point of the security domain (for example, as in Figure 1). Each user in a security domain is given his certification path to the trusted point of the domain. The user makes this certification path as freely available as he wants, in order to allow people to securely interact with him. For example, the certification path can be published in a directory, or a Web page, or can be included in outgoing signed messages. Since the certification path comprises a sequence of certificates, which are all signed and inter-linked, it cannot be tampered with without detection, and therefore does not need to be unduly restricted or protected. In particular, it does not need to be kept inside a user's PSE, which was a mistake that some early PEM implementations made.

When each user is given his certification path to his trusted point, this allows all users within the domain to securely interact with each other. For example in Figure 1, User 1 is able to validate the signature on the signed message from User 2, even though he had no prior knowledge of User 2's public key, since the message contains the certification path. In Figure 2, User 1 is similarly able to validate the signed message from User X who works for another organization, since both organizations conform to the same overall security policy.

Cross Certification

A user or his administrator could theoretically control which remote domains to trust, by vetting each one in turn, and configuring the public keys and policy ids of their remote trusted points into his PSE. Whilst this is allowed in the ICE-TEL trust model, it is obviously inefficient. A more efficient way is for a user to delegate this responsibility to the security administrator of his local trusted point. The security administrator can then vet the remote domains on behalf of all the users in the local domain who trust him to do so, and issue a cross certificate for each remote domain that he trusts. A cross certificate is simply the signed public key (and other information) of a remote CA, issued by the local CA. The cross certificate needs to say what policy in the remote domain has been vetted, and that only certificates issued according to this policy should be trusted by the local domain. This trust is enabled via new extensions (the *policy mappings* and *policies constraints* fields) in X.509 V3 certificates. The *policy mappings* field contains the policy id of the local domain, and the id of the equivalent policy in the remote domain that is to be trusted. The local administrator can mandate that all the certificates in the certification paths to the trusted point of the remote domain must contain the policy id of the remote domain (via the *require explicit policy* field of *policies constraints*), in order to ensure that the

[5] A certification path is not a meaningful concept for a remote single user domain.

same trusted policy was used throughout the remote domain. Validation software in the user's workstation checking the validity of a certificate from a remote domain, now has an unbroken chain of trust to the remote user's public key, starting with the key and policy id of the local domain's trusted point, mapping into the key and policy id of the remote domain's trusted point, and finishing with the certification path of the remote user, in which all the certificates contain the same remote policy id.

An additional V3 extension (the *name constraints* field) allows the local administrator to further constrain the trust that he is conferring to the remote domain, via the cross certificate. The local administrator can actually limit the number of users in the remote domain that are to be trusted, by specifying (via *name constraints*) a subset of the user names from the remote domain whose certificates are to be trusted by the local domain. This might be useful if a remote domain is large, but the local administrator only wants to trust a subset of their users, such as the sales department.

A *delegation* variable set by the user on the public key/policy id of a trusted point in his PSE, tells the validation software whether the user has fully or partially delegated trust to the administrator of this CA for the purposes of cross certification. A value of 0 signifies that the CA (administrator) is not trusted to cross certify other CAs (i.e. no delegation of trust by the user; the CA is only trusted to certify objects in its own domain). A value of 1 indicates that the CA (administrator) can cross certify other CAs but trust is not transitive to the other domain (i.e. delegation of trust for cross certification purposes is to this CA administrator only, and the cross certified remote CA administrator is not trusted to further cross certify other domains). Positive values greater than 1 recursively delegate or transfer trust to remote CA administrators that preceding administrators trust; the depth of delegation depending upon the size of the integer. A new X.509 V3 extension (*inhibit policy mapping*) (also an integer) is similarly used by a CA administrator to delegate cross certification trust (or not) to a remote CA administrator. A value of 0 indicates that trust is not delegated. Therefore, for cross certification trust to be passed to the CA administrator of a remote security domain, both the user and the local CA administrator have to delegate trust (as well as any intermediate trusted CA administrators).

