Overview of Certification Systems: X.509, PKIX, CA, PGP & SKIP

Do you understand digital certificates? Do you know what they warrant?

by Ed Gerck, Ph.D.[*]

To be secure, Internet communications need both encryption and digital certification – for example, for e-commerce and Internet voting. This work deals with digital certification issues and reviews the three most common digital certification methods in use today, which are based on X.509/PKIX Certificates and Certification Authorities (CAs), PGP and SKIP.

The certification methods are respectively classified as directory, referral and collaborative based. For two parties in a dialogue, the three methods are further classified as extrinsic because they depend on references which are outside the scope of the dialogue. A series of conceptual, legal and implementation flaws – including lack of suitability of purpose – is catalogued for each case, emphasizing X.509 and CAs. This analysis can be applied as safety guidelines for those who need to rely on digital certificates. Governmental initiatives introducing Internet regulations on certification, such as by TTP, are also discussed with their pros and cons regarding security and privacy. Throughout, the paper stresses the basic paradox of security versus privacy when dealing with extrinsic certification systems – which is very important in voting systems.

This paper has benefitted from the feedback of the Internet community and its online versions at the MCG received over 250,000 Internet visitors from more than 80,000 unique Internet sites in 1997/2000. The paper was also presented by invitation at the Black Hat Conference, Las Vegas '99. This 2000 revision is a major update, especially on the X.509 and PKIX sections. This version was published in THE BELL, ISSN 1530-048X, and is also available at the MCG web site and at www.thebell.net/papers/

Introduction

The Internet is an open system, where the identity [1] of the communicating partners is not easy to define. The communication path is non-physical, non-deterministic, and may include any number of eavesdropping and active interference possibilities. Furthermore, in a dialogue between two parties on the Internet, no party can control both sides of the communication channel or even the route between them – neither when sending nor when receiving. Notwithstanding the presence of a recognized sender's name in a message, the receiver must consider that message as if it were from an unknown sender (i.e., was anonymously sent) until proven otherwise. When sending a message, the sender must always assume until proven otherwise that the message was not received by the intended recipient (i.e., was anonymously received), even if there is a return confirmation message. In addition, both sender and receiver must always assume that anyone (i.e., anonymously) may have read or changed that message when it was routed from sender to receiver. Thus, Internet communication is much like anonymous postcards that are anonymously routed and answered . However, to be useful, these postcards, open for anyone to read, write, change, or discard, must carry messages between specific endpoints in a secure and private way.

The solution is to use encryption (to assure privacy) and digital certification (to assure that communication is happening between the desired endpoints and that it is tamperproof) [MOV97]. This paper deals extensively with certification, the ensuing paradox of privacy versus security, as well as the closely related matters of encryption and Internet protocols.

The question is whether we should be willing to sacrifice privacy in order to have security [Ger00]. In e-commerce the answer has been a resounding "Yes." And this approach has been quite successful. E-commerce Internet security is based on breaking privacy, from digital certificates as discussed here, to credit-card transactions, to registering a dot-com domain name [Ger00].

In elections, however, we need a "privacy wall" between the voter and the ballot – if I get the vote I cannot know who the voter is; if I get the voter I cannot know what the vote is. Some of the security technology provided by digital certificates, as discussed here and used in e-commerce, cannot preserve the anonymity of the vote [Ger00], a right protected by law and considered essential to election integrity, and democracy.

The problems that may be caused by false certification or no certification mechanisms can range from a "man-in-the-middle" attack (when an active attacker is able to alternatively pose as either party to the other party, so that while the parties believe that they are communicating securely the attacker reads all the traffic) in order to gain knowledge over controlled data, to a completely open situation to gain access to data and resources (when an attacker poses as a valid user). Such problems do not disappear with encryption or even with secure protocols such as SSL [Fel97]. If the user is led to connect to a site which appears to be the desired one, as in a spoofing attack [Fel97], the user may end up with a secure connection to a fraudster – which is worse because of the ensuing false sense of security.

This paper reviews the three most common certification methods in use today, which are based on X.509/PKIX Certificates and Certification Authorities, PGP and, SKIP.

These methods are studied from a systemic point of view. The main motivations for this paper are to: (i) Conduct a comparative review of the three methods, (ii) Unify a set of references to the most important issues in certification and encryption, as they are related to Internet needs and recent governmental policies, (iii) Provide a basis for the evaluation of other certification solutions available or to be developed, (iv) Identify room for improvements in the current security level of certification that could be dealt with by other methods, (v) Provide users with safety guidelines to be used when resolving certification issues, and (vi) Assess the impact on Internet transaction security due to the security control policy needs of governments currently actively promoting such policy solutions. The original version of this paper is online [Ger97a].

It is important to note that IETF's PKIX [PKIX] is a direct derivation of X.509. The reader will find essentially the same conceptual features, solutions and problems in PKIX as in X.509.

1. Certification Methods

Public-key cryptography may give the impression that security can be simply achieved. It seems that one only has to distribute the public-key at will, with no need to control it, and anyone can receive secure messages. The same procedure being applied to each side, sender and receiver, both could immediately engage in secure communication.

However, who is at the other side? Is the key really from the sender? Is the key still valid? Questions soon appear and it becomes clear that public-key cryptography has indeed solved the problem of public-key security but not the problems of public-key acquisition, recognition, revocation, distribution, re-distribution, validation and, most importantly, key-binding to an identifier and/or key-attribution to a real-world entity. In addition, communications can be verified neither for origin authentication (i.e., to verify whether the message was sent by the declared sender) nor for data-integrity authentication (i.e., to verify whether the message was changed after it was sent). Communications can be private but not secure.

Of course, a private communication with a fraudster is not secure just because it is private. Clearly, without binding the key to an identifier such as a person's common name, the key is just a byte string and can be yours as well as anyone else's. But common names or identifiers are oftentimes not enough. For example, where legal capacities must be defined, one needs to have some assurances that the key can be attributed to one well-defined, real-world entity such as a person or company.

Certificates provide a strong binding between the public-key and some attribute (usually the entity's name and/or the entity's real-world identity). Certificates still entail all the previous questions, such as certificate acquisition, recognition, revocation, distribution, re-distribution, validation and, most importantly, what is the intended meaning of key-binding to an identifier and key-attribution to a real-world entity. And certificates insert new questions, namely the privacy concern for identifiers and real-world entities (e.g., an Internet voter).

However, certificates are very useful and their benefits may far outweigh the difficulties mentioned above. Certificates introduce tamperproof attributes which can be used as convenient references to help someone receiving a message decide whether that message, the key and possibly the sender's name are what they appear to be – ***without asking the sender***.

Absolute certification methods are logically impossible because a certificate cannot certify itself. Even though various methods have been proposed to deal with this situation, for the sake of liability and validity analysis we can recognize three main classes as the author pointed out for the first time in 1997:

- Directory methods: X.509 and CA [X509a,-b], PKIX [PKIX]

- Referral methods: PGP [PGP]

- Collaborative methods: SKIP [SKIP]

Each of the above class presents a different certification paradigm (i.e., deals with certification questions in a different way), as analyzed in the following sections. However, they have a common ground because for two parties in a dialogue they depend on references which are external to the dialogue between the parties. Hence, these certification methods are also called extrinsic. Further discussion on the general characteristics of extrinsic certification, as well as the existence proof of two other

certification modes called intrinsic and combined, is presented by the author in [Ger97b].

2. X.509 and CAs

The ITU-T Recommendation X.509 (which has been implemented as a de facto standard) [X509b], describes two levels of authentication: simple authentication, using a password as a verification of claimed identity; and strong authentication, involving credentials formed by using cryptographic techniques. It is this second level that interests us here. It defines a framework for the provision of authentication services under a central control paradigm represented by a "Directory".

The "Directory" is implemented by a Certification Authority (CA), which issues certificates to subscribers (CA clients) in order for such certificates to be verifiable by users (the public in general). There are thus three main entities which can be outwardly recognized in X.509 certification procedures:

CA: a general designation for any entity that controls the authentication services and the management of certificates. The CA is also called the issuer. A CA can be public (a bank that issues certificates to allow its clients to access their bank accounts), commercial (a service provider that sells certificates to other parties, such as Verisign), private (a company that issues certificates to allow its employees to perform job duties), or personal (you, me). CAs are in general independent, even in the same country. The legal and technical relationships between a CA and its subscribers and users are governed by a Certification Practice Statement (CPS) issued by the CA. The CPS is internally defined by each CA within broad limits and lies outside the scope of X.509, even though X.509 references several items to be defined in the CPS, as discussed in the next items.

Subscriber: an entity that supplies to the CA the information that is to be included in the entity's own certificate, signed by the CA. The subscriber is a commercial client to a CA. Usually, as defined in the CA's CPS, the information supplied by the subscriber is "endorsed" by the issuer, where "endorsed" means "copied as received". This corresponds to "endorsement without recourse". For example, in English law one can endorse "without recourse" (or, as it used to be expressed, "sans recours"), which passes on the benefit of a bill of exchange without adding any guarantee. In other words, the CA copies the subscriber's information to the certificate, but neither denotes nor confirms it – there is no warranty.

User: any entity which relies upon a certificate issued by a CA in order to obtain information on the subscriber. Also called the verifier. Users may use any CA or any number of CAs, depending on their location and ease of access. The user should be central to the decision process in all steps, since the user is the party who is relying on the information and is thus at risk.

Another entity is the Naming Authority (NA), which is usually not outwardly perceived but which is the actual entity that defines the naming scheme used by a CA. The CA can double as a NA, but they provide two different functions. Semantically, the CA certificate refers to a name; however, it does not denote it. The NA denotes it.

The authentication services provided by CAs are especially relevant in regard to three central questions:

What is a X.509 certificate?

