Explicit Meaningless X.509 Security Certificates as a Specifications-Based Interoperability Mechanism

Thierry Moreau

Document Number C004635

2008/07/23

Document Revision History

C-Number	Date	Explanation
C004635	2008/07/23	Initial release
C004635		Current version

Table of contents

1. Introduction....3
2. Overview of the PKC-Only Security Model....3
3. Rationales, Design Goals, and Intended Applicability....5
3.1 Overall Applicability....5
3.2 Compatibility with the Installed Base of PKI Standard Protocols....5
3.3 Orderly Cohabitation of Ad-Hoc and Standards-Based Network Traffic....6
3.4 Allowing Multi-Application Efficiencies for the End-User....6
3.5 Human Friendliness in Assignment of Certificate Distinguished Names....6
3.6 Privacy Protection Not a Design Goal....7
3.7 Illustrative Provisions for Interoperability with PKI Implementations....7
3.8 Document Value As a Tutorial....7
4. Important Normative Provisions....7
4.1 Explicit Meaningless Root CA Certificate....8
4.1.1 Essential X.509 Trust Anchor Data....8
4.1.1.1 Public Domain Private Key....9
4.1.1.2 Root CA Distinguished Name....9
4.1.2 Other Provisions for Root CA Certificate....11
4.2 Explicit Meaningless End-Entity Certificates....11
4.2.1 Mandatory Provisions....11
4.2.2 Other Provisions for End-Entity Certificates....12
4.2.2.1 No Human Readable Elements to Rely Upon....12
4.2.2.2 Interoperability Aspects....13
5. Security Considerations....13
6. References....14
6.1 Normative References....14
6.2 Informative References....14
Informative Annex A Data Representations for the Public-Private Key Pair....15
A.1 Public Domain Private Key....16
A.1.1 ASN.1 Data Representation....16
A.1.2 PEM Encoding....18
A.2 Meaningless Public Key....18
A.2.1 ASN.1 Data Representation....18
A.2.2 Sample Meaningless Self-Signed Security Certificate....18
A.2.3 PEM Encoding....19
A.3 Summary of Object Identifiers Usage....19
Informative Annex B User Friendliness Criteria Applied to Distinguished Names....20

List of tables

Table 1) The distinguished name associated with the public domain private key....10
Table 2) Authority key identifier value....12
Table A.1) Public domain private key representation in ASN.1 format....17

1. Introduction

The present specifications document relates to a PKC (Public Key Cryptography) distributed application security model that does not rely on PKI security certificates. Throughout this document, the acronym PKC refers to public key cryptography applied to security strategies devoid of the "Infrastructure" element of a PKI (Public Key Infrastructure). The PKC-only security model is only described in general terms for to achieve the foremost document objective, i.e. to specify minimal protocol and data representation provisions required for hosting the PKC-only security model in the widely deployed Internet security protocols.

In the context of the present document, an *end-entity* has a public/private key pair. In the prevailing fielded scheme for the PKI model in the Internet, i.e. web access to sites protected by the HTTP protocol over an SSL or TLS connection [RFC2818], the end-entity at the browser side needs neither a security certificate nor a public/private key pair. Accordingly, when the present documents refers to a departure from the PKI security model, it refers to a complete deployment in which client security certificates would be the norm.

Thus, the present document background is an application security scheme, called PKC-only, which rests on the installed base of PKI implementations, but without relying on the PKI trust model. In here, the minimal mandatory compliance provisions are specified for interoperability with the PKI standards, as the core formal arrangement for PKC-only scheme implementations presence within the installed base of PKI implementations. Many more out-of-scope issues need to be considered for the successful deployment of the PKC-only scheme, including e.g. software compatibility with widely deployed public key cryptography implementations. The Informative Annex A provides limited supplementary information to assist the conversion of formal standards compliance provisions into software and data representation components that hopefully fit the more day-to-day compatibility issues.

In its first release, the present document includes section titles that refer to contents that is out of scope with respect to the minimal mandatory provisions. These sections are essentially empty, except for an editorial note. On one hand, it is uncertain whether the possible contents for these sections would be as reliable as the mandatory provisions (e.g. possible contents could contradict the PKI standard interpretation embedded in some software implementations). On the other hand, some of the possible contents could turn into formal provisions in a future release of the present document, e.g. for to record precise PKI interoperability aspects that turn into inescapable behavior of PKC-only security model implementations using meaningless X.509 security certificates.

2. Overview of the PKC-Only Security Model

While the X.509 PKI technology can be described as "security certificate centric," the very foundation of public key cryptography rests on the assumption that each participant continuously controls the private counterpart of her public key. In practice when designing, deploying, and supporting applied PKC schemes, experts tend to overlook the importance of the basic principle, e.g. as being trivial or obvious; and the end-users are extensively trained about security rules derived from ramifications of PKI certificates.