There is a requirement placed on the CA at a trusted point to make available (i.e. publish) the list of cross certificates that it has issued. This can be done in a number of ways, e.g. through its Web page, directory entry, or via E-mail requests. A similar requirement is placed on CAs to publish their CRLs (see next section) and the same mechanism may be used for both. The ICE-TEL trust model does not mandate any particular mechanism. It is currently left up to each software provider to decide the best mechanism to use, according to customer demand. User's should be able to download cross certificates at will, either for immediate use or for caching. In the latter case, if a user is unable to validate a particular signature because the public key at the start of a certification path is not available in his PSE or local cache, the software should be able to retrieve the latest cross certificates from his trusted point(s) in real time.

Certificate Revocation Lists (CRLs)

If a user's public key becomes comprised, or is lost or stolen, then the CA has to revoke the user's certificate, and issue a new one containing the new replacement public key. The list of revoked certificates are published by the CA, in a certificate revocation list (CRL), that is itself signed by the CA to prove its validity. The CRL may be published in a variety of ways, e.g. via the Web or a directory entry. A new extension in X.509 V3 certificates (the *CRL distribution points* field) allows each user certificate to hold pointers to the places where the CRL will be found i.e. the distribution points. A distribution point can be an RFC 822 name, a DNS name, an X.500 distinguished name, a Web URL or a variety of other places. Certificate validating software in the user's workstation needs to have access to the latest CRLs before confirming that a certification path is trustworthy. It is for each user to determine whether he is willing to trust a remote public key based only on periodic retrieval and caching of CRLs, or on the real time retrieval of them. The frequency of retrieval needs to be a configurable parameter of the user's PSE.

Embedded High Security Domains

Oftentimes there will be a unit within an organization, or an organization within a community, whose members are more trusted for certain tasks than the other users of the group. For example, the members of a CERT would be more trusted to issue security incident reports than normal users of a network. The more highly trusted users can be regarded as being members of a high security domain with the normal users being members of a low security domain. A high security domain is more trusted than a low security domain, because not only does it implement all the requirements of the security policy of the low security domain, but in addition, places additional criteria and constraints on its certification practices and policies. The correspondingly higher trust that can be placed in the members of a high security domain means that the authorizations that the domain bestows on them will be correspondingly greater than on those of a lower security domain. Clearly this could be handled by specifying both domains as independent security domains, conforming to different security policies. But this will require cross certification of the high security domain by the low security domain (and also possibly vice versa) before certificate validation can take place, and cross certification is more awkward for users to handle, due to the need to retrieve cross certificates.

A more efficient way of handling this common scenario, is to allow for high security domains to be embedded in low security domains. This scenario is supported in the ICE-TEL trust model, by issuing users in the high security domain (who wish to communicate with users in both the low and high security domains) with two logical certification paths: one originating from the trusted point of the low security domain and containing the policy id of the low security domain, and one originating from the trusted point of the high security domain and containing the policy id of the high security domain. There are two different ways in which the high security CA can implement this scenario, and certify each of its users. It can place the policy ids of both the low security and high security policies in each user certificate it issues, since its certification procedures conform to both policies. In this case the user has only one physical certification path and one certificate. Alternatively, it can issue two certificates and two certification paths for each user, with a different policy id in each[6]. Both of these alternatives are shown in Figure 3.

Whenever a user in the high security domain sends a message to a user in the low security domain, the low security certification path is appended. These messages will be trusted by users in the low security domain just as if they had originated from any other member of the low security domain. The salient point is that the low security domain users will not have had to alter their PSEs in order for this to work (or indeed take any other action, such as having to retrieve cross certificates, which would have been the case if the domains had not been embedded). In addition, if the users in the low security domain were to add the public key and policy id of the high security domain CA to their PSEs, then messages from users in the high security domain with the high security certification path appended will be given a higher level of trust conversant with the high security policy.

Whenever a user in the low security domain sends a message to a user in the high security domain, it will not be trusted by members of the high security domain, unless the public key and policy id of the low security CA are added to the PSEs of the users in the high security domain. Even then the messages will only be trusted for a limited set of actions, conversant with the policy of the low security domain. The completely non-trusting scenario is depicted in Figure 3. Since User 1 has only the public key and policy id of the high security CA (HSPCA) stored in his PSE, he will not trust messages originating from users in the low security domain.

