

NATIONAL SECURITY AGENCY
CENTRAL SECURITY SERVICE



R22

INFOSEC Engineering

TOKENEER



**User Authentication Techniques
Using Public Key Certificates
Part 1: Certificate Options**

24 December 1997

Lawrence A. Reinert
Stephen C. Luther

User Authentication Techniques using Public Key Certificates
Part I: Certificate Options

TABLE OF CONTENTS

1.0 Scope	1
2.0 Background	1
2.1 Advances in Authentication.	1
2.2 Introduction to Public Key Infrastructure (PKI).	2
2.3 Introduction to Authentication Information (AI).	3
2.3.1 An Introduction to Biometrics	4
2.4 Introduction to Security Policies	4
2.5 The Importance of Risk Analysis	5
2.6 Terminology	5
2.7 Symbols and Abbreviations.	7
3.0 Reference Documents	8
4.0 Examining Identification and User Authentication	9
4.1 User Identities	9
4.1.1 Authenticating a User's Identity with Public Keys	10
4.1.2 Why Place Authentication Information in a Certificate?	10
4.2 The Effect of System Architectures on I&A	11
4.2.1 Registration of Users	11
4.2.2 Services at System Access Points.	13
4.2.3 Examples of System Architectures.	14
4.2.3.1 Trusted Third Party Architectures	14
4.2.3.2 TOKENEER Architecture	15
5.0 Examining Certificate Standards	17
5.1 The X.509 Certificate	17
5.1.1 The Relative Distinguished Name (RDN)	19
5.1.2 The DistinguishedName (DN)	20
5.1.3 Attributes	21
5.1.4 Encrypted Attributes.	22
5.2 X.509 Version 3 Extensions	23
5.2.1 Standard Extensions	23
5.2.1.1 Key and Policy Information Extensions	23
5.2.1.2 Certification Path Constraints Extensions	24
5.2.1.3 Certificate Subject and Certificate Issuer Attributes Extensions	24
5.2.1.4 CRL Extensions	24
5.3 PKCS #6: Extended Certificate Syntax Standard	25
5.4 Attribute Certificates	25

TABLE OF CONTENTS

6.0	Considering System Characteristics	28
6.1	Attribute Identification	28
6.2	Attribute Confidentiality	28
6.3	Need To Know	29
6.4	Certificate Storage	29
6.5	I/O Throughput	29
6.6	Certificate Processing Time	29
6.7	Certificate Architecture	29
6.8	Certificate Management Complexity	30
7.0	Exploring the Alternatives for Authentication Information	31
7.1	Using X.509 Version 3 Extensions	31
7.1.1	Advantages	32
7.1.2	Disadvantages	32
7.2	Using PKCS#6 Extended Certificates	32
7.2.1	Advantages	33
7.2.2	Disadvantages	33
7.3	Using Attribute Certificates	34
7.3.1	Advantages	34
7.3.2	Disadvantages	34
8.0	Conclusions	35
8.1	Seven Steps to Analyze the Certificate Authentication Information	35
8.2	Mapping the System to the Right Solution	36
8.3	The Bottom Line	38
	APPENDIX I Orange Book Requirements	39
	APPENDIX II X.509 ASN.1 Syntax	42
	APPENDIX III X509 Attribute Certificate ASN.1 Syntax	44

1.0 Scope

The intent of this three part study is to investigate where user authentication information can be placed in public key certificates. This document (Part 1) examines the options which exist within X.509 (and related standards) for the placement of user authentication information. Part 2 [17] investigates the specific authentication information, such as biometrics and passwords, which should be considered in the development of a system. Part 3 [18] presents a case study of how to architect such a system through an example implementation.

The intent of Part 1 of this study is to explore the options available for placing authentication information in X.509 and related standards. An overview of several technologies will be explored in the process. It is not the intent of this document to suggest a preferred approach or implementation.

2.0 Background

User authentication is merely the verification of a claimed identity. There are many ways for an individual to claim his identity, and often it is through the presentation of a trusted item, such as a notarized birth certificate, a social security card, or a state driver's license. These are all examples of tokens.

Tokens are used for the purpose of identification, whether it be identification of a credit card account or of a privilege to use a system. Possession of a token claiming an identity is not meaningful unless a method exists to verify the validity of the information contained on the token as well as the individual. In the typical merchant transaction, a credit card "token" is swiped or the identifying account information is manually entered into a system to verify that the account is a valid account. The merchant then verifies the individual possessing the card by comparing the signature on the receipt to the one on the card.