The PKC-only security model makes the participant's private key the only security concern, or by far the participant's foremost security concern. If by some magic it was possible to deploy protocol changes to the Internet, there would be no PKI security certificate in any form in the participant's environment. The PKC-only security rests on the assumption that the server can validate client public keys with direct access to an on-line database. The arrangements for registering and managing clients and their key pairs and their authorizations is outside the scope of the present document. Also out of scope is the on-line database organization and its availability to application servers.

The present specifications document contributes to the feasibility of *explicitly meaningless PKI certificates* as a mechanism to deploy and support the PKC-only security model in the current PKI ecosystem in the Internet.

A meaningless end-entity certificate has an unreliable digital signature; hence every data element is unreliable, notably the end-entity *distinguished name* found in the certificate subject field. Thus, the security properties of a meaningless end-entity certificate are summarized in the following three observations.

- The certificate merely holds a subject public key as an informative data element.
- The certificate provides no reliable information about the association between the public key and an entity as would be the case for the certificate subject entity in a normal security certificate in the absence of security breaches.
- As always, the association between a public key value and its private key value counterpart is governed by the mathematical properties of public key cryptography, which thus should be invariant factors in a security analysis of germane versus explicitly meaningless root CA.

As usual in public key cryptography, a participant targeted by an end-entity certificate may be a physical person, an organizational unit, a device with processing capabilities, or a software function within a computer system. However, since the intended field of application replaces third party trust by an assumed trust database available to the remote party, a targeted participant is unlikely to be at the server side of a client-server e-commerce relationship.

At the heart of the present specifications document, a public domain private key allows the issuance of compatible meaningless security certificates at will, by any computing environment where it is convenient or useful to do so. This very ease of meaningless certificate issuance at once deprives the certificates from their third party trust conveying potential and provides the interoperability facilitation intended by the PKC-only security model.

The public domain private key specified below in subsection 4.1.1.1 also characterizes an *explicit meaningless root CA* (Certification Authority), an associated meaningless root CA *trust anchor*, and maybe a meaningless root CA *self-signed certificate*. These terms and the corresponding data elements in protocol frames and system configurations are governed by the PKI processing rules and are needed for interoperability purposes. For instance, if a client public key could be introduced in the SSL or TLS connection establishment phase outside of an X.509 security certificate object, the PKC-only application security model would not need the public domain private key, the meaningless root CA and so forth.

An alternative to the public domain private key is an e-commerce application scheme where the server blindly accepts any X.509 security certificate from clients, and authenticates the public key using the same direct access to an on-line database, exactly as needed with an explicitly meaningless certificate. This is deemed to scale as an operational nightmare because the independent certificate providers have no incentive to preserve the continuity of end-user association with a given public key value (i.e. client public key pairs are conveniently renewed upon certificate renewal). Stated differently, a PKI attempts to make the *current* certificate trustworthy, and expects the public key pair referenced by the certificate to be securely handled. In contrast, the PKC-only model puts the onus on the end-user to safeguard the private key as the single basis for authentication support, which implies a *lasting* end-user public-private key pair.

3. Rationales, Design Goals, and Intended Applicability

The simple rationale for meaningless PKI certificates is the power of the installed base of PKI schemes. The precise motivation for the present document is the desire to use TLS client public keys in e-commerce application schemes in which a server can validate such public keys with direct access to an on-line database. The current TLS protocol specification does not allow the use of a client public key without a certificate.

In the subsections below, further details are provided about the rationales, design goals, and intended applicability for the present specifications document.

3.1 Overall Applicability

Almost since the inception of public key cryptography, secure client-server applications could be envisioned such that client public key pairs are validated by servers through direct access to a trusted on-line database and, and no reliance on trust information distribution by the mechanism of PKI security certificates. It is outside of the scope of the present document to further describe application schemes of this type; it is sufficient to state that such schemes may exist, e.g. in a proprietary trust arrangement where application clients are directly registered to an Internet-enabled e-commerce operator or an affiliate organization having a privileged trust relationship.

The present specifications document focuses on non-reliance on X.509 security certificates as a workable substitute for a complete absence of security certificates in a PKC client-server application security scheme. It does so mainly through standardization of meaningless PKI security certificates, i.e. data structures.

3.2 Compatibility with the Installed Base of PKI Standard Protocols

The present standardization of meaningless PKI security certificates is generally intended to support interoperability with the installed base of PKI protocols, including [RFC5280], [X.501], [X.509], and [RFC4346]. This interoperability design goal pertains to operational use of X.509 security certificates, where the PKC-only application security scheme puts meaningless security certificates where the PKI protocols mandate a genuine security certificate carrying trust information according to the PKI model rules.

By design, the meaningless PKI security certificates specified herein avoid interoperability requirements related to PKI certificate management operations, e.g. reference [RFC4210].

3.3 Orderly Cohabitation of Ad-Hoc and Standards-Based Network Traffic

Nothing prevents the private and undocumented implementation of the basic idea of meaningless security certificates for compliance with interoperability requirements without reliance on certificate trust assurance potential. If the security management of such ad-hoc application scheme is properly effected, there would be no ambiguity about the certificate value for any participant and/or operator.