Whenever a user in the high security domain sends a message to another user in the high security domain, the high security certification path is appended. The receiving user will trust this message, because he has the key of the high security trusted point stored in his PSE.

[6] By extending this approach, it is possible for a user to be a member of two completely disjoint security domains by having his key certified by two different CAs that support completely different security policies.

Certification Paths

Either (User 1’s Certificate $_{\text{HS Policy}}$)signed HSPCA
+
(HSPCA’s Certificate $_{\text{LS Policy}}$) signed LSPCA
(User 1’s Certificate $_{\text{LS Policy}}$) signed HSPCA

Or (HSPCA’s Certificate $_{\text{LS Policy}}$) signed LSPCA
(User 1’s Certificate $_{\text{LS Policy , HS Policy}}$) signed HSPCA

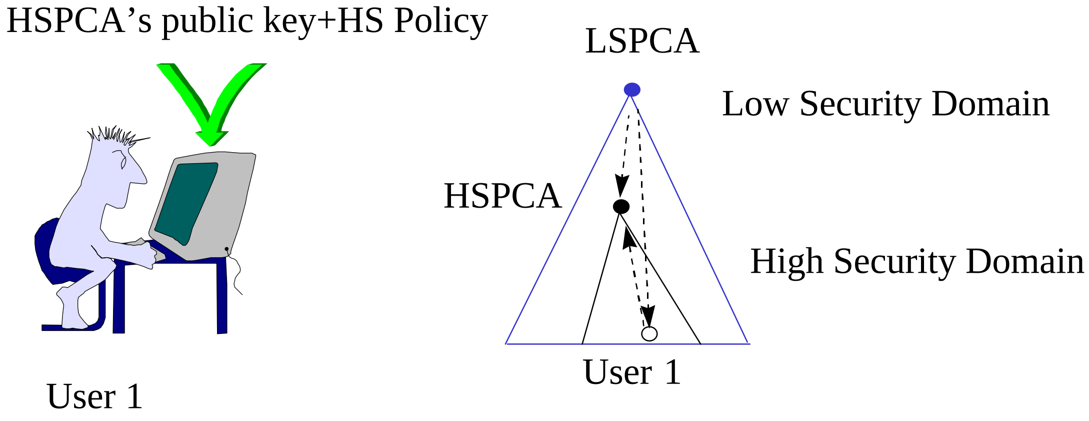

Figure 3 Embedded High Security Domains

The ICE-TEL Trust Model - An example

Figure 4 represents a scenario where there are 3 autonomous security domains, with CAs PCA1, PCA2 and PCA3 at each of the trust points, and an independent user, User 4, who is not attached to any certification infrastructure. User 1 has decided that he trusts User 4 and has entered her public key into his PSE. User 1 also trusts his trust point CA, and has entered the public key and policy id of this into his PSE. User 1 is unaware of the other security domains, but has delegated authority to the administrator of his domain to cross certify any that are in the latter’s opinion trustworthy. This administrator has checked the policy of the PCA2 domain, and decided that it is trustworthy. Consequently, PCA1 has cross certified the public key of PCA2, and stored the cross certificate locally.

Suppose User 1 receives signed messages from User 2, User 3 and User 4. The signature from User 4 can be immediately verified and trusted from the contents of User 1’s PSE. The other two messages cannot be. The User 1’s software will have to retrieve the set of cross certificates that have been issued by his CA (PCA1). The set will contain one cross certificate, that of PCA2, and so the signature from user 2 can be verified and trusted. The signature from User 3 cannot be verified, since User 1 has no trust path to the public keys of either User 3 or her CA (PCA3).