Even though section 3.3.3 of X.509v3 defines a certificate as: *"user certificate; public key certificate; certificate: The public keys of a user, together with some other information, rendered unforgeable by encipherment with the private key of the certification authority which issued it."*, there are several open questions regarding the contents of certificates and their issuance conditions which need to be discussed, as well as the issue of certificate revocation.

What is the naming scheme used in X.509 such that a certificate can be associated with a user?

Section 11.2 of X.509v3 – "Management of certificates"– states that the certificate allows an association between a name called "unique distinguished name," or DN for the user, and the user's public-key*: "A certificate associates the public key and unique distinguished name of the user it describes."* Section 7 explains that such DNs are essential to the security design of X.509*: "Authentication relies on each user possessing a unique distinguished name."* But how are DNs assigned? Where are they unique? The DN is denoted by a NA and accepted by a CA as unique within the CA's domain, where the CA can double as a NA. It is interesting to note, however, that the same user can have different DNs in different CAs, or can have the same DN in different CAs even if the user is not the first to use it in any of the CAs.

What are the validation procedures for the certified data that is included in a certificate?

X.509 is moot on validation procedures for data included in a certificate, such as the user's name, with the exception of validation procedures for the user's public-key which are suggested (not mandated) in Section 10 of X.509v3. For example, regarding validation procedures for the user's identity, Section 11.2.a states that: *"a certification authority shall be satisfied of the identity of a user before creating a certificate for it"*, which means that identity validation procedures are to be satisfied in the CA's frame of reference by following the CA's own self-defined rules (the CPS), which can be entirely different for different CAs. Furthermore, commercial CA's CPSs generally accept indirect references when issuing certificates, such as using an ID as identity proof, which can be easily subject to fraud and lead to public risks.

Thus, X.509 focuses on defining a mechanism by which information can be made available in a secure way to a third-party – the certificate itself. However, X.509 (and PKIX) do not intend to address the level of effort which is needed to validate the information in a certificate, nor do they define a global meaning for that information outside the CA's own management acts.

The main purpose of a CA is to bind a public key to the name contained in the certificate and thus assure third parties that some measure of care was taken to ensure that the binding is valid for both name and key. However, the issue whether a user's DN actually corresponds to identity credentials that are linked to a person or simply to an e-mail address, and how such association was verified, is outside the scope of X.509 and depends on each CA's self-defined CPS and on each NA.

Regarding the all-important DN specification denoted by the NA and accepted by the CA, the X.509 DN scheme is based on ITU-T X.500 Recommendation [X500a], [X500b]. But X.500 is not completely defined and apparently never will be. There is no Internet workgroup, not even ITU-T as its proponent, that currently works on X.500 final naming definitions. This is due to several factors, such as the lack of a centralized world body that would be acceptable to all parties and needs, and most importantly, the perception that global indexes involve strong privacy concerns.

Thus, there has been ample room for many different readings of the proposed X.509 Recommendation, since different implementations had to ad hoc define how DNs would be used in X.509. The X.509 Recommendation also depends on many other factors, including ISO, ANSI, ITU, and IETF standards, amendments, meeting notes, draft standards, committee drafts, working drafts, and other work-in-progress documents. In addition, the convoluted language used in some of these specifications makes their use difficult by itself, as pointed out by Peter Gutmann [Gut98].

A characteristic of X.509 is that almost all issues that involve semantics or trust are delegated to a CA's CPS, the Certification Practice Statement, which is declared out of scope in relationship to X.509. The CA's CPS is the governing law that the CA presents to potential clients and represents a top-down framework. While some consider the CPS mechanism to be a good way to introduce flexibility in X.509 because each CA can have their own rules for different needs, such a mechanism can be considered as X.509's ***black-hole*** and cannot be directly harmonized for different CAs.

Thus, while this ***black-hole*** mechanism affords a "solution" to the undefined semantic and trust features in X.509 (as they are declared out of scope and delegated to the CPS), this *laissez faire* attitude leaves ample room for strong differences between CAs and for a biased "take-it-or-leave it" attitude regarding what a CA subscriber can expect.

These problems have caused independent interpretations of X.509 in actual implementations, e.g., as shown in products from Netscape, Microsoft, RSA and others, and by CAs.

For example, lack of CPS harmonization does not allow X.509 to directly scale to a planetary Internet, when different CAs would need to allow for cross-certification (i.e., when subscribers of different CAs are users to one another). Even though cross-certification could work in a parochial Internet where everyone knows what to expect and shares a common law and trust system, it is doubtful that it could be successfully applied between competing businesses or different states in a country, much less between different countries, since there is no common world law. There are also subjective and intersubjective aspects of certification and trust [Ger97c]which are needed, but which cannot find a unified global expression that would be required for X.509 cross-certification.

Besides, X.509 certificates are not human readable and the user cannot easily see what is being accepted [GerBoh]. In fact, the user has to take for granted that what is being used is correct even when a browser presents a readable conversion. Even experts disagree on basic X.509 issues, as explained above, and there is usually ample room for doubt about what exactly a X.509 certificate is, why it is acceptable or why it is not acceptable. In other words, ***X.509 certificates have a twilight zone exactly on the most important issue with certification: what has been certified***.

Thus, since the objective of digital certificates is to allow credentials to be denied, the main risk is to accept what should be denied. The reverse side, to deny what should be accepted, is usually just a nuisance but can also impact security in a denial-of-service attack (e.g., preventing an otherwise legitimate transaction to occur), in a diversion attack (e.g., where the user is led to abandon a secure channel and use an insecure channel instead, where the attack is then possible), or in any other form of attack that depends on legitimate credentials being denied.

Another point is that X.509 certificates need a "Directory" service provided by a CA, that deals with the users and supplies copies of the certificates, even though the certificate is used off-line with the CA. This means that a CA is needed for two basic reasons: (i) to issue "standard" X.509 certificates that can be interpreted unambiguously and, (ii) to make it possible to have their validity verifiable by a user. This introduces limitations and regulation issues associated with other types of centralized control [Ger00a].

However, who verifies the CA's validity? The CAs themselves are usually "self-certified" or depend on a CA that is "self-certified".

The CA paradigm in X.509 and PKIX is thus, essentially, to rely on an authentication chain that ends in a CA that eventually certifies itself – which is a logical impossibility. Therefore, the validity problem is shifted from a local

perspective to a global perspective, with the whole chain of trust depending on one final link. At the end point, ignorance (and the possibility of fraud) is leveraged to a high degree, in that one weak link may compromise a whole chain of certificates.

What are the causes for weak links? In addition to the conceptual points above, there are several more, including interpretation and implementation. For example, consider the protocol SSL (Secure Socket Layer, e.g., version 3.0 [SSLa], [SSLb], with an IETF equivalent specification being developed as TLS) [2]. If an X.509 Certificate is acceptable by a vendor's SSL product and behaves in a certain way, it does not mean that it will necessarily do the same with products from another vendor.

In general, this paper recommends that certificate users and subscribers carefully use their due diligence when relying on certificates and CAs. The following items provide a series of security considerations to be used as guidelines to enhance security and privacy in Internet communications when X.509/PKIX certificates are used. Certificates are not magically infused with trustworthiness just because they are digitally signed. The signature, the contents, the validity or all three may be wrong, the result of fraud, the result of a bug, or revoked. The fact that this is not clear to the average user is a further cause of concern since reckless behavior usually magnifies problems. To help remedy this situation, this paper also presents several hints to the user. When the hints may be especially noteworthy, they are underlined or emphasized.

2.1. Initial decision of trust removed from the user.

This is a design flaw that pervades the use of certificates on several levels. Most of the servers that use CA certificates force the user to accept certain CA's signatures which are "hardwired" into commercial client software, such as browsers, by commercial agreements between client software vendors and CAs. Some particular CA signatures may allow for enhanced browser functionality, such as 128-bit strong encryption, which the user cannot control either. Thus, the decision of trust is entirely removed from the user right from the beginning, which contradicts the premise [3] that the user should be central to the decision process in all steps since the user is the party that is relying on the information and is thus at risk. The problem here is that a decision to trust someone, or a source of a communication, or a name on a certificate, or a certificate (in short, information) must be based on factors **outside** the information itself: "Trust is that which is essential to a communications channel but which cannot be transferred from a source to a destination using that channel." [Ger99c]

One simple solution to the majority of these problems would be to ask users to digitally sign ***all*** CA certificates ***before*** they are usable by the browser. Thus, the browser could still have all the commercially-defined CA certificates that the vendor trusts, but this would ***not*** translate into an initial decision of trust in the user's name. It is quite puzzling to see that this simple measure is not currently implemented by any browser vendor. However, it is the same idea one finds in PGP certificates (see section 3) and it is certainly possible in technical terms even in the X.509/PKIX format.

2.2. Automatic trust.

Forcing a trust decision upon the user, as explained in 2.1. above, is compounded with what the author has called ***"automatic trust"*** [Ger99a]. Here, the browser or email agent actually makes the trust decisions for the user, storing otherwise untrusted certificates in the user's computer and using those certificates on a par with some certificates that the user himself may have stored. This happens, for example, with Netscape Communicator. In the specific case cited in [Ger99a], the certificate becomes automatically trusted both for its authenticity as well as for its capability of supporting encrypted messages sent to that purported email holder using "his" purported public-key within the certificate's purported validity.

The difficulty here is that the browser is making secondary decisions, even beyond the initial trust that was forced upon the user (see 2.1.). Trust has become fully "automatic." Of course, even if the user would be willing to accept (see 2.1.) that the browser already includes some CA certificates that the browser unilaterally trusts (including a generic "Intranet CA" [Ger99a]), the user should not be forced to also accept that mere inclusion of these certificates will result in derivative work being performed by the browser. But this is the case and such derivative works include, for example, files being stored in the user's computer without limit (e.g., if the user receives 100 signed messages a day, there will be 100 certificates stored a day), without warning and without the user's consent. The situation is more troubling when these files are actually certificates that then become automatically trusted by transitivity from that same CA certificate that the user never accepted or even knew existed (e.g., "Intranet CA" [Ger99a]).