The present specifications document alludes to such practice and allows its use without restriction on the relying party, i.e. nothing of a proprietary or secretive nature in the present specification should prevent any relying party from using compliant meaningless security certificates as a mechanism to achieve protocol interoperability. Once so documented and applied according to the present specifications, ad-hoc use of the basic idea generates network traffic readily identified as using explicit meaningless certificates. This turns into orderly cohabitation of ad-hoc and standards-based network traffic, which is a rationale for the present document. See the security considerations section 5. The fact that the X.509 client security certificate mechanism deployment is limited does not change the relevance the orderly cohabitation of ad-hoc and standards-based network traffic.

3.4 Allowing Multi-Application Efficiencies for the End-User

Although the suggested basic PKC-only security scheme could require the end-user to register her public key with each independent e-commerce operator, nothing prevents the use of the same public key in every case. It may be efficient for the user to do so, e.g. when the acquisition costs for a public key pair is not negligible, such as with token-based public key cryptography.

It is a design goal to allow the use of a single public key, and preferably a single explicit meaningless certificate, across multiple applications. However, this design goal is of secondary relevance, i.e. it should not expand the scope of standardization beyond what is required for interoperability with PKI standard protocols and PKI implementations. For instance, the end-user name need not be coordinated among independent e-commerce operators, and thus is outside the scope of the present specifications.

3.5 Human Friendliness in Assignment of Certificate Distinguished Names

In a sense, the PKI technology is centered on name management, i.e. its basic function is to ascertain the link between a public key and an entity name. The foremost X.509 entity name construction and encoding rules are those of "distinguished names" defined in [X.501]. As further detailed in the informative annex J of the reference [X.501], distinguished names are intended to be human readable and preferably user friendly.

The distinguished names of explicit meaningless security certificates specified herein are required mainly for interoperability purposes, but can not be hidden from humans when the normal certificate handling process exposes names in human readable form. It is thus a design goal of the present document to include a reasonable solution to the dilemma of assigning human readable names to certificates as data objects deprived of their essential trust carrying function, hence unreliably linked to any specific entity.

3.6 Privacy Protection Not a Design Goal

The privacy implications of X.509 security certificate data fields are often seen as an impediment to the greater reliance on client-side public keys. In this case, the installed base main defect is the absence of "traffic flow confidentiality" in the TLS protocol, as it might be shown from an historical study of emerging PKC protocols at the time the SSL protocol was conceived. Irrespective of the root cause for the core X.509 privacy weakness, privacy concerns might motivate PKC-only application security schemes. Thus, privacy concerns may be an argument for to consider the meaningless certificate scheme, but privacy support as such is not a design goal for the present specifications.

3.7 Illustrative Provisions for Interoperability with PKI Implementations

The PKI installed base encompasses more than just the network protocol specifications. PKI standards contain endless provisions covering security certificate handling rules in local computer processes. The overall PKI scene is compounded by diverse interpretations of the standards, and plain non-compliant logic entrenched in widely used implementations. This creates an interoperability minefield. Even with the non-reliance on the trust carrying capability of explicit meaningless certificates, a definitive assurance of interoperability is an unrealistic target.

It is nonetheless a document editorial goal of the provide illustrative provisions addressing interoperability issues with existing PKI standard implementations.

3.8 Document Value As a Tutorial

It is an editorial goal for the present document to have some value as a tutorial for to promote the use of the public domain private key specified herein for the fielding of new Internet applications based on a PKC security model not relying on security certificates.

4. Important Normative Provisions

Two types of X.509 security certificates are potential targets of contemplated standard provisions: an explicit meaningless root CA (Certification Authority) certificate, and explicit meaningless end-entity certificates for participants in the PKC scheme.

In order to achieve the intended purpose, the present specification could provide normative provisions about various aspects of the meaningless root CA certificate and the meaningless end-entity certificate, basically in three categories:

1. a meaningless root CA public domain private key;
2. some human readable identification text for security certificates issued using the public domain private key, however potentially restricted to a single language due to the disambiguation requirement for automated certificate handling logic;
3. attributes and extensions needed for interoperability, and to be embedded in the security certificates issued using the public domain private key.

The category 2 definitely includes the issuer distinguished name, and could include the subject distinguished name. The single language restriction applies to the issuer distinguished name that needs to be globally unique.

Provisions in category 3 would be justified by the intended purpose of e-commerce without third party trust, in order to support the intended interoperability. In practice with the implementation diversity in the PKI installed based, provisions in the category 3 can hardly be definitive for every possible uses of certificates issued using the public domain private key.

4.1 Explicit Meaningless Root CA Certificate

4.1.1 Essential X.509 Trust Anchor Data

The essential parameters of an X.509 trust anchor are specified in section 6.1.1 of [RFC5280] and sections 3.3.60 and 10.1 of [X.509]. According to these, it is sufficient to define a) the trust anchor public key, b) a digital signature algorithm indication, and c) the exact CA name by which the public key may be referenced.