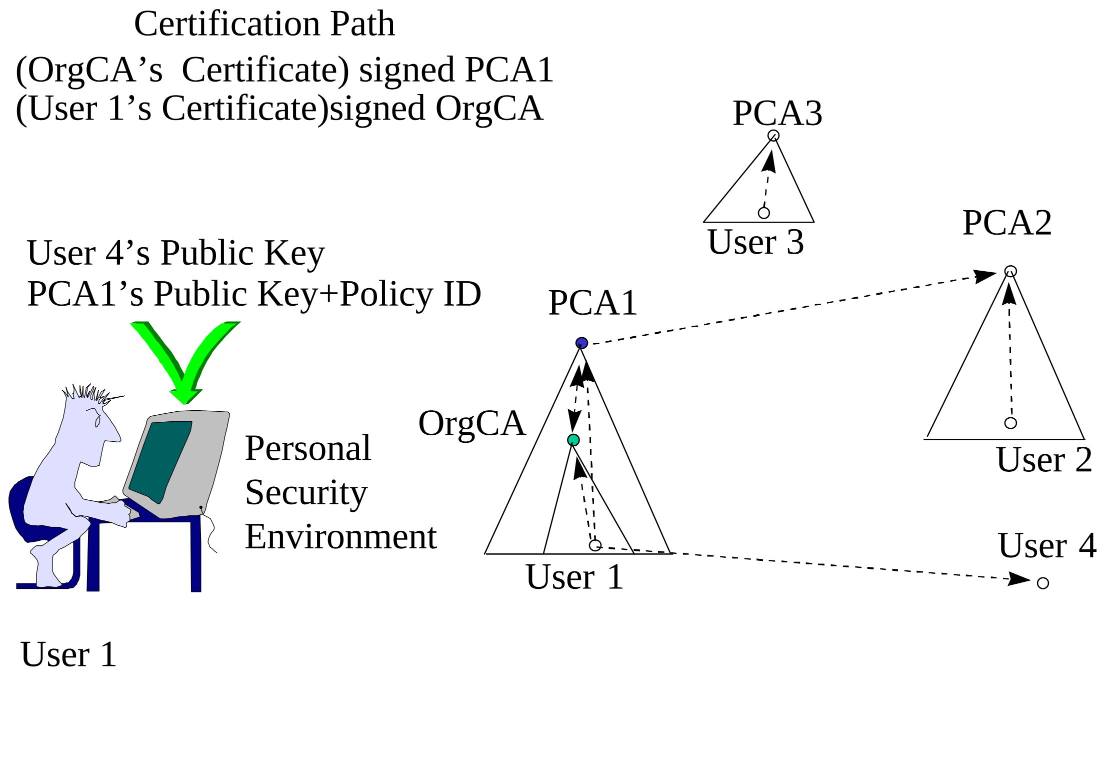

Figure 4 ICE-TEL Trust Model - an Example

Summary of ICE-TEL X.509 V3 Certificate Contents Used to Transfer Trust

The ICE-TEL trust model requires the following V3 Certificate extensions, in order to enforce the trust that a user has placed in his PSE, to be transitively conferred onto the public key of a remote user, through the certificate validating software in his workstation. If an extension is flagged as critical, then the certificate validation software must be able to process the extension and understand it, otherwise the certified user cannot be trusted.

i) *Basic Constraints Field*

the CA boolean must be set to true for certificates in which the subject is a CA. This allows the validation software to differentiate between CA and user certificates. This extension is set to critical.

ii) *Certificate Policies Field*

contains the policy id or ids of the security policy(ies) used by the CA when issuing the certificate. This allows the validation software to confirm that the same policy(ies) are still in effect by the issuing CA, as those entered by the user into his PSE. This extension will be set to critical or not as determined by the CA issuing the certificate (non-critical is allowed in order to cater for those security domains that will only ever support one policy; therefore it is not critical if the certificate validation software does not understand which policy was used).

iii) *Policy Mappings Field*

is only present in cross certificates, and maps the local policy into the policy of the remote domain. This allows the issuing CA to stipulate which remote policy is the one to be trusted, and the validation software can use this when validating certification paths. This field is set to non-critical. Certificate validating software that does not understand policy mappings will reject certificates after this one, as the policy id is no longer the same as the trusted one. To circumnavigate this, these implementations will require the users' PSEs to hold the policies of the remote CAs that are trusted (i.e. the user is not able to delegate cross certification to his CA administrator).

iv) *Policies Constraints Field*

If this extension is present, it is always set to critical.