But it gets worse. In the case of Netscape Communicator 4.51 and possibly other browsers, if that new certificate was signed by a CA that not even the browser trusts, then it is still automatically stored without warning or consent in the user's computer. True, if the user tries to send an encrypted message using that certificate then the browser balks, so not much harm is done except for further polluting the user file of now "automatically trusted but-otherwise-useless certificates."

One of the causes of these problems is clear: trying to make security easier for users. However, security is very much in opposition to functionality in digital certificate design. Let us recall that the objective of digital certificates is to allow credentials to be denied. In general, security involves a basic "hass le price" and the ***design question should thus be to make it worth the hassle rather than to decrease the hassle at all costs including that of security itself***. Thus, "routing around" the manual steps of accepting CA certificates (2.1.,

above) and also "routing around" the manual steps of accepting new user certificates (as commented in this item), all in the good name of functionality, will decrease the level of security afforded by digital certificates. So why have it? Since "automatic trust" does not exist, then the desired "automatic security" must also be as fictional.

The simple solution would be to review all such steps of "automatic trust" that are infused in clients such as Netscape Communicator and other applications, providing users with clear dialogue boxes and clear choices whenever ***any*** decision must be made that is not supported by directly affirmed user trust. Trust is, essentially [Ger97c], "that which I rely upon for my decisions" and so must be used by any of my agents, including my browser, if they are to be trusted by me. Otherwise we would be trusting that which runs amok.

2.3. "Hotel California " certificate registry.

This item is a consequence of the flaw described in 2.2., with the additional deviant behavior that the user may delete the untrusted message (for example, once he sees who sent it or what the message contains), but browsers (e.g., Netscape Communicator) will keep the corresponding certificate stored in the user's computer without warning to the user. Furthermore, the certificate is trusted by the browser just as it was before the message was deleted. In "Hotel California" (a well-known song by the Eagles), we have a similar behavior – "we are programmed to receive and you can check out any time you like, but you can never leave."

The simple solution to this is for the vendors to change this behavior in the browser codes, since this has nothing to do any X.509/PKIX requirement.

2.4. Virgin birth certificates.

The problem is that the certificate stored according to the behavior described in item 2.3. becomes an orphan when the user deletes the message, so that the user will not be able later on to even connect the certificate with its origin. A "virgin birth" certificate is created. Furthermore, since the user (probably) did not request that signed message, then the user (very probably) will not verify it either. Nonetheless, the certificate was stealthly and automatically stored, made trusted and made active for its declared lifetime (in general, one year or more). If the user decides to verify the message's signature (out of no actual need) while the message is not yet deleted, this is what Netscape Communicator says: "This Certificate has automatically been added to your list of People's Certificates to make it possible for you to send secure mail to this person." Thus, the browser further blindfolds the user and also makes believe that a "person" sent the email. The virgin birth is complete.

The simple solution to this is for the vendors to change this behavior in the browser codes, since this has nothing to do any X.509/PKIX requirement.

2.4. Need to trust untrusted CAs.

The client receives a list of server-trusted DNs (Distinguished Names) in the "certificate request" message sent by the server in SSL. If there are no CAs that the user directly trusts in that list, the user may need to accept a CA unknown (i.e., untrusted)to the user, or a CA that is trusted by a CA that the user trusts. The first case is an open security risk while the second case contradicts the principle that trust is not transitive in general. In other words, if you trust your brother it does not mean that you must equally trust the same friends that your brother trusts.

This problem has two simple solutions. First, in X.509/PKIX users should be able to decide how many levels of trust they are willing to accept between themselves and the CA they trust. Second, in SSL if the client application would send to the server a list of CAs that are trusted by the user then the server would not be groping in the dark as to what CAs the user trusts and would thus be able to provide a better list of CAs to the client – which behavior, by the way, is already what happens from the server side. In SSL and as commented above, the server does send to the client the list of CAs that the server is willing to accept in the "certificate request" message for server authentication. Why should the client application not have a chance to define to the server what the user is willing to accept?

2.5. A rose by any other name would smell just the same, but not DNs.

We may recall the famous phrase by William Shakespeare, which reminds us that names are simply pointers to objects. The same certainly happens with a DN (Distinguished Name) in a certificate. However, according to X.509 and PKIX, determination of whether the same name in two certificates relates to the same entity is outside the scope of the standard. A rose may not smell the same even if it is called rose. As noted before, the same user can have different DNs in different CAs, or can have the same DN in different CAs even if the user is not the first one to use it in any of the CAs. Hence, different DNs for different CAs do not necessarily mean different users and vice-versa. Furthermore, a DN may not contain the user's real-world name or location.

2.6. Certificates expire in a domino effect.

The life of a server or client certificate cannot extend beyond the life of the certificate of the signing CA, which creates a domino effect that collapses all certificates signed by a CA when the root CA certificate expires. The principle here is that after the expiration of the CA's certificate, one should assume that the corresponding root-key may have been cracked (which is why it has a finite life in the first place) or discarded without enough care (e.g., by wiping disk areas, surely destroying all copies, etc.). Anything signed by that key thereafter should be suspected of being a forgery. In other words, if

someone presents to you a certificate today that was signed by a key linked to a certificate that expired last month, you should assume it might be a forgery. However, if you knew that the certificate was signed during the lifetime of the signing CA's certificate, you could assume that it was authentic (based on the principle that the private key had not yet been cracked). The problem is that currently there is no way to ascertain from the certificate exactly when it was signed in relation to other lifetimes.

2.7. Certificates can have very short lifetimes.

The lifetime of a certificate also depends on various factors, as pointed out by Ian Simpson [Sim97]. A simplified mathematical analysis and simulation shows that optimal certificate lifetimes may be as short as a few weeks, rather than a year or more as is the case with some current commercial CA infrastructure offerings. However, such short lifetimes mean a large overhead in cost, time and effort. Another factor, according to a mathematical model proposed and discussed by the author [Ger99b], is that the lifetime of a certificate is inversely proportional to the sum of the inverse lifetimes of each of its attributes. This means that adding attributes always decreases the lifetime of a certificate, so that adding information in order to increase the lifetime of a certificate will actually achieve the opposite result.

2.8. Multiple certificates are needed for the same key.

You must have multiple certificates for the same key in order to cope with the use of different non-communicating CAs and with different expiration dates. For example, if you have a server that is certified by a CA, your own certificate must be substituted before it expires, while the older one is still valid and is registered somewhere in someone's access files.

2.9. Protection is an inverse function of worth.

The Certification Authority's public key might be the target of an extensive decryption attack. For this reason, CAs should use very long keys and change keys regularly. Unfortunately, top-level CAs are exceptions to this rule: it may not be practical for them to change keys frequently because their keys may be written into software (such as the browser you are using now) used by a large number of verifiers. Thus, the very CAs that may be the most probable targets are the ones that offer the smallest protection level. There is also a serious question of how one would distribute an updated top level CA certificate when the expired certificate is "hardwired" in the software. Unless there is a second trusted CA who can sign the distribution, the new certificate cannot be certified.

2.10. Certificates can be compromised by a chain of events.

CA's may suffer internal problems (software bugs, internal frauds, collusion, bad bookkeeping, bad auditing practices before issuing a certificate, etc.) that may compromise any number of certificates, including the CA's own root certificate. Any of these problems may compromise a large number of issued certificates without the user being aware of them.

2.11. Certificates do not include direct verification data.

Certificates do not usually include information about how a relying-party could verify the data that is certified. Since certificates are provided to users without warranties and without suitability of purpose (see item 2.18), providing the users with means to validate data in certificates would be very useful. Certificates also do not provide commercial information such as that which will control the flow of documents and monies, for example the correct phone number to call for a bank account to receive deposits. They also do not allow for temporary changes of personnel in charge to cope with, for example, vacation schedules. To alleviate this problem, Netscape has proposed a new type of certificate, to be used together with X.509/PKIX Certificates, called Attribute Certificates. This is being studied by the PKIX WG. Attribute Certificates are signed objects that assert additional properties about a particular identity certificate. An attribute certificate has no associated key pair and consequently cannot be used to establish identity.

Informally, one can think of attribute certificates as a mechanism for extending the attributes of an identity certificate without requiring that the identity certificate be reissued. Formally, they are a "patch" type of solution that may introduce a series of inconsistencies (e.g., revocation lists for either type of certificate, cross dependencies, etc.).

As modeled by the author [Ger99b], certificates with more attributes will tend to have a shorter lifetime than certificates with a lesser number of attributes. This means that it is indeed advantageous to place information in certificates so that expiration of one attribute will not invalidate an entire certificate with otherwise valid attributes. An analogy can be made with a box of dynamite sticks, where upon the explosion of one stick the entire box explodes. A box with two sticks will tend to explode in half the time as a box with one stick, and so on. Placing too many attributes in one certificate is generally not a good idea because certificate lifetime is inversely proportional to the number of attributes [Ger99b].

2.12. Certificate revocation lists are a will to revoke, not an actual revocation.

Revocation lists, or CRLs, are needed to notify users that an otherwise valid certificate is not valid. This, which was first thought to be a positive aspect of relying on CAs, as compared to PGP for instance, presents several unsolvable problems [4]. For example, there may be a

considerable delay (no warranties can be found in the CAs contracts on upper limits for such delays) between the actual need to revoke a certificate and the reflection of this need in a certificate chain with different CAs. Further, the major X.509 security application today, SSL, does not check revocation lists, so they are near to useless. Also, the user is not able to check server certificates (and certificates in the CA chains) against revocation lists. An exemplary case against CRLs is that they are a will to revoke but not an actual revocation. Few recognize that CRLs are a solution to a non-existent problem, while the real problem is left utterly unsolved. The non-existent problem solved by CRLs is how to communicate that a certificate is no longer valid, because if a certificate were really no longer valid (as it should be) then no one would need to find a CRL to know about it.