With the present meaningless security certificate scheme, the notion of trust anchor refers to the public key corresponding to the public domain private key that allows unrestricted generation of security certificates by any party in any computing environment. Thus, the trust anchor qualification is as meaningless as the security certificates generated in this context; it refers only to the use of the public key for interoperability purposes in the same role as a genuine X.509 trust anchor public key, also known as a root or trusted CA public key. The definition of a public domain private key implies the required public key definition. The definition of a CA name allows referencing for interoperability purposes.

Note: Actually, the practice of embedding the essential X.509 trust anchor data in a self-signed security certificate is not rooted in definite PKI standard provisions. In the PKC-only application security scheme, the servers operated by relying parties may require such self-signed certificates based on their local software environment, but any required self-signed certificate may be created and tailored by any party using the public domain private key.

4.1.1.1 Public Domain Private Key

The public domain private key is to be used with the RSA digital signature algorithm with the following core mandatory provisions:

- The signature scheme with appendix specified in section 8.2 and 9.2 of [RFC3447], using the SHA-1 as the hash function selection, SHALL apply to digital signature generation (respectively validation) using the public domain private key (respectively the corresponding trust anchor public key).

- The public domain private key modulus SHALL be the product of the prime numbers

  ```
  12223765854511530769064115990504106362482595762023720622684318319
  51889058868129609891630676827464082905755511412119178742463065097
  18784087338007614700520 87
  ```

  and

  ```
  11481808643981977405895116018975918518368605263997862331761888912
  29973266216248285057603007180959451279607897732835496720301348896
  25289234206927104520880 83
  ```

- The corresponding trust anchor public key SHALL have the public exponent value 65537.

See also the Informative Annex A for more convenient data representations of the key material.

4.1.1.2 Root CA Distinguished Name

The leftmost column in the table 1 starting on on page 10 defines the ASN.1 encoded distinguished name by which the above meaningless PKI trust anchor key may be referred. The two other columns are explanations of the ASN.1 syntax and semantic that might be inferred from the relevant PKI standard documents, notably [X.501] for the definition of a distinguished name and [X.520] for the four components in the defined name.

Hexadecimal representation of ASN.1 BER encoding	ASN.1 prefix indentation level	ASN.1 prefix explanation or textual representation of ASN.1 data value
30 81 ab	1	SEQUENCE OF (length=171)
31 0b	2	SET (length=11)
30 09	3	SEQUENCE OF (length=9)
06 03	4	OBJECT IDENTIFIER (length=3)
55 04 06		{2 5 4 6}, i.e. countryName
13 02	4	PrintableString (length=2)
41 41		"AA"
31 46	2	SET (length=70)
30 44	3	SEQUENCE OF (length=68)
06 03	3	OBJECT IDENTIFIER (length=3)
55 04 0a		{2 5 4 10}, i.e. organizationName
13 3d	4	PrintableString (length=61)
54 68 65 20 64 75 6d 6d 79 20 6e 61 6d 65 20 58 20 72 65 66 65 72 73 20 74 6f 20 74 68 65 20 6f 70 65 6e 6c 79 20 69 6e 73 65 63 75 72 65 20 70 75 62 6c 69 63 20 6b 65 79 20 2e 2e 2e		"The dummy name X refers to the openly insecure public key ..."
31 48	2	SET (length=72)
30 46	3	SEQUENCE OF (length=70)
06 03	4	OBJECT IDENTIFIER (length=3)
55 04 0b		{2 5 4 11}, i.e. organizationalUnitName
13 3f	4	PrintableString (length=63)
2e 2e 2e 20 6f 66 20 61 20 6e 6f 6d 69 6e 61 6c 20 43 41 20 64 65 76 6f 69 64 20 6f 66 20 6f 62 6a 65 63 74 69 76 65 20 50 4b 49 20 43 41 20 63 68 61 72 61 63 74 65 72 69 73 74 69 63 73 2e		"... of a nominal CA devoid of objective PKI CA characteristics."
31 0a	2	SET (length=10)
30 08	3	SEQUENCE OF (length=8)
06 03	4	OBJECT IDENTIFIER (length=3)
55 04 03		{2 5 4 3}, i.e. commonName
13 01	4	PrintableString (length=1)
58		"X"

Table 1) The distinguished name associated with the public domain private key.

The ASN.1 encoded definition of a name is unambiguous and representative of the protocol packet contents. Distinguished names are DER encoded when present in SSL/TLS "certificate request" messages defined in section 7.4.4. of [RFC4346]. However, PKI distinguished names are often communicated in textual form more or less compliant to the LDAP standard documents. The latter repeat and expand the PKI technological base: distnguished names are introduced in section 2.3.2 of [RFC4512], name components are covered in [RFC4519] and the textual representation is introduced in [RFC4514]. With these conventions, the distinguished name is specified as

```
CN=X,OU=... of a nominal CA devoid of objective PKI CA characteristics.,
O=The dummy name X refers to the openly insecure public key ...,C=AA
```

Formally, this is an incomplete specification because the name component string values might validly be encoded with ASN.1 types other than PrintableString. Moreover, the formal listing order for the name components is reversed from the maybe more natural top down approach that is often used in various publications and operator display in software products.