Require Explicit Policy

if present is set to zero. This allows the CA administrator to ensure that future certificates in the certification path must always contain a policy id.

Inhibit Policy Mapping

if present is set to zero. This ensures that only the CA at the user's trusted point can cross certify other CAs, and further cross certification by remote trusted points is forbidden i.e. delegation of trust to remote administrators is forbidden. Note that if the security administrator of the trusted point CA wishes to relax this requirement, and allow his trust in a remote CA to be transitively transferred to another CA as determined by the remote CA administrator, then the integer can be set to one. A value greater than one will progressively increase the distance of transitive trust, whilst removing the inhibit flag altogether allows unlimited transfer of trust. Note however that the validation software in the user's workstation is ultimately responsible for enforcing the depth of delegation according to the *delegation* variable set by the user on the CA key/policy id stored in his PSE.

v) *Key Usage Field*

is set as appropriate to limit the use to which the public key may be put, as defined by the security policy of the trusted point. This field is set to critical, so that validating software that does not understand the use of the public key will not trust it.

vi) *CRL Distribution Points*

is set to point to the places that the CRL for this certificate may be found. The user's validation software must retrieve the CRLs according to the frequency as directed in his PSE. This field is set to non-critical, since it is assumed that the certificate validating software can also have other ways of knowing how to retrieve the CRLs, apart from those given in the certificate.

vii) *Name Constraints*

is set to limit (the names of) the users that can be trusted within a remote security domain. This field is critical, so that if the validating software does not know how to interpret the field it must fail to trust all remote certified users.

Implementation to date

Initial installation of the ICE-TEL PKI was completed in September 1996. A hierarchy of CAs has been established with the ICE-TEL Policy CA at the root. The root CA is being managed by GMD, the German National Research Organization, and leader of the ICE-TEL consortium. The public keys of 13 country level CAs (see table 1) have been certified by the ICE-TEL CA, and the validation software at each location is capable of processing the certificates generated by any of the CAs. The initial infrastructure uses X.509 V1 certificates, since these were all that were available at the start of the project. Four of the project partners: GMD, ISODE, COST and SSE, are currently implementing the X.509 V3 certificate extensions listed above, and this software will be distributed to the project partners and incorporated into the infrastructure during1997.

Country	Organization Name	Type of Organization
Austria	IAIK, Technical University of Gratz	University
Denmark	Danish Computing Centre for Research and Education (UNI-C)	Government institution under the Danish Ministry of Education
Estonia	Institute of Cybernetics (IOC)	State Research Institute
Germany	University of Hamburg, on behalf of Deutsches Forschungsnetz (DFN), the German Research Network	University
Greece	Intrasoft SA	Commercial Software House
Ireland	Software & Systems Engineering Ltd (SSE)	Commercial Software House
Italy	Politecnico di Torino	University
Norway	UNINETT A/S	Norwegian Academic and Research Network Operator
Portugal	Fundacao para a Computacao Cientifica Nacional (FCCN)	Portugese R&D Network Operator
Slovenia	Jozef Stefan Institute (IJS)	Research Institute
Spain	Fundacio Catalana per la Recerca (FCR)	Not-for-profit organisation promoting scientific research
Sweden	Computer Security Technologies CST AB (COST)	Commercial Software House
United Kingdom	University College London (UCL)	University

Table 1 The Organizations Operating Country CAs

Figure 1 shows the geographical locations of the ICE-TEL PKI.

Certification services will be offered to both end users and organization based CAs. It is the intention that all of the CAs will be able to offer a certification service able to receive certificate requests in either PEM or PKCS-10 format, or by WWW forms. The CA will generate a request id, return this to the requester, and in addition tell them the supplementary information that is needed in order to positively identify them. This supplementary information will be sent by an external channel. Once this has been done, the CA will generate a certificate and return it to the requester, in a format that can be used by either Netscape or Microsoft Explorer (e.g. as a MIME type). This service is already operational in Norway using tools based on the SECUDE package.

