Entrust Technologies White Paper

Version 3 X.509 Certificates

Author: Ian Curry
Date: July 1996
Version: 1.0

1. Introduction

This document provides a description of the version 3 X.509 certificate standard. Primarily, this document focuses on describing the fields contained in X.509 public key certificates. Version 3 of the X.509 standard for public key certificates allows for flexible extensions to certificates. In addition to the traditional fields in public key certificates (i.e., those defined in versions 1 and 2 of X.509), this paper discusses the version 3 extensions referred to as *standard extensions*.

As the version 3 X.509 standard was only recently ratified, Entrust currently does not support all of the standard extensions. The intent of this document is to provide Entrust customers with a clear understanding of the certificate format so they can provide feedback to Entrust Technologies stating their requirements. As such, the paper does not present the standard in detail or in full completeness, but rather at a level intended to inform the reader of important issues surrounding the various fields in certificates.

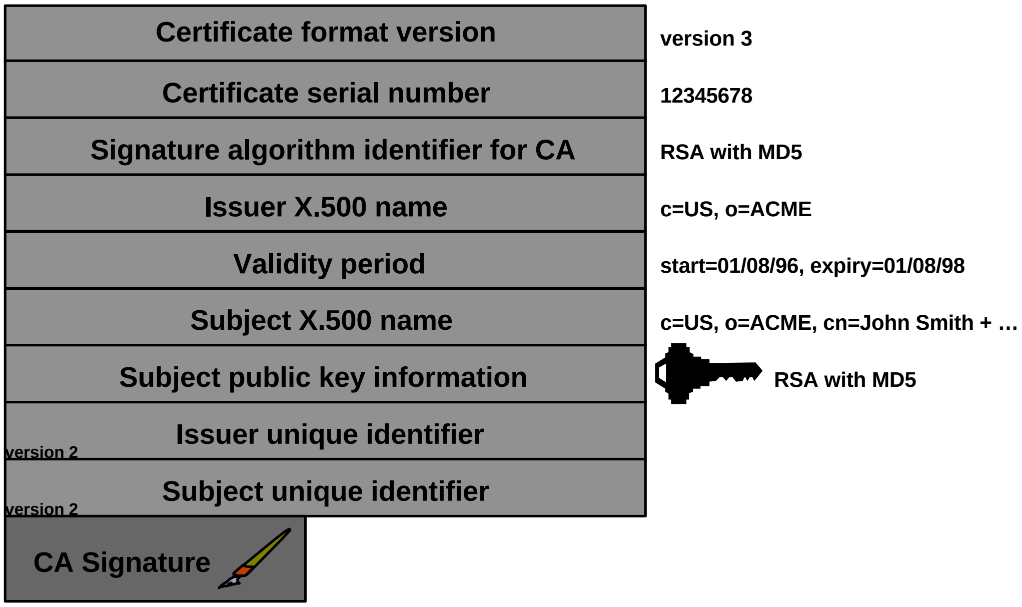
Initially, the paper describes the fields in versions 1 and 2 of the X.509 certificate standard. The paper then describes the standard extensions and the general extension mechanism as defined in the version 3 standard. The paper does not describe the version 2 X.509 Certificate Revocation List (CRL) format.

The paper assumes readers understand the purpose and general uses of certificates. For those readers wishing to understand more about the purpose and uses of certificates, refer to the White Paper titled "The Concept of Trust in Network Security."

2. Version 1 and 2 X.509 Certificates

This section of the paper describes the fields published in the version 1 and 2 X.509 standards for public key certificates. Version 1 of the standard was ratified in 1988. The version 2 standard, ratified in 1993, contained only minor enhancements to the version 1 standard.

The following diagram shows the fields in the version 1 and 2 certificate standards.