The presence of the nonexistent country code "AA" (freely user assigned according to [ISO3166TABLE]) serves a dual purpose: a) to make more explicit which of the bottom up or top down listing order is in use, and b) to isolate the public domain private key from the nationality of any specific organization. The rationale for the writing style of this distinguished name is further explained in the Informative Annex B.

4.1.2 Other Provisions for Root CA Certificate

Editorial note: In the current state of the document, this section is empty. Possible contents is under investigation and it is uncertain whether it could be as reliable as the provisions in other sections. Such investigation could reveal PKI interoperability aspects that would deserve a formal compatibility provision.

4.2 Explicit Meaningless End-Entity Certificates

4.2.1 Mandatory Provisions

In the X.509 certificate validation model, the CA subject name is present as the issuer name in security certificates signed directly by the CA.

Implementations claiming compliance to the present specification MUST NOT issue security certificates having the root CA distinguished name specified in 4.1.1.2 as the issuer name and a digital signature using a private key other than the one specified in 4.1.1.1.

Conversely, implementations claiming compliance to the present specification MUST NOT issue security certificates digitally signed using the private key specified in 4.1.1.1 and not having the root CA distinguished name specified in 4.1.1.2 as the issuer name.

Security certificates digitally signed using the private key specified in 4.1.1.1 and having the root CA distinguished name specified in 4.1.1.2 as the issuer name MUST include the authority key identifier certificate extension value using the first method indicated in section 4.2.1.2 of [RFC5280] (the authority key identifier extension itself is defined in section 4.2.1.1.in the same reference). This establishes the key identifier value shown in the leftmost column in the table 2 starting on page 12 from the SHA-1 hash fingerprint of the ASN.1 encoded RSA public key. This compliance provision merely ensures that two independent PKC-only scheme implementations avoid an interoperability pitfall built-in the PKI standards, given the lack of readily identifiable security benefit from the authority key identifier mechanism.

Hexadecimal representation of ASN.1 BER encoding	ASN.1 prefix indentation level	ASN.1 prefix explanation or extplanation of ASN.1 data value
30 16	1	SEQUENCE OF (length=22)
80 14	2	CONTEXT defined type 0 (length=20)
b0 05 f3 b7 e4 34 9e e6 b3 8a 85 80 a2 60 c6 56 94 1c 6b 93		<public key fingerprint>

Table 2) Authority key identifier value

Implementation note: Other interoperability pitfalls may exist in the PKI standards provisions applicable to end-entity certificates. Such a pitfall could be impossible to avoid, e.g. if a given PKC-only application requirement is incompatible with another application context. This is unlikely in the case of the authority key identifier extension, which is a characteristic of the meaningless CA public key common to every explicit meaningless end-entity certificate. In the case of unreconciled application requirements, an end-entity would require a different explicit meaningless security certificate for each application usage, each carrying the same client public key value according to the PKC-only security model.

4.2.2 Other Provisions for End-Entity Certificates

4.2.2.1 No Human Readable Elements to Rely Upon

Implementations claiming compliance to the present specification SHOULD NOT put specific human readable information on which users are expected to rely, or could rely according to the information face value, in data elements of security certificates digitally signed using the private key specified in 4.1.1.1. See also the Informative Annex B.

4.2.2.2 Interoperability Aspects

Editorial note: In the current state of the document, this section is empty. Possible contents is under investigation and it is uncertain whether it could be as reliable as the provisions in other sections. Such investigation could reveal PKI interoperability aspects that would deserve a formal compatibility provision.

5. Security Considerations

The very idea of using meaningless X.509 security certificates signed with a public domain private key introduces a PKC-only security model in a distributed application environment based on a full PKI model, and the security implications should be carefully weighted. A main area of care should be the relying party processes that handle on session data received from remote end entities having used an explicit meaningless certificate for session authentication purposes. In server sites where multiple client authentication schemes may be supported, the PKC-only scheme based on meaningless certificates needs specific processing according to local rules for client public key validation.

Privacy protection is not a design goal for the explicit meaningless certificates mechanism for PKC-only application security. Nonetheless, the client data privacy aspects include pitfalls and opportunities which should be thoroughly studied for the deployment of any specific PKC-only scheme. For instance, a software utility that generates a meaningless certificate might query the local environment, maybe by requesting input from the user, for client identification data, which can only be detrimental to client data privacy. Also, a privacy protection opportunity arises from the facts that the PKC-only scheme a) embeds client authentication capability without any authentication secret transmission (e.g. password), and b) uses a human unfriendly client identification field, namely the very client public key value. The public key value used as an index in a server's client database is not a cryptography-based anonymity scheme, but it is at least a small improvement over cleartext transmission of a full featured X.509 security certificate. In any event, it should be kept in mind that technology-intensive mechanisms are of little use against social engineering attacks, and any effective line of defense includes well-studied end-user awareness promotion and training activities. The simplistic user model for the PKC-only security scheme is deemed to be a good start for an end-user awareness program with social engineering resistance.