2.13. Certificates are legally warranted for methods, not results.

In business practice, it is often reasonable for one party to rely upon the representations of another without verifying them. This provides for added assurances to the relying-party. Would this be the case for the relationship between a CA and a user, when a certificate issued by a CA to a subscriber is presented to a user? At first glance, it would seem so because in this case *"...the user is the relying party and views the other party as an expert."* [5], the "other party" being the CA that issued the subscriber's certificate. Certainly, in the user's view, the CAs are experts on certificates. Also, to add further weight to this presumption, this is a case where *"... the other party's statements seem reasonable on their face value but are especially hard to check."* [5], (e.g., the statements on the CA's subscriber certificate, especially with the name "certificate" together with "authority"). It is also hard to check the validity of the CA's signature in the subscriber's certificate. Unfortunately, these well-known legal principles and guidelines are not valid for certificates, as exemplified by the proposed U.S. Uniform Commercial Code, because *"for data processing and design or consultation work, the basic focus for gauging liability centers on the process, rather than on a guaranty of correct result."* [6] Indeed, data processing services are usually warranted for methods, not results, so that the usual business interpretation of relying-party jurisprudence given in footnotes [5] and [6] cannot be considered.

2.14. Certificate users are not legal relying-parties to the CA.

The user of a certificate is not privy to the contract between the CA and the CA subscriber, hence there is no relying-party relationship formed between a user and a CA – even indirectly, because CAs deny open warranties. Therefore, even though the user can reasonably rely upon the certificate presented by the subscriber to be free from tampering (i.e., because of the cryptographic assurances) after it was issued by the CA, the user cannot rely upon the CA's declarations in it and must somehow judge the reliability of the data contained in the Certificate, which is further complicated by the question of the applicable governing law. However, the ABA (American Bar Association) Guidelines seem to suggest that such a decision must indeed involve a final "relying party" attitude because the user must rely upon the last CA as "reasonably reliable" before considering the server certificate valid: *"... a person relying on the certificate must verify its digital signature by referring, in turn, to another certificate, and so on along a chain of certificates until reaching a valid certificate digitally signed by a primary certification authority, whose digital signature is reasonably reliable..."*. This is confusing because it means that even though the expected behavior is not to be expected, the user has no choice but to rely upon the CA's representations in the certificate.

2.15. Client and S/MIME certificates are issued using insecure on-line protocol.

It has been discussed in the MCG and elsewhere that mainly MS Internet Explorer and secondarily Netscape Communicator do not follow a secure protocol for clients' certificates issued for SSL or S/MIME. Thus, both browsers introduce the possibility of implicit key-escrow, weak-keys and covert channels without the user being able to verify it (though less pronounced with the Netscape browser). This means that a system could be imposed where a CA would demand the user's encryption keys before signing the certificate for the public key. However, this would not happen if the private key generation procedure was done entirely off-line, without an Internet connection, which is simply not needed for the procedure. This simple act would completely allay such privacy concerns, and yet it is not done in MS Internet Explorer and Netscape Communicator.

2.16. "Certificate Authorities" and "Certificates" are misnomers.

The denominations "Certificate Authorities" and "Certificates" are certainly not to be understood by their etymology. A "Certificate", which has the same root as the word "certain", is used in day-to-day words such as Gold Certificates, Certificates of Deposit, etc., with a very clear and precise meaning beyond any doubt. The situation is quite different with the "certificates" we are dealing with here. In this case, the "certificate" represents a wrong contextual clue that leads to a type of implicit spoofing situation, in which unwary users, or even the non-technical users who are the majority, are led to believe that the words "authority" or "certificate" carry the same weight as their dictionary entries and day-to-day experience would imply. The user does not expect them to be misnomers. To quote from Ed Felten et al [Fel97]: *"The names of objects can convey context. People often deduce what is in a file by its name. Is manual.doc the text of a user manual? (It might be another kind of document, or it might not be a document at all.) URLs are another example. Is MICR0S0FT.COM the address of a large software company? (For a while that address pointed to someone else*

entirely. By the way, the round symbols in MICR0S0FT here are the number zero, not the letter O.) Was dole96.org Bob Dole's 1996 presidential campaign? (It was not; it pointed to a parody site.)".

2.17. "Certificate Authorities" and "Certificates" are actually certified by the users.

The X.509 system gives the impression of a self-regulated and safe system, whereas it is clear from the points above that the user is the one and only true authority who will eventually certify the so-called "Certificate Authorities" and "Certificates". However, as given above, there are many reasons that may jeopardize a "certificate", create a weak link or give the wrong contextual clues for on-the-spot decision making. The uncertainty reaches a point of almost uselessness [McCur], where CAs usually explicitly state in the certification contracts that the CA is exempt of all or almost all responsibility regarding the "certificate," its accuracy and its data. For example:

> VERISIGN DISCLAIMS ANY WARRANTIES WHATSOEVER WITH RESPECT TO THE SERVICES PROVIDED BY VERISIGN HEREUNDER, INCLUDING WITHOUT LIMITATION ANY AND ALL IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. VERISIGN MAKES NO REPRESENTATION OR WARRANTY TO ANY PERSON THAT ANY CA OR USER TO WHICH IT HAS ISSUED A CERTIFICATE IN THE VERISIGN SECURE SERVER HIERARCHY IS IN FACT THE PERSON OR ORGANIZATION IT CLAIMS TO BE IN THE INFORMATION SUPPLIED TO VERISIGN, OR THAT ANY CA OR USER IS IN FACT THE PERSON OR ORGANIZATION LISTED IN THE CERTIFICATE. VERISIGN MAKES NO ASSURANCES OF THE ACCURACY, AUTHENTICITY, INTEGRITY, OR RELIABILITY OF INFORMATION CONTAINED IN CERTIFICATES OR IN OR OTHER CERTIFICATE STATUS MECHANISMS COMPILED, PUBLISHED, OR DISSEMINATED BY VERISIGN, OR OF THE RESULTS OF CRYPTOGRAPHIC METHODS IMPLEMENTED IN CONNECTION WITH SUCH CERTIFICATES.

To make matters more complicated in terms of understanding the above in a uniform manner, such CA disclaimers are part of the CA's Certification Practice Statement (CPS) which is outside the scope of X.509 (the ***"black-hole"*** mechanism in X.509, as already mentioned). Thus, while the CPS is the governing law that the CA presents to potential clients, the CA may vary the CPS at will irrespective of any standards.

The CA's disclaimer is generally not visible in the certificate itself, as seen in a browser for example. Some browser and X.509 implementations, however, do allow the CPS to be referenced in the certificate –even though the CPS itself is not included to be read by a user. Referencing the CPS legally situates it in the "four corners of the document." Unfortunately, a legal reference to a (usually) 116-page CPS [VS00], which also references other documents and so on, is of no help to the average Internet user.

> Marx Brothers fans will possibly recall the scene in "A Day at the Races" in which Groucho, intending to put his money on Sun-up, is induced instead to buy a coded tip from Chico and is able to establish the identity of the horse only, at some cost in terms of time and money, by successive purchases from Chico of the code book, the master code book, the breeders' guide and various other works of reference, by the end of which the race is over, Sun-up having won.

What is needed is a simple, readable, unabridged and no-roundabout way to say what X.509/PKIX digital certificates issued by a commercial CA are good for. After reading several 116-page CPSs, a savvy user and a lawyer would perhaps summarize an average CPS as given in the paper *"X.509 Certificates: a readable unabridged inside view"* [GerBoh], which provides a graphic illustration of many items discussed here.

Current CPSs can be seen as a legally-enforced way to reduce deliverables to almost zero, which is an accepted legal practice to effectively reduce liability to zero. However, the point is not so much that CAs deliver a product with zero warranty (which could be disputed in court even in countries with common-law systems) but that CAs are delivering a service with almost zero content (which any court would accept as involving almost zero liability). This is clearly exemplified in the second and third sentences of the above disclaimer ("...MAKES NO REPRESENTATION ..." and "...MAKES NO ASSURANCES ...").

Actually, the content of a CA's services in relationship to the subject's data is not zero because there are two items that X.509 mandates that a CA must deliver in a certificate without content disclaimers (but which may still be limited in scope by the CPS):

> (i) that the subject's public-key has a working private-key counterpart elsewhere (with no warranties that the public/private key pair is not artifically weakened, that it is actually in the possession of the named subject and that no one else has obtained a copy of it), and
> (ii) that the subject's DN is unique to that CA (with no warranties that such DN contains the actual subject's name, location or that the subject even exists or has a correctly spelled name – as in "Internet Serices" [RSA]).

There are also other relevant items in the certificate, with no relationship to the data supplied by the subject but which

are necessary for the proper use of the certificate as a secure transport for information in the X.509 model, such as its serial number, date of issuance, validity, the CA's signature, etc. These items, even though entirely supplied by the CA, usually also carry limited CPS warranties (e.g., as in the case of fraud, computer viruses, etc.) and may jeopardize the secure use of certificates without the user being even aware of it.

2.18. Certificates cannot provide the assurances users normally expect.

It is important to acknowledge the reasoning behind certificate disclaimers, as typically given in item 2.17, and understand what they mean. The typical disclaimer of a CA does not say that the CA has no warranty on its services or that it does not take any liability for them. Such a disclaimer only says that the CA has no warranties and accepts no liability for services that it does not recognize it provides. The lesson here is clear: buyer beware. Users of certificates must recognize the limitations of what a certificate can really provide. Thus, the solution to providing assurances may well be to educate the users rather that expect CAs to do what is perhaps technically unfeasible in terms of X.509 or PKIX.