Certificate format version
version 3
Certificate serial number
12345678
Signature algorithm identifier for CA
RSA with MD5
Issuer X.500 name
c=US, o=ACME
Validity period
start=01/08/96, expiry=01/08/98
Subject X.500 name
c=US, o=ACME, cn=John Smith + …
Subject public key information
RSA with MD5
Issuer unique identifier
version 2
Subject unique identifier
version 2
CA Signature

The following subsections describe each of the fields in a version 1 or 2 certificate. Entrust uses all of the fields in the version 1 certificate and does not use the fields in the version 2 extensions.

Version

The *version* field indicates the X.509 version of the certificate format (1, 2, or 3), with provision for future versions of the standard.

Serial Number

The *serial number* field specifies the unique, numerical identifier of the certificate in the domain of all public key certificates issued by the Certification Authority (CA). When a certificate is revoked, it is actually the certificate serial number that is posted in a certificate revocation list signed by the CA (posting the entire certificate would be wasteful and is completely unnecessary). It is for this reason that the serial number for each certificate in the domain *must* be unique.

Signature Algorithm

The *signature algorithm* field identifies the algorithm used by the CA to sign the certificate. The algorithm identifier, which is a number registered with an internationally-recognized standards organization (e.g., ISO), specifies both the public-key algorithm and the hashing algorithm (e.g., RSA with MD5) used by the CA to sign certificates.

Issuer X.500 Name

The *issuer X.500 name* field specifies the X.500 distinguished name (DN) of the CA that issued the certificate; for example, the DN *c=US, o=ACME Corporation* might be used as the DN for the CA issuing certificates to the employees of the ACME Corporation in the United States.

Validity Period

The *validity period* field specifies the dates and times for the start date and the expiry date of the certificate. Every time a certificate is used in Entrust, the software examines the certificate to ensure it is still within its validity period.

Subject X.500 Name

The *subject X.500 name* field specifies the X.500 distinguished name (DN) of the entity holding the private key corresponding to the public key identified in the certificate; for example, the DN *c=US, o=ACME Corporation, cn=John M. Smith* might be the DN for employee John Smith of the ACME Corporation.

Subject Public Key Information

The *subject public key information* field identifies two important pieces of information: a) the value of the public key owned by the subject, and b) the algorithm identifier specifying the algorithm with which the public key is to be used. The algorithm identifier specifies both the public-key algorithm and the hashing algorithm (e.g., DSA with SHA-1).

Issuer Unique Identifier (version 2 only)

The *issuer unique identifier* field was added to the X.509 certificate definition as part of the version 2 standard. The field, which is optional, provides a location to specify a bit string to uniquely identify the *issuer X.500 name*, in the event that the same X.500 name has been assigned to more than one CA over time.

Subject Unique Identifier (version 2 only)

The *subject unique identifier* field was added to the X.509 certificate definition as part of the version 2 standard. The field, which is optional, provides a location to specify a bit string to uniquely identify the *subject X.500 name*, in the event that the same X.500 name has been assigned to more than one subject over time (e.g., one John M. Smith leaves ACME Corporation and a second John M. Smith joins ACME Corporation two months later).

This field is not used by Entrust for various reasons, although primarily because there are more convenient ways to uniquely identify a subject. Specifically, Entrust uses the *serialNumber* attribute in an X.500 common name as the means to uniquely identify certificate subjects. Generally, customers using Entrust specify the subject's unique employee number, or equivalent, in the *serialNumber* attribute. Such a scheme fits well within an organization's administrative and directory management procedures because employees require a unique identifier in their X.500 common names anyways (e.g., to handle the case where there are two John M. Smith's in the organization at the same time).

3. Version 3 X.509 Certificates

This section of the paper describes the standard extension fields published in the version 3 X.509 standard. Version 3 introduces a mechanism whereby certificates can be "extended," in a standardized and generic fashion, to include additional information. There are numerous reasons why this additional information is required, and some of these reasons will be discussed as the standard extensions are presented.

The term *standard extensions* refers to the fact that the version 3 X.509 standard defines some broadly-applicable extensions to the version 2 certificate. However, certificates are not constrained to only the standard extensions and anyone can register an extension with the appropriate authorities (e.g., ISO). Over time, it is expected that new broadly-applicable extensions will be added to the set of standard extensions. It is important to recognize, however, that the extension mechanism itself is completely generic.