An attack scenario for a PKI application is to induce a relying party to trust a root CA public key of which the private counterpart is available to the adversary. The adversary would thus be in a position to issue misleading security certificates to any end entity. The network traffic pattern induced by this attack scenario is similar to the one induced by the legitimate but private and undocumented use of meaningless security certificate mentioned at the beginning of section 3.3: operational use of the security certificate, but no certificate management protocol traffic with any genuine CA entity. The explicitness provided by compliance with the present specification isolates hidden private use of meaningless or bogus certificates, including dubious network activity triggered by the attack scenario.

6. References

6.1 Normative References

[RFC3447]
J. Jonsson, B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography, Specifications Version 2.1", IETF Network Working Group, RFC 3447, February 2003

[RFC4346]
T. Dierks, E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", IETF Network Working Group, RFC 4346, April 2006

[RFC5280]
D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", IETF Network Working Group, RFC 5280, May 2008

[X.501]
International Telecommunication Union, "Information technology - Open Systems Interconnection - The Directory: Models", ITU-T Recommendation X.501, August 2005

[X.509]
International Telecommunication Union, "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks", ITU-T Recommendation X.509, August 2005

[X.520]
International Telecommunication Union, "Information technology - Open Systems Interconnection - The Directory: Selected attribute types", ITU-T Recommendation X.520, August 2005

6.2 Informative References

[ISO3166TABLE]
ISO 3166 Maintenance agency (ISO 3166/MA), "ISO 3166-1 decoding table", http://www.iso.org/iso/iso-3166-1_decoding_table, visited on June 18, 2008

[RFC2313]
B. Kaliski, "PKCS #1: RSA Encryption Version 1.5", IETF Network Working Group, RFC 2313, March 1998

[RFC2818]
E. Rescorla, "HTTP Over TLS", IETF Network Working Group, RFC 2818, May 2000

[RFC3279]
W. Polk, R. Housley, L. Bassham, "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", IETF Network Working Group, RFC 3279, April 2002

[RFC4210]
C. Adams, S. Farrell, T. Kause, T. Mononen, "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)", IETF Network Working Group, RFC 4210, September 2005

[RFC4512]
K. Zeilenga, "Lightweight Directory Access Protocol (LDAP): Directory Information Models", IETF Network Working Group, RFC 4512, June 2006

[RFC4514]
K. Zeilenga, editor, "Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names", IETF Network Working Group, RFC 4514, June 2006

[RFC4519]
A. Sciberras, editor, "Lightweight Directory Access Protocol (LDAP): Schema for User Applications", IETF Network Working Group, RFC 4519, June 2006

Informative Annex A Data Representations for the Public-Private Key Pair

This informative annex provides useful data representations for the key pair, i.e. the public domain private key specified in section 4.1.1.1 and its public key counterpart.

The reference [RFC3447] (notably section A.1 in Appendix A) specifies data representations for RSA private and public keys (the section 2.3.1 in the reference [RFC3279] repeats the public key format specification). These rules are applied below, respectively in the following section A.1.1 for the private key and in section A.2.2 for the public key embedded in a sample self-signed security certificate.

The document section 4.1.1 introduced the somehow abstract notion of a trust anchor. With the prevailing PKI practice, the interchange format for the trust anchor is a self-signed X.509 security certificate, which is applied in the sample given is the following section A.2.2. Note that the PKI trust model implies no need to qualify the sample self-signed certificate itself as "meaningless" since self-signed certificates in general are trustworthy only through the application of "out of band" trust distribution mechanisms. In any event, self-signed security certificates are usually handled with the issuer distinguished name as their identification, and the issuer distinguished name is already quite explicit.

The installed base of PKI software recognizes the "PEM encoded" format for public key material and security certificates, which facilitates data interchange through ASCII representation of

DER encoded ASN.1 data. The PEM encoding for the private key is provided in the following section A.1.2. The PEM encoded version of a sample self-signed security certificate is to be found in section A.2.3.

A.1 Public Domain Private Key

A.1.1 ASN.1 Data Representation

The table A.1 starting on page 17 lists the public domain private key in ASN.1 format according to section A.1.2 in the appendix A of [RFC3447].