Thus, in the author's opinion, such CPSs and disclaimers are not at all at odds with X.509 or legislation. Indeed, we see that CAs have adopted them without adverse reactions. Maybe they truly represent the best that one commercially and technically could wish for in X.509's and CPS's terms.

Thus, for any generic CA one might expect a similar reasoning. Indeed, if the only thing that a CA does (regarding X.509) is to challenge the subscriber's private-key in order to bind the corresponding public-key with the subscriber's DN, and if it signs the certificate with a CPS that says that any other data are being copied as received (and have thus no warranty), then the CA has no responsibility for the contents of the certificate, except for the positive acknowledgment that the public-key did have a counterpart when it was linked to that DN (where the CPS could further provide exceptions for frauds, virus, MITM (man-in-the-middle) attacks, etc.).

Why are such content limitations and disclaimers necessary for certificates? First, a certificate is not like a car that has a limited liability in space and time. (After all, a car is a localized entity that can contain a limited number of people and only one driver). A certificate can be endlessly multiplied and simultaneously presented in a planet-wide area. Certificates are used without limit in a chain of events, which can include other fully unrelated certificates and people [7]. With the growing attitude of legally seeking large compensation for one's lack of foresight, the liability pyramid created by a lesser disclaimer could easily extend to the CA's client's clients and so on.

2.19. Insurance has limitations.

Insurance protection may help here, but there are several issues that must be touched upon. The use of insurance always signals lack of knowledge. However, insurance clearly cannot replace knowledge. There is no insurance needed for a sure event and there is no insurance possible for a sure risk. Furthermore, if a user (i.e., a CA subscriber) is going to for pay insurance to cover his liabilities and the CA's liabilities (which is what it amounts to), then responsibility has gone full-circle and is now only in the user's hands, both to get adequate coverage and to pay for it. Meanwhile, the CA has zero risk and cashes in as the middleman between the user and the insurance companies. However, that does not solve the risk problem for the user either, because one cannot make the whole world sign up for one huge insurance policy. So the user and the CA may be protected by the insurance policy that the user bought using their names as beneficiaries, but that does not protect a third-party (i.e., the rest of the world). Finally, since CA auditing does not help here, then insurance does not have a reliable risk estimator either, even for the CA subscriber.

Another argument against the use of insurance as the "final solution" in certificate reliance is that one must limit the number of users that are insured per certificate (and use an estimated total loss) and/or the amount of total loss per certificate (and use an estimated average number of users). This is not satisfactory, however, because the number of users has been always difficult to estimate in Internet e-commerce. Some products draw just a few thousand clients in spite of all projections while others are able to attract hundreds of thousands. Currently, there are 400,000,000 Internet users and any scalable Internet security solution must take these numbers into account, as well as user mobility. Internet users can go anywhere, at any time.

2.20. Law may help but it cannot create engineering security.

Regarding recent legislation efforts, such as in Utah (US), Illinois (US) and federal legislation S.761 recently signed by President Bill Clinton (The Millennium Digital Commerce Act), it is clear from the above discussion that demanding broader warrants by law can be self-defeating because CAs may then be forced to reduce the deliverables even closer to zero, instead of coming out and providing for more warranties. There simply is a limit to what X.509 and the CA paradigm can offer regarding legal certificate reliance and, most importantly and often confused with the former, legal certificate content reliance. In short, law cannot push the technical envelope of X.509.

2.21. Auditing is not very effective with CAs.

When confronted with risk situations, a normal business solution is to rely on auditing. However, auditing of a CA's certificates is also a difficult, if not impossible, task. This is due to X.509, which allows CA's practices and policies to be

built upon islands of self-regulation exactly on the most important issues of trust and trust management. As publicly declared by Phillip Hallam-Baker, a Verisign consultant, not only are the CPSs indeed different and self-made by each CA, but they are not designed to be audited either: *"There is not as yet a defined standard for CA practices against which a company may be audited. In effect, each company states their own practices in their Certificate Practices Statement (CPS). The CPS is not a document designed for auditing use however. It describes a 'specification', it does not describe details which may be checked by a third party in a systematic manner."* Currently, this situation is changing by the use of ad hoc auditing procedures, as one would audit any business.

2.22. Legal reliance is local to the issuer, not to the user.

A X.509 certificate may well be called a "bag of bytes" whose meaning and validity strongly depend on the CA. Thus, in legal reliance terms, one may trust the confirmation procedures of the CA during certificate reliance, but one cannot rely upon them for other than their value as a representation of the CA's authentication management act expressed in the CA's own terms and rules. Therefore, a X.509 certificate is neither necessarily meaningful nor valid in a user's reference frame or for the user's purposes.

2.23. Trust is earned, not given away.

When one watches for some time the different mailing lists that collect doubts and questions on certification systems from users, or when one reads the majority of newspaper or magazine articles on the subject, one cannot help but perceive a prevailing feeling in the user community that a certificate is magically infused with trustworthiness. This feeling, however, implies a deterministic and absolute view of certification. For example, as one user wrote: *"Please provide me with a list of all trusted CAs so that I can enter those certificates into my browser."* Few understand that trust must be evaluated relative to the user, who is the party at risk. Thus, the very names Trusted Third Party or trusted CA raise several questions:

- trusted in relationship to whom?
- trusted by whom?
- trusted for what?
- trusted for how long?

etc.

How are these questions answered? By each user (i.e., verifier or relying party–who is at risk) in his own domain, references and terms. This means that certificates are essentially statements from a CA [8], not facts, and that meaning and trust in a certificate (like beauty) is in the eyes of the beholder, i.e., depends on each user.

2.24. Non-repudiation is ill-defined.

Non-repudiation is defined in X.509 as follows: *"This service provides proof of the integrity and origin of data both in an unforgeable relationship which can be verified by any third party at any time."* However, this is not the only definition of non-repudiation in X.509. There are conflicting versions as we will discuss shortly. Furthermore, this definition also differs from the one found in Menezes in the Handbook of Applied Cryptography [MOV97]): *"Non-repudiation: a service that prevents the denial of a previous act."* The PKIX WG also pursued other definitions, such as equating non-repudiation with long lifetimes in "non-ephemeral authentication."

However, as the author discussed in the PKIX WG [Ger99c], we may agree that defining non-repudiation in X.509 as *"This service provides proof of the integrity and origin of data both in an unforgeable relationship which can be verified by any third party at any time"* is just a round-about way of defining strong authentication. In terms of mathematical logic, this author sees authentication as that which affirms the truth of an act (which can be verified by any third party at any time) whereas non-repudiation denies the falsity of an act, thus preventing the denial of the act. Both are equal if and only if the proposition is boolean (i.e., either true or false), which is almost never the case in security (where propositions are not atomic and are multivalued). So, confusing non-repudiation with "authentication" or "strong authentication" or "non-ephemeral authentication" and thus actually vacating the concept of non-repudiation, will not go very far because it would leave non-repudiation undefined, even though a definition is still needed.

It is useful to list what X.509 actually says. These are all the occurrences that one can find when looking for the key text "repudiation" in X.509:

> *"It is a matter for the security policy and responsibility of the CA to keep old certificates for a period of time if a non repudiation of data service is provided."*
>
> *"If a non-repudiation of data service is dependent on keys provided by the CA, the service should ensure that all relevant keys of the CA (revoked or expired) and the timestamped revocation lists are archived and certified by a current authority."*
>
> *"nonRepudiation: for verifying digital signatures used in providing a nonrepudiation service which protects against the signing entity falsely denying some action (excluding certificate or CRL signing, as in f) or g) below);"*
>
> *"The date in this extension is not, by itself, sufficient for nonrepudiation purposes. For example, this date may be a date advised by the private key holder, and it is possible for such a person to fraudulently claim that a key was compromised some time in the past, in order to repudiate a validly-generated signature."*
>
> *"Repudiation – The denial by a user of having participated in part or all of a communication."*
>
> *"Non-repudiation -- This service provides proof of the*

integrity and origin of data -- both in an unforgeable relationship – which can be verified by any third party at any time."

"The data integrity mechanism supports the data integrity service. It also partially supports the non-repudiation service (that service also needs the digital signature mechanism for its requirements to be fully met)."

"The digital signature mechanism supports the data integrity service and also supports the non-repudiation service."

The above shows that X.509 itself is in contradiction when defining non-repudiation. However, X.509 consistently associates proof of integrity with the digital signature mechanism. The situation is made clearer in Table B.1 "Threats and protection", where we can read that the non-repudiation service protects against the threats of replay and repudiation, while "data integrity" and "end authentication" are different services.

Thus, some non-repudiation definitions in X.509 MUST be wrong, as well as the consequences derived from them. Comparing these with the technical definition of Menezes (cited above), we see that the same definition used by Menezes is found in X.509 if we choose the X.509 definition for repudiation quoted above:

"Repudiation –The denial by a user of having participated in part or all of a communication."

and construct the logical complement of 'non" as:

Non-Repudiation –Preventing the denial by a user of having participated in part or all of a communication.

Alternatively, if one says that "provable data origin authentication with integrity" is an appropriate name for non-repudiation in X.509 (instead of the above choice), the author takes exception and cites X.509 itself. In X.509, authentication that is neither provable nor provides for integrity is ***not*** authentication in the sense of strong authentication as used in X.509 digital signatures (Section 10.1), where two-way authentication provides assurances for "the integrity and originality of the authentication token sent in the reply."

Thus, as the author discussed elsewhere and as shown above, X.509 textually agrees with the definition by Menezes et. al. and with the usual sense people would attach to something that is the complement of repudiation, the complement of denial.