Each extension consists of three fields: *type*, *criticality*, and *value*. The following diagram shows the structure of an extension:

Type	**Criticality**	**Value**

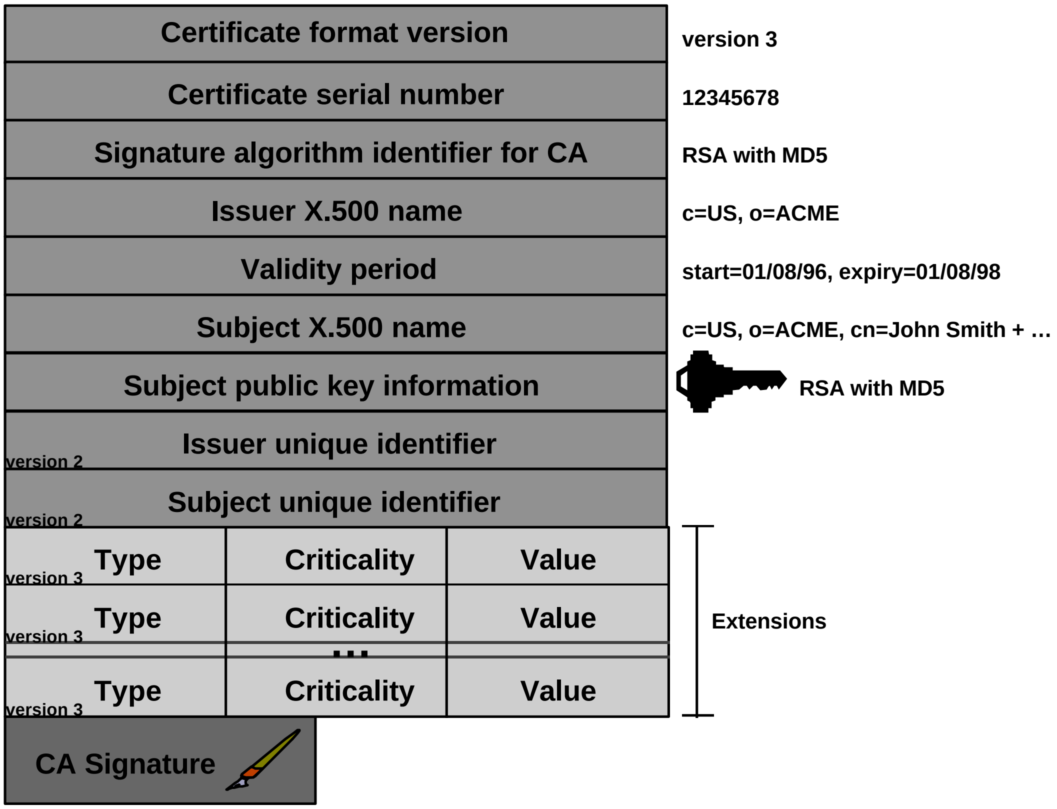
The *extension type* field defines the type of the data in the *extension value* field. The type could, for example, represent a simple text string, a numerical value, a date, a graphic, or a complex data structure. To promote interoperability, all extension types should be registered with an internationally-recognized standards organization.

The *extension criticality* field is a single-bit flag. When an extension is flagged as critical, it indicates that the associated *extension value* contains information of such importance that an application cannot ignore the information. If a particular certificate-using application cannot process a critical extension, the application should reject the certificate.

There is an important distinction between a critical extension and required information in a certificate. A particular application may require that a certain extension be available in any certificate processed by the application. This, however, does not imply that the extension needs to flagged as critical. Critical extensions are only intended for information so important that the data must be understood by all applications (e.g., information critical to preventing misuse or unsafe use of a certificate). Consequently, the vast majority of extensions are non-critical. Critical extensions should only be added to a certificate after much consideration and with the understanding that doing so could create interoperability problems with other CA security domains and applications.