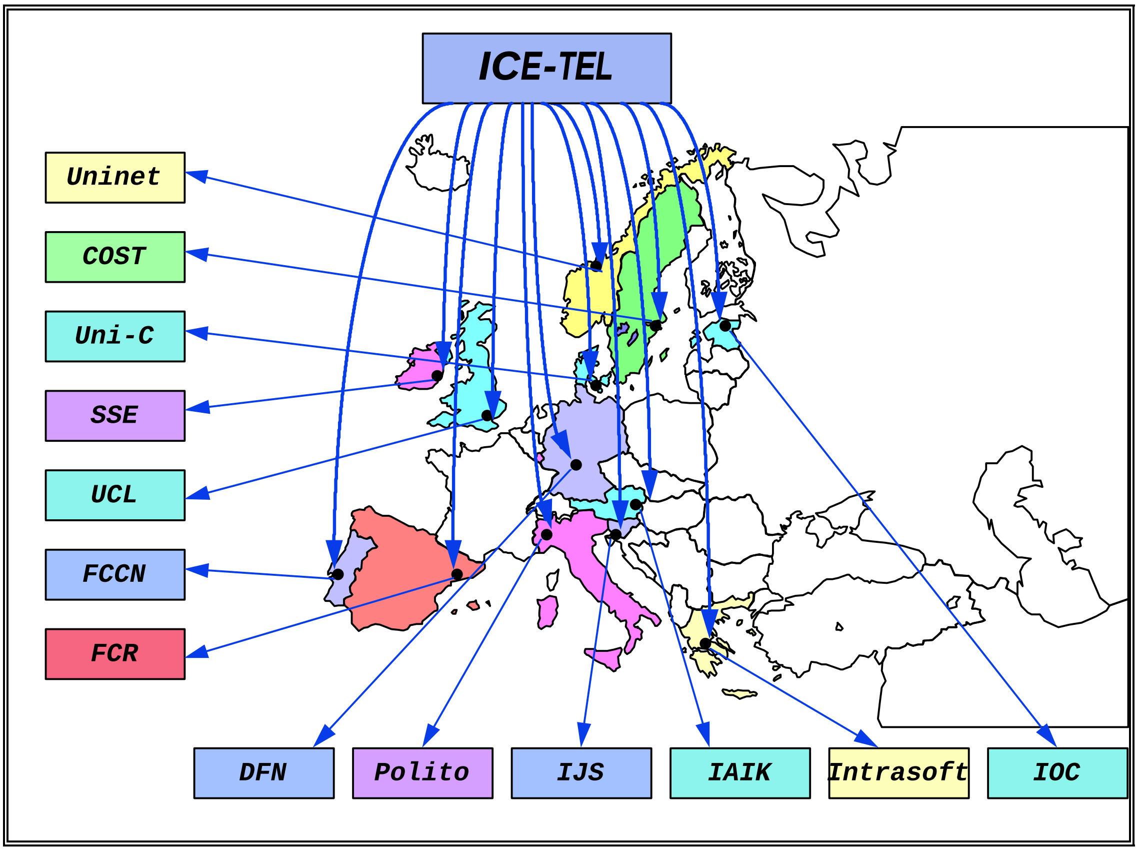

Figure 5 Initial Structure of the ICE-TEL Certification Infrastructure

Conclusion

The ICE-TEL trust model has taken the best elements of the existing PGP and PEM (and to a lesser extent SPKI) trust models, whilst avoiding their worst features. The resulting trust model exhibits useful properties which include:

- the user is at the centre of the trust model, and determines who he will trust to do what. This information is stored in the user's personal security environment (PSE)
- the model supports trust in both the public key to name binding (authentication) and public key to policy binding (authorisation)
- the model supports both organic growth, without requiring certification authorities, plus scaleable growth through having certification authorities
- the model supports delegation of trust from the user to the security administrator of his CA, and transfer of trust from a local user and/or security administrator to a remote security administrator
- the trust model is enforced through standard fields defined in X.509 V3 certificates, and by validation software in the user's end system that checks upon the contents of these fields, and is driven by the information in the user's PSE.