In terms of PKIX, the issue seems clearer when compared to X.509. The PKIX WG now seems to agree that non-repudiation is a service that prevents a user falsely denying having participated in part or all of a communication. The word "falsely" is still ambiguous, however, because a user may truly deny having participated in an act and yet the non-repudiation framework that the user has accepted may make such denial invalid though truthful. For example, a user who denies having signed a check that was stolen will probably not prevail at a bank if the bank did not receive from the user a notice that the check was stolen prior to the check being paid and if the bank could not distinguish the signature from a signature that the user did not make.

Some other general topics were considered by the author in the PKIX WG, especially in the message 'Simple answers"[Ger99d].

To conclude, we are led to observe that X.509/PKIX "certificates" are not at all "certain" as their etymology would imply. Certificates are mere indications with no assurances or warranties. Certificates also increase system cost and complexity. Certificates are however useful as convenient references to help someone receiving a message decide whether that message, the key and possibly the sender's name are what they appear to be – *without asking the sender*. To rely on digital certificates, either as a subscriber or as a user, the recommendation is that one must, in a case by case analysis, provide or ask for additional assurance mechanisms – which can be technical, legal (additional contracts) and/or policy-based. Subscribers and users must also weigh the privacy versus security tradeoffs and limitations in the X.509/PKIX model.

2. PGP

PGP has two parts: certification and encryption. The discussion below is centered exclusively on the certification aspects of PGP.

Comparing PGP with X.509 can be very instructive. X.509 is frequently cited as predicating a top-down trust structure (see the CPS discussion above) that is just dictatorially imposed upon the verifier, while PGP follows a grass-roots approach and is thus more Internet-like. However, both PGP and X.509 define their central role to be played by the verifier regarding certificate acceptance, while certificate metrics is defined in both cases without any influence from the verifier (thus, "dictatorially" for both). Furthermore, both are key-transport protocols, and they depend on two types of external references: keys (quantitative) and trust (qualitative). Another similarity is that the web-of-trust in PGP has a parallel in the X.509 CPS, where the issuer sets the rules and defines semantic acceptance conditions before certificate signature.

The first main difference is possibly syntactic, in the sense that PGP allows certificates to be stacked as signatures upon signatures, whereas in X.509 the certificates are linked one to another as in a one-way linked-list (though X.509 could also include PGP syntax). A second main difference is semantic, in which PGP allows an association between keys

and real-world persons by the web-of-trust but not transitive trust, whereas X.509 binds keys to names and accepts transitive trust, even though a proper CPS could allow the same in X.509 as a function of the CA's policies. (Think again of the analogy between the web of trust rules in PGP and the CA's CPS rules in X.509).

PGP is based on an "introducer-model" which depends on the integrity of a chain of authenticators, the users themselves. The users and their keys are referred from one user to the other, as in a friendship circle, forming an authentication ring. The term ring is not used here to denote a closed structure but a mathematical set which can at present be loosely modeled as a list or "web of trust". At the end, you may not know very well the last person who entered the ring, but you hope that someone else in the ring does. Or you may have different rings with "contact points" which guarantee the referrals. However, the reader should note that no user can know for sure if everyone in his authentication ring has a valid entry. The term "chain" can be used to denote such connected rings, which can also, of course, have multiple connections.

The reader should notice further that the maintenance of this chain-changing, adding or deleting data is done by the authenticators themselves in a happenstance pattern. There is no guarantee if and when the chain is up-to-date. Everyone familiar with the classical problem (or need) of file-locking in a multi-user environment will recognize that there is no "file-locking" mechanism here. So, while PGP enforces a model of "hard-trust" with "trust is intransitive" to setup entries in the web of trust, it uses "soft-trust" to upkeep entries, without discussing its validity/gauge or allowing for time factors such as lack of synch.

There are several other problems and benefits of PGP which this paper will not address. This is not intended to be a dismissive treatment of PGP, but rather a focus on commercial applications. It is important, however, to note that one of the benefits of PGP is that it can interoperate with a CA fully-trusted by all parties in a domain (such as an internal CA in a company) that is willing to guarantee certificates as a trusted introducer. Better tools would certainly be necessary for central administration of PGP trust parameters in a corporate system, but the flexibility of PGP makes it a good example of a quasi-decentralized system.

The concept of "central administration of PGP," which to some might sound even sacrilegious, is a way to guarantee accountability, coherence, dependability and, above all, correct authentication. Of course, within a circle of close friends this is not important.

Because there is no entity responsible if (or when) something goes wrong – not even the user – the use of PGP in a commercial situation is difficult and may not adequately protect the business interests involved, which usually need to be guaranteed in well-defined contracts with loss responsibilities and fines. Furthermore, PGP does not scale well in size (because of the aforementioned asynchronous maintenance difficulties of the web of trust) or time (because of the same maintenance problems reflected in the so-called certificate revocation certificates, a CRL for PGP certificates). But again, within a circle of close friends this is not important.

3. SKIP

SKIP implements a linked chain of two-sided node authenticators. Each node authenticator derives its information from a type of "Directory service." Without dismissing SKIP as a valid and interesting protocol, it is important to note that non-repudiation and other security features that depend on certificates will necessarily also depend on data from the application layer. But since every step of the SKIP authentication process happens at the protocol level (not at the message level), the SKIP protocol needs to be complemented by a second authentication protocol in a higher layer.

Note that SKIP is transparent to the user, for better or for worse. This means that SKIP lacks user-tunable controls, such as the rejection, revocation, visualization or choice of certificates. Here, also, a type of "Directory service" must be used by the node authenticators to obtain information. This is a type of "central administration of SKIP" which is needed to guarantee accountability, coherence, dependability and, above all, correct authentication. The difficulties of implementing such an administration system worldwide are compounded by the fact that a given packet route will not have a unique path, not even if the same packet route is being used for a series of requests, such as those produced when a Web site is visited. The situation is totally different from, for example, a PGP session, where the authentication is done at fixed endpoints and the routing path is not important.

Therefore, in SKIP the user has no practical way to control the process, cannot decide which node authenticator is reliable, cannot exclude nodes which have been infected by the enemy, cannot choose to choose certificates, etc.

In addition, SKIP cannot be used with network address translation, a common technique nowadays to conserve IP numbers and to hide the IP numbers of a network to another (usually an intranet to the Internet) as done in firewalls, where network interfaces inside a network are renumbered when seen from the outside. The same problem happens with IPSEC, an IETF point-to-point security protocol.

The use of SKIP in a commercial situation is thus difficult because the control decisions are totally removed from the user who is the one at risk. Furthermore, the system liabilities are ill-defined, responsibilities for fines and losses are hard to recover, and SKIP will not work with network address translation (IP translation).

3. Certification, Risks and Privacy Rights

As explained in the introduction to this paper, in the Internet encryption is not a luxury, but a necessity. And encryption without certification is an open door to spoofing and other kinds of attack. However, in each of the three methods analyzed above, the basic certification questions remain: ***Who is on the other side? Is the certificate valid? What is certified?***

To try to cope with this situation, which can have national and international impact, governments have proposed several initiatives. Because (i) certificates depend on some form of encryption (e.g., X.509), and (ii) encryption does not make sense without certification, the two issues, certification and encryption, are inherently present in all proposed initiatives. Before discussing the other aspects of these initiatives, which range from anti-terrorism to politics and beyond, it is important to review the two most important technical issues, as they have been discussed in this work.

First, the issuing of a certificate by a CA can be seen as a public service and thus must be on a par with other public services which must be, and indeed have been, regulated by the government to avoid abuse and misuse. For example, we could use the biscotti paradigm:

> If you want to make and eat your own biscotti, fine. If your neighboor wants to eat some of the homemade biscotti you made for your own consumption, this is also usually fine (unless you fear a lawsuit in case of food poisoning). If you want to buy it from someone else for your own private consumption, it is your decision. But if you want to sell it to the general public or to a store, then you must have a seal of approval and must comply with a set of rules.

The same principle applies to a public service that sells services to provide signature keys, public-keys or passwords. They can be subjected to impersonation, falsification, blackmail, etc., all with great potential harm. In other words, because the social order may be disrupted by this service, it must be regulated by the government. The alternative would be for it to be in effect regulated by criminals, which no one would support.

Second, one must take into account the lawful possibility and need to track communications, under court order, to prevent theft, terrorism and all types of crimes, including spying and national security threats. False certificates or too strong encryption can thwart these lawful prerogatives.
These two technical reasons have provided governments both with a reason, as well as with an excuse, to step in and issue or propose regulations for the Internet on a national, as well as on an international level, regarding the issues of certificates and encryption. Much controversy has been going on in the Internet in this area [EPIC], both pro and con. It is not a purpose of this paper to add to this controversy or to take issue on these subjects, but rather to present and discuss factual data that is pertinent to the technical discussion of the certification question. Besides, this paper focuses on the question of certificates, leaving the issue of cryptography for other efforts.

According to the TTP [TTP97] certification initiative led by the US [NISTa] and being tested as a potential government policy in the UK, as well as its derived Public-Key Infrastructure (PKI) [NISTb] proposal, the certificates issued by CAs and the CAs themselves would be vouched for by a complex chain of certificates that would all depend on some government appointed agency, a type of "seal of approval", which also provides for mandatory key escrow. Other initiatives, which can work together with TTP-certification, are the so-called key escrow or key recovery schemes [NISTa], the Clipper chip [Gret], GAKware [Shos95], the so-called International Cryptography Framework [HPICF], weak cryptography such as using single-DES, or even propositions to cripple cryptography [NISTa]. All these methods, besides the obvious advantages of introducing a centralized control method, provide a back door into each person's or company's private businesses by giving agencies the possibility of easy decryption of otherwise private messages. One could add that these methods make network systems insecure by design, whereas before they were insecure by accident.

At the very least, these methods may eventually represent the end of the Internet as we have it today: a free, essentially self-regulated and uncensored territory, with no visible national borders. While this is justified by some [Den96] as necessary *"in order to control crime and anarchy,"* [9] for others [Wise95], *"one should avoid the indiscriminate extension of government to the Internet"*.