The *extension value* field contains the actual data for the extension. The format of the data is reflected in the *extension type* field.

The following diagram shows the format of version 3 X.509 certificates with the extension mechanism:



The remainder of this section discusses the version 3 X.509 standard extensions. The standard extensions for public key certificates can be separated into the following groups:

- key information
- policy information
- user and CA attributes
- certification path constraints

3.1 Key Information Extensions

The standard extensions present four fields providing information about the intended uses of a public key pair and a certificate: *authority key identifier*, *subject key identifier*, *key usage*, and *private key usage period*.

Authority Key Identifier

The *authority key identifier* field specifies a unique identifier of the key pair used by the CA to sign the certificate. This identifier aids in the process of verifying a certificate signature in the case where a CA has used multiple key pairs in its lifetime (e.g., when the CA's signing key pair is updated over time).

This field is specified by Entrust in users' encryption and verification public key certificates.

Subject Key Identifier

The *subject key identifier* field serves much the same purpose as the *authority key identifier* field. The *subject key identifier* field is used to identify the particular key pair associated with the public key in the certificate. This field is useful when a user has updated his key pairs (both signing and encryption) multiple times during his existence in the CA security domain. In such a case, the *subject key identifier* field is most useful when a user is attempting to decrypt a file encrypted for him with a public key that is not his current encryption public key.

This field is specified by Entrust in users' encryption and verification public key certificates.

Key Usage

The *key usage* field specifies the intended use(s) of the key. The following list represents the settings for the *key usage* field: non-repudiation, certificate signing (e.g., a CA key pair), CRL signing (e.g., a CA key pair), digital signature (other than non-repudiation, certificate signing, or CRL signing), symmetric key encryption for key transfer, data encryption (other than a symmetric key), and Diffie-Hellman key agreement.

The *extension criticality* field denotes two separate uses for the *key usage* field. If the extension is noted as critical, then the key in the certificate is *only* to be applied to the stated uses. To use the key for another purpose in this case would be to break the CA's policy. If the extension is not noted as critical, the *key usage* field is simply there as an aid to help applications find the proper key for a particular use.

This field is specified by Entrust in users' encryption and verification public key certificates.

Private Key Usage Period

The *private key usage period* field specifies the date on which the signing private key expires for a user's digital signature key pair. There is no such requirement for decryption private keys because they never expire.

This field is used by Entrust to specify either: (i) the date prior to which the digital signature key pair must be updated if automatic key update is specified; or (ii) the date on which the signing private key expires in the case where a user's key pair is not to be updated.

3.2 Policy Information Extensions

The policy information extensions provide a mechanism for the CA to distribute information regarding the ways a particular certificate should be used and interpreted. There are two fields specified as policy information extensions: *certificate policies* and *policy mappings*.

Certificate Policies

The *certificate policies* field specifies the policies under which the certificate was issued to the user and/or the types of uses applicable to the certificate. Certificate policies are represented by specially-formatted numbers, known as object identifiers, which are registered with an internationally-recognized standards organization. It is possible to designate a number of certificate policies within a certificate; naturally, the specified policies for a particular certificate cannot be conflicting.

If the *certificate policies* field is set to be non-critical, the CA indicates which policies apply to the certificate, but is not requiring the certificate to be limited in use to situations only in accordance with those policies. If the field is flagged as critical, the CA is specifically limiting use of the certificate to situations in accordance with the policies.

Entrust will support the *certificate policies* field in release 3.0.

Policy Mappings

Whereas the *certificate policies* field applies to both user certificates and CA cross-certificates, the *policy mappings* field only applies to cross-certificates. A cross-certificate is created by one CA when it certifies the verification public key of a different CA.

The *policy mappings* field provides a mechanism for the signing CA to map its policies to the policies of the CA specified in the cross-certificate. This policy mapping information is critical when an application is processing a certificate chain that crosses CA domain boundaries. An application uses the mapping information to ensure that a consistent and acceptable policy (or set of policies) applies to all certificates in the chain.