ASN.1 Syntax Rule	Hexadecimal Representation of Data Values
RSAPrivateKey ::= SEQUENCE {	
version Version,	00
modulus INTEGER, -- n	C7DDCAA78D6B07D7F2C57B9B5FD67F9DE2174673F902A9 CD083CE655F2A52BCC52631DEDF57409D2C8880E011A46E 642927D8895E9E9E2318D7188D189DED39D8E17988F0B3ED 3475316201432DEFB0A4A9F6734D0FCBDB9ED388F37CD9 EA0571B0A0F3DFC7D84B7075174878A0A85A76AF7E28CF B04A1235B0A161FB66F6F55
publicExponent INTEGER, -- e	10001
privateExponent INTEGER, -- d	799B0E09C236C40FF56B88D8B1882E1F9F07B05B31C01816 6313C2C5BA9C1AB8F7CBCCD3130C2649F4AF2B6E2A51C7 19EC4DAB0423CDA54DA714D43D41D8AD01D5A4FDABB CADD1E43C252E4DD8569F3F6FE1A31B2CA72A7A7ABF35 39EFADE4F56F3D7936449019DB2882880CE74C6C9DD4FEC 71AE01CF40FBCC061E5B8B12EE5
prime1 INTEGER, -- p	E96485DC497AE8AA01BAFA53A68A0C915C33C7075204801 25CC6188D110EE820FFA3E52AE11F44984DFE040934E7BA 118286C3F28EDA80BC881DDD8C267566F7
prime2 INTEGER, -- q	DB39E95E1E745FE8A19DF3A659A617177C9E2F0B69D5292 8561F73CA320B106BC075A09C235F223D47F4BF0F363D8F A080EA193B5A5305707B1BBCF6BFA3CD13
exponent1 INTEGER, -- d mod (p-1)	8C451A1E971B0392898FB2BB6BB034757890B5E1B46D77F B913E8DAADAC1B6C97E5B26746AF0BBED3D9299225F659 5F2C81BEED06D02EDB461AA7168B6048169
exponent2 INTEGER, -- d mod (q-1)	014DF8EB8C76D8D34392F30E3C5E56A8D71F01DCA986913 6223D11AD545AE8801BA7178B96524C9BDCBFC21B4F1152 B77BCCE8D5879612E4367755CA8A1D696D
coefficient INTEGER, -- (inverse of q) mod p	A63279F950E850E92BEF5EF3D45230D8D5B72AB394BC5AF 66A3CEF95510A17D598DCED3913929863B9330EDDD31A85 7D81FBEC67ED37D7D6EDA7BC6CBD39511E
otherPrimeInfos OtherPrimeInfos OPTIONAL	
}	

Table A.1) Public domain private key representation in ASN.1 format.

A.1.2 PEM Encoding

The public domain private key specified in section 4.1.1.1 is shown below using the PEM format.

```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQDH3cqnjWsH1/LFe5tf1n+d4hdGc/kCqc0IPOZV8qUrzFJjHe31
dAnSyIgOARpG5kKSfYiV6eniMY1xiNGJ3tOdjheYjws+00dTFiAUMt77CkqfZzTQ
/L257TiPN82eoFcbCg89/H2EtwdRdIeKCoWnavfijPsEoSNbChYftm9vVQIDAQAB
AoGAeZsOCcI2xA/1a4jYsYguH58HsFsxwBgWYxPCxbqcGrj3y8zTEwwmSfSvK24q
UccZ7E2rBCPNpU2nFNQ9QditAdWk/au8rdHkPCUuTdhWnz9v4aMbLKcqenq/NTnv
reT1bz15NkSQGdsogogM50xsndT+xxrgHPQPvMBh5bixLuUCQQDpZIXcSXroqgG6
+lOmigyRXDPHB1IEgBJcxhiNEQ7oIP+j5SrhH0SYTf4ECTTnuhGChsPyjtqAvIgd
3YwmdWb3AkEA2znpXh50X+ihnfOmWaYXF3yeLwtp1SkoVh9zyjILEGvAdaCcI18i
PUf0vw82PY+ggOoZO1pTBXB7G7z2v6PNEwJBAIxFGh6XGwOSiY+yu2uwNHV4kLXh
tG13+5E+jarawbbJflsmdGrwu+09kpkiX2WV8sgb7tBtAu20YapxaLYEgWkCQAFN
+OuMdtjTQ5LzDjxeVqjXHwHcqYaRNiI9Ea1UWuiAG6cXi5ZSTJvcv8IbTxFSt3vM
6NWHlhLkNndVyoodaW0CQQCmMnn5UOhQ6SvvXvPUUjDY1bcqs5S8WvZqPO+VUQoX
1Zjc7TkTkphjuTMO3dMahX2B++xn7TfX1u2nvGy9OVEe
-----END RSA PRIVATE KEY-----
```

A.2 Meaningless Public Key

A.2.1 ASN.1 Data Representation

The meaningless public key does not occur in TLS security protocol packets, so its data representation is somehow irrelevant outside of local software compatibility issue. Protocol-wise, the key is referred through the distinguished name specified in section 4.1.1.2. The intricacies of ASN.1 syntax apply when the public key is included in a self-signed security certificate, but there is nothing special about the public key specified herein in this respect. Nonetheless, among the ASN.1 intricacies, there are three object identifiers linked to the selection of a specific RSA signature scheme in document section 4.1.1.1. Further explanations is to be found in section A.3.

A.2.2 Sample Meaningless Self-Signed Security Certificate

Editorial note: In the current state of the document, this section is empty. Possible contents is under investigation and it is uncertain whether it could be as reliable as the provisions in other sections. Such investigation could reveal PKI interoperability aspects that would deserve a formal compatibility provision.