Adding other political and commercial undertones, the U.S., the UK, Australia, Belgium, Canada, France, and The Netherlands have tried to impose, or did imposed at some time, restrictions on cryptography, authentication or CAs. However, such restrictions, which were usually mandated by military treaties and alliances such as NATO but have no influence on countries which do not have such commitments, are being gradually lifted. ***Countries are realizing that allowing for strong cryptography is in the best interest of their citizens and private business, especially in view of the widespread possibility of eavesdropping on satellite and wireless channels, and on fax, voice, and Internet communications.*** A comprehensive survey on cryptography restrictions is being conducted by B-J Koops [Koop98], with data on almost all countries.

Thus, while some countries may have considered restrictions on cryptography as an acceptable solution to information control, it is becoming clear that not everyone and not every corporation wants their private correspondence to be written on an open postcard. Legal requirements, such as client-lawyer secrecy, patent rights, voting laws and other

internationally protected rights such as diplomatic mail and commerce, need a different solution.

Against the expansion of centralized information control, the OECD (Organization for Economic Co-operation and Development) has issued its Cryptography Policy Guidelines [OECD] that state:

> THE FUNDAMENTAL RIGHTS OF INDIVIDUALS TO PRIVACY, INCLUDING SECRECY OF COMMUNICATIONS AND PROTECTION OF PERSONAL DATA, SHOULD BE RESPECTED IN NATIONAL CRYPTOGRAPHY POLICIES AND IN THE IMPLEMENTATION AND USE OF CRYPTOGRAPHIC METHODS. GOVERNMENTS SHOULD CO-OPERATE TO CO-ORDINATE CRYPTOGRAPHY POLICIES. AS PART OF THIS EFFORT, GOVERNMENTS SHOULD REMOVE, OR AVOID CREATING IN THE NAME OF CRYPTOGRAPHY POLICY, UNJUSTIFIED OBSTACLES TO TRADE.

Thus, while it is clear that all the models reviewed (e.g., X.509, PKIX, CA, PGP, SKIP) do not offer adequate certification per se and actually demand some type of control in order to avoid crime and anarchy, the consequences of a centralized control (e.g., TTPs, GAKware or key escrow) would most probably jeopardize the free use of standard Internet practices, such as PGP mail encryption, SSL-enabled connections, e-commerce, etc.

6. Conclusions

X.509 Certificates, PKIX, CAs, PGP and SKIP need some type of centralized certification control systems in order to be useful in commercial situations. There seems to be, therefore, a basic systemic conflict between this need and the Internet architecture, which is totally decentralized and very independent in actions and in form. While this seems to demonstrate the uselessness of centralized governmental controls, since we have no centralized world governance or law, how can we provide for the necessary controls in the Internet environment?

This question has also been addressed in other application areas, such as Internet domain names. The author does not believe that strengthening centralized control and making it a single handle of control is a solution, because such control then becomes a single point of failure. Strong centralized control becomes the one basket for all eggs, which everyone wants to possess. This argument is further discussed by the author in [Ger00a] in terms of domain name issues, but may be appropriately applied here in terms of CA control.

This paper recommends that users and subscribers carefully use due diligence when relying on certificates and CAs, and provides a series of security considerations as guidelines to enhance security and privacy in Internet communications. Certificates are not magically infused with trustworthiness just because they are digitally signed. The signature, the validity, the contents or all three may be wrong, the result of a fraud or revoked.

One further conclusion is that governments may need to better adapt their control policies to the times at hand. Everyone knows the story of the person who always used a hammer in his work and thus thought that all problems were nails. Modern problems need modern solutions.

It is one of this paper's assertions that technology may provide the answer just as it provided the problem, but the answer does not lie in an increased centralized control that would be impossible to attain. Rather, this paper proposes a paradigm change: "The Internet is at odds with centralized control. Thus Internet control must be decentralized in order to be effective."

The author has also applied this principle to other areas in the interest of pursuing it as a new control paradigm that could be applied to the Internet. One example is the Internet Domain Name System (DNS), currently experiencing difficulties because of competing control interests from trademark owners who want to assert global name rights based on local trademark rights, versus Internet users who want to defend global name routing [Ger00a].

Note that this argument of a new control paradigm does not imply the absence of some type of control or "checks and balances" situation. That would be incoherent with this paper and with logic. This paper indicates that a new control paradigm is needed, not that no control is needed. This need for a new control paradigm can perhaps already be felt in many areas, in grass-roots developments that collide with the traditional vision of centralized control. For examples, the reader is referred to the discussions in the Internet Paradigm exposition by Einar Stefferud [Stef98].

And, as Gordon Cook wrote in the April 2000 COOK Report on Internet, regarding the author's essays published in that Report [Ger00a], law enforcement cannot solve the problems of Internet security either. *"One of the problems facing the Internet, is that we have, sometimes with chewing gum and bailing wire, built it into something on which a very large proportion of our economy is riding. The prevailing opinion in the wake of the DDoS [Distributed Denial of Service] attacks is to call in law enforcement, build the security walls ever higher and hunker down with publicly reassuring words to the effect of –don't worry we are in charge here. A careful reading of the technical discussion on pages 2 through 16 of this issue will show that this position is founded on quicksand. A reading of the Gerck essays and interview will reinforce this conclusion."*

The Meta-Certificate Group (MCG) [MCG] is an international non-profit open group that includes participants from several countries. The MCG was founded in 1997 as a fresh exploration of applied cryptography to solve real-world Internet security issues, for both individuals, corporations and governments, initially around

digital certificate questions. Work began with a period of public discussions from 1997 to 1999, with several published papers and discussion lists that laid out the framework for meta-certificates. A meta-certificate is an object that operates within a layered certification protocol. During 2000, the MCG is compiling the developed material into a new site with a call to general public participation, focusing on meta-certificate applications in Internet certification, privacy and security. The MCG is also drafting a proposal for an open Internet standard describing an object-oriented layered certification protocol called "Meta-Certificate Protocol," designed to enhance security and flexibility while preserving privacy. The protocol will allow standalone operation as well as interoperation with current technologies such as X.509, PKIX, CAs, PGP, SKIP, etc. Meta-certificates also offer some very interesting possibilities to build mechanisms where two parties which are unknown to each other may use a third neutral environment in which to securely negotiate conditions of trusted operation.

The first version of this paper was published in 1997 as part of the fact finding and modeling work for such efforts, in particular to define how meta-certificates could interoperate with other certificates.

Acknowledgments

The author acknowledges helpful hints and discussions with members of the Meta-Certificate Group, L. Zorzella, L. Machado, Einar Stefferud, Peter Williams, Gordon Cook, Tony Bartoletti, and Nicholas Bohm MA (Cantab), Solicitor of the Supreme Court of Judicature in England and Wales, and also members of the e-carm list, ssl-talk list, ssl-users list, cert-talk list, SPKI (Simple PKI) list, S/MIME list, PKIX list, Usenet newsgroups such as talk.politics.crypto, comp.security.misc, comp.security.pgp.discuss, sci.crypt, and the Internet community. Eva Waskell's kind help in editing this version is also acknowledged. The names herein cited do not indicate endorsement of or responsibility for this work, which is the sole responsibility of the author and reflects his viewpoints, not the viewpoints of any corporation, company, agency or governments.

FOOTNOTES

[1] Identity: There are cases for which the identity of the communicating partners is not necessarily relevant. Also, anonymous speech is useful in many circumstances, even when privacy is required. The US case of "Deep Throat" disclosing information about the criminal activities of President Nixon is one example of an anonymous though identifiable source in a private and secure environment. The public release of RC4 algorithm information on the Usenet is an example of an anonymous unidentifiable source in a public and insecure environment. To assure anonymity is sometimes as difficult as assuring identification. This paper however deals with the commercial relevant cases of identification needs. See [Boh97].

[2] SSL: SSL is not a socket protocol as the name might indicate but allows for encryption and certification functionality in a TCP/IP environment. SSL is perhaps the widest used security protocol on the Internet today and implements X.509 certification as interpreted by SSL's proponent, Netscape. There are also other implementations of SSL, such as a free implementation called SSLeay [SSLy] which is export-free and user-friendly and was developed by Eric Young and Internet collaborators. SSLeay is widely used with the well-known Apache Web server. The first fully functional version of Apache with SSL support was implemented by Ben Laurie. A full-Java implementation of SSL3.0 called J/SSL is available from Baltimore Technologies.

[3] Subjective and Inter-Subjective: Subjective means that one needs to take a subjective or personal instance in order to evaluate an object. Inter-subjective means that this instance can yield different results for objects of the same class. For example, beauty and trust are subjective concepts ("beauty is in the eyes of the beholder" and "trust depends on the observer") because trust and beauty are abstract objects that cannot be differently instantiated, while a medical diagnosis for a patient is inter-subjective because the diagnosis itself is a particular instance from the class of all diagnosis possible for that patient at that time, each clearly dependent on the patient's relationship to the physician and different from the other. An inter-subjective concept is overly-variable in reference to a subjective concept because it also depends on the particular instance of the classes' object. So even though trust is subjective, trust in a CA certificate is inter-subjective because it cannot be harmonized or harmonizable for all CAs or even for all similar certificates issued by a particular CA.

[4] CRL control issues: Besides the CRL problems presented in the text, it is worth mentioning a few other cases. Requiring the user to check with a CA for a CRL before sending a message makes the use of multiple CAs much more difficult, unless the CAs can be convinced to work together. This presents an interesting problem for competing businesses. Constant checking with a single CA also makes traffic analysis much easier. Even if the attacker cannot intercept the message which is sent, if the attacker can monitor the central CA (with a single administrative order and a GAKware system to circumvent any encryption), everyone's communication patterns can be seen. Also, an attacker can fool a CA into revoking a key – a denial-of-service attack.