3.3 User and CA Attribute Extensions

The user and CA attribute extensions provide additional mechanisms to specify identifying information (e.g., name types) for a user or CA. To maintain coherence with the X.509 standard, the remainder of this subsection refers to users as *subjects* and CAs as certificate *issuers*.

Subject Alternative Name

The *subject alternative name* field specifies one or more unique names for the certificate subject. The permissible name forms are as follows:

- Internet e-mail address
- Internet domain name
- Internet IP address
- X.400 e-mail address
- EDI party name
- Web uniform resource identifier (a uniform resource locator, or URL, is a sub-type of the uniform resource identifier)
- any other name type with a recognized object identifier

The purpose of these additional name forms is to support applications, such as e-mail, where a user's name must be unique, but it is not the same as the user's X.500 distinguished name.

Release 3.0 of Entrust provides support for specifying a subject's Internet e-mail address as part of the certificate.

Issuer Alternative Name

The *issuer alternative name* field specifies one or more unique names for the CA. The permissible name forms are the same as those for the *subject alternative name* field, provided above.

Subject Directory Attributes

The *subject directory attributes* field provides for additional X.500 Directory attributes to be included in the subject's certificate. This field could be used to specify additional identification information beyond that provided in the *subject X.500 name* and *subject alternative name* fields. Organizations must choose attributes that change infrequently, so as to minimize the potential administrative overhead involved in re-issuing certificates due to changes in non-essential information.

3.4 Certification Path Constraints Extensions

The certification path constraints extensions provide mechanisms for a CA to control and limit extended third-party trust in a cross-certified environment. There are three fields provided: *basic constraints*, *name constraints*, and *policy constraints*.

Basic Constraints

The *basic constraints* field simply indicates whether or not the subject of the certificate may act as an end user only or as a CA. If the subject is a CA, then the certificate is a cross-certificate. A cross-certificate may also specify the maximum acceptable length of a certificate chain beyond the cross-certificate. If the length is specified as 1, for example, then users may only verify end-user public key certificates and CRLs issued by the CA specified in the cross-certificate (i.e., third-party trust cannot extend transitively beyond the domain of the CA specified in the cross-certificate).

Entrust only uses this extension to denote the type of certificate (i.e., an end user public key certificate or a cross-certificate). Currently, Entrust does not support specification of a certification path length constraint in a cross-certificate; however, Administrators can limit chain lengths through settings in the Entrust configuration file on client-side workstations (i.e., in the *entrust.ini* file).

Name Constraints

The *name constraints* field is used in cross-certificates. The field provides administrators with a mechanism to restrict the domain of trustworthy names in a cross-certified environment. Whereas the *basic constraints* field provides a simple mechanism to restrict chain lengths, the *name constraints* field provides a sophisticated and potentially complex mechanism to define a trusted domain of names.

The *name constraints* field allows the CA issuing a cross-certificate to specify the domain of acceptable names in a certificate chain extending from that cross-certificate. For example, suppose ACME Corp. and EMCA Corp. are to cross-certify. ACME wants to accept all certificates issued by EMCA to its own employees, but not certificates issued by EMCA to anyone outside of EMCA. To constrain the acceptable name space, ACME could issue a cross-certificate for EMCA with the *permissible name constraints* field set to "o=EMCA Corp., c=US" (assuming EMCA is a US-based organization). This example is quite simple and only illustrates a small portion of the capabilities of name constraints.

There is also a mechanism whereby a cross-certificate can specify excluded X.500 subtree name domains, although it is expected that this is less useful than the ability to specify acceptable subtree name domains.

Entrust does not currently support the *name constraints* field.

Policy Constraints

The *policy constraints* field is used in cross-certificates. The field provides administrators with the capability to specify the set of acceptable policies in a certificate chain extending from a cross-certificate. The policy constraints field can specify whether or not all certificates in a chain must meet a specific policy and whether or not to inhibit policy mapping when processing a chain.

Entrust does not currently support the *policy constraints* field.