References

[1] "Information Technology - Open Systems Interconnection - The Directory - Authentication Framework" ISO-IEC STANDARD 9594:1990-8 | CCITT X.509 (Blue Book Series), 1989

[2] VeriSign "Frequently Asked Questions: Answers About Today's Digital IDs", available from the Web http://www.verisign.com
[3] Details about the ICE-TEL project can be found at http://www.darmstadt.gmd.de/ice-tel/
[4] National Computer Security Center “Trusted Computer System Evaluation Criteria”, Dec 1985
[5.1] Zimmermann, P. “The Official PGP User’s Guide”, MIT Press, ISBN 0-262-74017-6, 1995
[5.2] PGP frequently asked questions at: http://web.mit.edu/afs/net/mit/jis/www/pgpfaq.html
[5.3] Zimmermann, P. “PGP User’s Guide, Vol 1: Essential Topics” available free with PGP software
[5.4] Stallings, W. "Protect Your Privacy: the PGP User's Guide". Englewood Cliffs, NJ: Prentice-Hall, 1995 ISBN 0-13-185596-4
[6.1] Linn, J. "Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures", RFC 1421, February 1993
[6.2] Kent, S. "Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management", RFC 1422, February 1993
[6.3] Balenson, D. "Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes and Identifiers", RFC 1423, February 1993
[6.4] Kaliski, B. "Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services", RFC 1424, February 1993
[7] Veling, S.H.J. “Project Proposal - SURF-ACE PGP-Support”, Tilburg University Computer Center, Jan 1995
[8] “Information Technology - Open Systems Interconnection - The Directory - Authentication Framework” ISO-IEC STANDARD 9594:1993-8 | ITU-T X.509, 1993
[9.1] “Information Technology - Open Systems Interconnection - The Directory.” ITU-T REC. X.500-X.525 | ISO/IEC STANDARD 9594:1993 Parts 1 to 9
[9.2] Chadwick, D.W. “Understanding X.500 (The Directory)” (1st Reprint), International Thomson Publishing, July 1996, ISBN 185 0322 813
[10] “Information Technology - Open Systems Interconnection - The Directory - Authentication Framework - AMENDMENT 1: Certificate Extensions”, Draft Amendment, May 1996
[11] Housley, R., Ford, W., & Solo D. “Internet Public Key Infrastructure. Part 1: X.509 Certificate and CRL Profile”, Internet Draft <draft-ietf-pkix-ipki- part1-03.txt.Z>, Dec 1996
[12] “Secure Electronic Transactions (SET), Book Two: Technical Specifications”, 7 Aug 1996, available from http://www.mastercard.com/set/set.htm
[13] For PGP software distribution, and location of key servers, see http://web.mit.edu/network/pgp.html (US and Canada) or http://www.ifi.uio.no/pgp/ (Rest of World)
[14] Mendez, S & Huitema, C. “A New Approach to the X.509 Framework: Allowing a Global Authentication Infrastructure without a Global Trust Model”, p172-189, IEEE, 1995
[15] L. M. Kohnfelder, "Toward a Practical Public-Key Cryptosystem", B.Sc. Thesis, MIT Department of Electrical Engineering, 1978.
[16] The current SPKI Draft Specification can be found at http://www.clark.net/pub/cme/spki.txt

[17] “ICE-TEL Deliverable D3 - Architecture and General Specifications of the Public Key Infrastructure”, September 1996.
[18] ISO/IEC 8824:1988 | CCITT X.208 Specification of Abstract Syntax Notation One (ASN.1)
[19] Berge, N. “UNINETT PCA Policy Statements”, RFC 1875, December 1995
[20] “Basic Security Policy to be employed in the ICE-TEL Project”, Version 1, 18 July 1996, available from http://www.darmstadt.gmd.de/ice-tel/euroca/policy.html
[21] Farrell, S., Adams, C., & Ford, W. “Internet Public Key Infrastructure. Par III: Certificate Management Protocols”, Internet Draft <draft-ietf-pkix-ipki3cmp-01.txt.Z>, Dec 1996

Acknowledgments

The authors would like to thank DGX111 of the European Commission for funding the ICE-TEL project.