[5] There are several examples of this logical principle in which the user has to rely on an expert, as in Hartong v. Partake, Inc., 266 Cal. App. 2d 942, 966, 72 Cal. Rptr. 722, 737 (1968); Hefferan v. Freebairn, 34 Cal. 2d 715, 719-20, 214 P.2d 386, 388-90 (1950).

[6] This is a logical situation where the user can see that the information appears correct as given but cannot check it using reasonable time and effort, as in Mariani v. Schonfeld, 126 Cal. App. 2d 187, 189-90, 271 P.2d 940, 942 (1954).

[7] Spoofing chain: A spoofing chain is an operation that tries to obfuscate false data, by giving it a shroud of credibility based on secondary steps that may not be perceived as insecure by a third person. For example, to obtain a false ID a person might begin by obtaining a copy of a true birth certificate of a deceased person, faking mail addresses directed to that name, obtaining secondary IDs such as library cards and working up the ladder to reach a higher level ID and even a SSN. However, because of such frauds, some governments now stamp with "deceased" copies of birth

certificates of deceased persons and routinely cross check IDs on a local and national level. This increase in government control of personal IDs because the public order is at risk is a parallel situation to the current increase of government control of "Internet IDs."

[8] Presumed "certification": The issuing of a certificate that contains false data – certified or not – gives credibility to that data, which may be used for a second step in a spoofing chain. Thus, it is not acceptable for a certificate to include fields that carry no verification, without explicitly declaring so, as the case with e-mails in current certificates. It is a poor practice to provide a "security" feature that can not be verified or enforced. It gives a presumption of safety to the unwary user. "In California, for example, each drivers license features a photo, several holograms, and a metallic strip for fraud prevention. But this didn't stop employees of the state's Department of Motor Vehicles from issuing bogus licenses to anyone willing to fork over the right amount of cash. An estimated 250 DMV employees have issued over 25,000 genuine-looking, but fraudulent licenses in a two year period. Some were paid as much as $1,000 for such licenses. 'Ironically, as our documents become more tamper-proof, it's become more of a problem', DMV Director Sally Reed admitted to the San Jose Mercury News", as reported by Nathan J. Muller.

[9] Internet access blocking: Some countries block themselves from the Internet, where a sanitized intranet serves the country. The same happens in reverse, when entire countries or domains are or were blocked. In 1997, an entire ISP in Holland was blocked from the Internet in Germany because of some www pages published by one of the ISP's clients. On April 11th 1997, the XS4ALL website www.xs4all.nl was censored by the Deutsches Forschungsnetz, the German academic Internet provider. This was confirmed by Dr. Klaus-Eckart Maass, managing director of Deutsches Forschungsnetz in a press release. The pages in question, however, were mirrored in other sites elsewhere in the world and it was thought that it would be almost impossible to find and block all such sites. The same situation happened in Canada, in the Homolka case, with the result that it was indeed impossible to block the spread of information and access to it.

References

[Ave98] Juan Avellan "Digital Signatures Links", in http://www.qmw.ac.uk/~tl6345/

[Bak98] Phillip Hallam-Baker, public discussion, in http://www.mcg.org.br/cgi-bin/lwg-mcg/MCG-TALK/archives/mcg/date/article-379.html

[Boh97] N. Bohm, "Identity", In MCG Web Page, http://www.mcg.org.br/identity.txt, April1997

[Canada] Canada Post 20-year certificate, in http://www.mcg.org.br/2016_cert.gif

[Den96] Dorothy Denning, "THE FUTURE OF CRYPTOGRAPHY", in http://www.cypher.net/info/pub/itar/denning_0296_cryptoanarchy.article

[EPIC] "INTERNATIONAL CRYPTOGRAPHY POLICY", in http://www.epic.org/crypto/intl/

[Fel97] Edward Felten et al., "Web Spoofing: An Internet Con Game", in 20th National Information Systems Security Conference (Baltimore, Maryland), October,1997. Also, http://www.cs.princeton.edu/sip/pub/spoofing.html

[Ger97a] E. Gerck. "Overview of Certification Systems: X.509, CA, PGP and SKIP". In MCG Web Page, http://www.mcg.org.br/cert.htm, April, 1997.

[Ger97b] E. Gerck. "Certification:Extrinsic, Intrinsic and Combined". In MCG Web Page, http://www.mcg.org.br/cie.htm, July, 1997.

[Ger97c] E. Gerck, "Trust: Reliance on Qualified Information". Published in the Internet by the MCG, http://www.mcg.org.br/trustdef.htm , January, 1998.

[Ger99a] "Automatic Trust", in the discussion list cert-talk, http://www.mail-archive.com/cert-talk%40mail.structuredarts.com/msg00834.html

[Ger99b] "General formula", in the discussion list of WG PKIX, http://www.imc.org/ietf-pkix/mail-archive/msg00993.html

[Ger99c] "Non-Repudiation", in the discussion list of WG PKIX, http://www.imc.org/ietf-pkix/mail-archive/msg01784.html

[Ger99d] "Simple answers", in the discussion list of WG PKIX, http://www.imc.org/ietf-pkix/mail-archive/msg01748.html

[GerBoh] E. Gerck and N. Bohm, "X.509 Certificates: a readable unabridged inside view" in http://www.mcg.org.br/x509cert.htm

[Ger00] E. Gerck, "Would You Vote Naked?", in THE BELL Newsletter, ISSN 1530-048X, Vol.1 No.2, June 2000, p. 3, online at http://thebell.net/archives/thebell1.2.pdf

[Ger00a], E. Gerck, "Thinking" in Cook Report On Internet, ISSN1071-6327, Vol. IX No.1, April 2000, p. 23, online summary at http://www.cookreport.com/09.01.shtml

[Gret] Ed Grether, "Clipper Chip Information Resources", in http://www.northlink.com/~egrether/clipper.html

[Gut98] Peter Gutmann, "X.509 Style Guide", in http://www.cs.auckland.ac.nz/~pgut001/pubs/x509guide.txt

[HPICF] "Hewlett Packard's International Cryptography Framework", in http://www2.hp.com/pressrel/nov96/18nov96c.htm

[Illi] "ILLINOIS ELECTRONIC WRITING AND SIGNATURE ACT – Draft with comments", in http://www.magnet.state.ma.us/itd/legal/106192.htm

[Koop98] Bert-Jaap Koops, “Crypto Law Survey”, in http://cwis.kub.nl/~frw/people/koops/lawsurvy.htm

[McCur] Kevin McCurley, “DigiCrime is now known as root@localhost”, in http://www.digicrime.com/id.html

[MCG] MCG - The Meta-Certificate Group, is an international non-profit open group. The MCG is a fresh exploration of applied cryptography to solve real-world Internet security issues of today, for both individuals, corporations and governments, as represented by the current certificate questions. The MCG Home-Page is at http://www.mcg.org.br

[MOV97] A. Menezes et al., Handook of Applied Cryptography, CRC Press, New York, 1997.

[NISTa] “US Cryptography Policy”, in http://csrc.nist.gov/keyrecovery/policy.txt

[NISTb] “NIST PKI Program”, in http://csrc.nist.gov/pki/

[OEDC] “Cryptography Policy Guidelines”, in http://www.oecd.org/dsti/sti/it/secur/index.htm

[PGP] Website at http://www.pgp.com

[PKIX] IETF PKIX, in http://www.ietf.org/html.charters/pkix-charter.html

[RSA] “Internet Serices” [SIC] certificate, in http://www.mcg.org.br/certspell.gif

[Shos95] Adam Shostack, “Response to NAS Crypto Questions”, in http://www.homeport.org/~adam/NAS.html

[Simp97] Ian Simpson, “Modeling the Risks and Costs of Digitally Signed Certificates in Electronic Commerce”, in http://www.ini.cmu.edu/NETBILL/pubs/certlife/certlife.html

[SKIP] Website at http://skip.incog.com

[SSLa] http://home.netscape.com/eng/ssl3/ssl-toc.html

[SSLb] http://www.netscape.com/newsref/std/SSL.html

[SSLy] SSLeay, in http://www.psy.uq.oz.au/~ftp/Crypto/

[Stef98] Einar Stefferud, “What is the Internet Paradigm? A Virtual Seminar”, in http://www.mcg.org.br/paradigm.htm

[TTP97] “LICENSING OF TRUSTED THIRD PARTIES FOR THE PROVISION OF ENCRYPTION SERVICES”, in http://www.cl.cam.ac.uk/users/rja14/dti.html

[UCC] US Uniform Commercial Code, in http://www.law.upenn.edu/bll/ulc/ucc2/ucc2b296.htm. See also Hartong v. Partake, Inc., 266 Cal. App. 2d 942, 966, 72 Cal. Rptr. 722, 737 (1968); Hefferan v. Freebairn, 34 Cal. 2d 715, 719-20, 214 P.2d 386, 388-90 (1950), Mariani v. Schonfeld, 126 Cal. App. 2d 187, 189-90, 271 P.2d 940, 942 (1954).

[Utah] “Utah Digital Signature Program”, in http://www.commerce.state.ut.us/web/commerce/digsig/dsmain.htm

[VS00] https://www.verisign.com/repository/CPS1.2/CPS1.2.pdf

[Wise95] Dov Wisebrod, “Controlling the Uncontrollable: Regulating the Internet”, in http://www.catalaw.com/dov/docs/dw-inet.htm

[X500a] RFC 1308, in http://www.cis.ohio-state.edu/htbin/rfc/rfc1308.html

[X500b] ITU-T X.500, in http://www.mcg.org.br/mirrors/97x500Rev0.doc

[X509a] “Summary of ITU-T Recommendation X.509", in http://www.itu.int/itudoc/itut/rec/x/x500up/s_x509_e_30924.html

[X509b] ITU-T X.509v3, in http://www.mcg.org.br/mirrors/97x509final.doc