A.2.3 PEM Encoding

Editorial note: In the current state of the document, this section is empty. Possible contents is under investigation and it is uncertain whether it could be as reliable as the provisions in other sections. Such investigation could reveal PKI interoperability aspects that would deserve a formal compatibility provision.

A.3 Summary of Object Identifiers Usage

The use of ASN.1 object identifier values in public key cryptography algorithms may be confusing. With the present limited compliance to PKI standards, there are two distinct routes through which object identifiers enter the picture:

- the X.500 distinguished name definition (section 4.1.1.2) use object identifiers to specify the type of name components, which is unrelated to the use of any specific cryptographic algorithm, and

- the selection of a specific RSA signature scheme in document section 4.1.1.1 triggers the presence of three object identifier values in implementations of the present specifications, each with its own context and significance, as explained below.

The RSA public key is identified by an object identifier specified both in [RFC3447] and [RFC3279]. The former reference links the object identifier to algorithmic details including private key representation; the latter one links to X.509 security certificate structure element ([RFC5280]). Despite the actual text of the two references, they are related: the citation of [RFC2313] in the reference [RFC3279] is actually a predecessor of [RFC3447]. The object identifier value that identifies a public key as an RSA public key is

```
rsaEncryption OBJECT IDENTIFIER ::={ 1 2 840 113549 1 1 }
```

The digital signature occurring in a security certificate is labeled with an algorithm identifier where another object identifier value is found. This indication refers to the precise RSA signature scheme selected in document section 4.1.1.1, and implies that the SHA-1 hash algorithm is used in the signature operations. The object identifier value that identifies a digital signature as the scheme selected in document section 4.1.1.1 is

```
sha1WithRSAEncryption OBJECT IDENTIFIER ::={ 1 2 840 113549 1 5 }
```

Confusion may arise among these two object identifier values from their common occurrence in ASN.1 structure type called "AlgorithmIdentifier" which qualifies a public key in the first case and a digital signature in the second case.

The third mandatory object identifier value is used in the context of digital signature operations, and more precisely as a hash algorithm indication encoded during the initial step of signature

generation where the hash operation is performed. This value is not directly reflected in either the digital signature value or the security certificate. The object identifier value is

```
id-sha1 OBJECT IDENTIFIER ::={ 1 3 14 3 2 26 }
```

These three object identifier values are mandated by implication when compliance with the present specification is asserted in a meaningless security certificate with the issuer name specified in section 4.1.1.2. They do not apply to the client public key found in the security certificate: the PKC-only security scheme makes no a-priori restriction on the type of public key algorithms used by clients, as long as the client public key may be represented in an X.509 security certificate in a syntax mutually recognized by the client and server systems. The object identifiers used for client generated digital signatures (and client-side public key decryption capability) are out of scope for the present document.

Informative Annex B User Friendliness Criteria Applied to Distinguished Names

According to annex J of [X.501], distinguished names should be user friendly. With the present technical specifications, the user friendliness design criteria turns into the need for a user message conveying the *meaningless property* of digital signatures using a public domain private key.

The meaningless property has direct consequences essentially for relying parties in the PKI model. When a legitimate user of a private key submits a meaningless security certificate in an e-commerce session, she is not in the relying party position and she may expect the remote party to accept the key occurrence in a digital signature as valid authentication of her participation in the e-commerce scheme. But neither her nor any other person or entity claims endorsement by a third party based on the meaningless security certificate. In such typical e-commerce scheme envisioned by the present document, the relying party is an automated server process.

Thus, in the remaining of the present section, a representative "end-user" for whom user friendliness is intended would be among the support personnel for an e-commerce operation.

There are the two distinguished names involved in the meaningless security certificate scheme: the root CA distinguished name targeted by mandatory provisions in subsection 4.1.1.2, and the end entity distinguished name subject to the recommendation in subsection 4.2.2.1.

Instructions about what to do and what to avoid should be given to the end-user by means other than the meaningless root CA distinguished name. In whatever context the user sees the name by itself, she should receive neutral summary information about the CA referred to by the name. Here is a tentative explanation for this approach: if the meaningless CA distinguished name appears in a list of CA subject names and/or other security certificate subject names, either such neutral summary information is sufficient for the user to handle the meaningless CA in the proper way, or no amount of care and foresight in the meaningless CA labeling might induce the user to do so with any useful success probability. When the user sees the meaningless CA distinguished name, her appropriate action for to achieve the task at hand in a secure manner may be either way (e.g. continue current user

interaction or abort, select this CA versus another, ...) and nothing in the static CA name can anticipate the many possible contexts. These observations also apply when a distinguished name appears as the issuer name of an end-entity security certificate.

Thus, the user message embedded as a distinguished name should not attempt to go beyond a simple statement of the meaningless property. Specifically,

- the meaningless CA distinguished name should not refer to any organization or entity;
- the meaningless CA distinguished name should not refer to any specific application use of the present scheme;
- the meaningless CA distinguished name should not convey any guidance to the user for to avoid misleading use of digital signatures using a public domain private key;
- the meaningless CA distinguished name should not refer to the possible misleading use of digital signatures using a public domain private key.