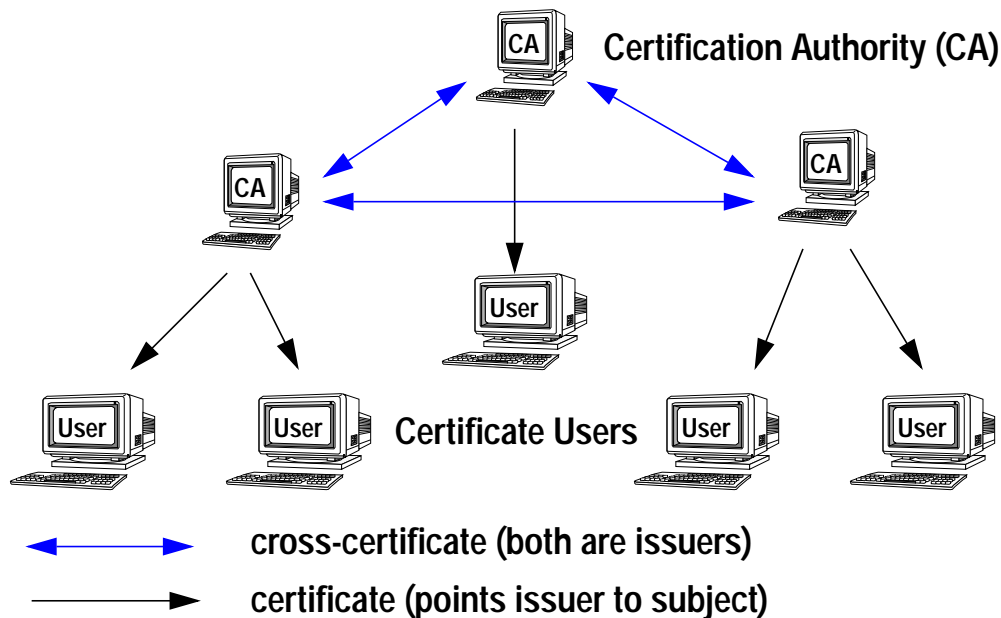
2.1 Advances in Authentication

New advances in biometric identification and token based public key cryptography have prepared a path for fulfilling a number of computer security requirements in a manner more robust than the simple password. Using the basic formula of the example above, with more robust components (public key cryptography versus an account number, and biometric matching versus a visual comparison of signatures), a capability to reduce the threat of unauthorized access to computing resources is on the horizon.

In the credit card example above, there is mention of a "system" which verifies the account. A similar system must be in place for public key cryptography as well as for the biometrics for these technologies to be utilized. It is the definition of this public key and biometric "system" or

subordinate to a common CA. This topology seems to be a better fit for the competitive marketplace.

Network Infrastructure



The structure of the CAs is important in that any format chosen for integrating Authentication Information (AI) into the public key certificates must either be adopted or ignored by all CAs which a user wishes to interoperate with.

2.3 Introduction to Authentication Information (AI)

Users' identities are verified using one or more of three generic methods: something they know (PINs, passwords, memory phrases, etc.) something they have (a physical token such as a magnetic stripe card, a physical key, a smart card, etc.), or something they are (i.e. biometric verification).

ECMA.219 [5] categorizes authentication information (AI) according to the following principles:

- Principle A - something known (examples: passwords, encrypted passwords, encrypted replay protected passwords, hashed passwords, and hashed replay protected passwords),
- Principle B - something possessed (examples: passive token, active token)
- Principle C - immutable characteristic (examples: voice print, signature, fingerprint, retina pattern)
- Principle D - trusted third party (examples: user certificate, zero knowledge method)

- Principle E - context (example: location)

The AI can be thought of as two separate pieces of information: the validated authentication information (VAI) and the request authentication information (RAI). The RAI is what the requesting entity (usually a user) presents to request authentication. The VAI is what the request is compared against. In some cases (such as with passwords) the RAI is the same as the VAI. In other cases (such as with encrypted passwords, hashed passwords, or fingerprints), the RAI must be processed prior to the verification of the RAI. Refer to Part 2 of this study for further details on AI.

AI is typically initialized during the user enrollment process. The entity which performs the enrollment must be a trusted authority which can validate the information in accordance with the system's security policy. Entities which utilize this information must verify that the information has been validated by the trusted authority and that the information has not been altered by any other source. Digital signatures can be used to provide such functionality. By placing the AI in the public key certificate (which is signed by the trusted authority) we can provide such features to the system.

2.3.1 An Introduction to Biometrics

Biometrics utilize authentication and identification technologies based upon unique biological characteristics. Inherent biological traits include voice, fingerprints, hand geometry, facial features, retinal patterns, etc. The biometrics industry is still in its infancy. Each vendor has its own system containing proprietary algorithms, templates, and hardware.

A biometric template is a data set representing the biometric measurement of an enrollee which is maintained on file and used by a biometric verification device for comparison against subsequently submitted biometric samples. The template cannot usually be used to recreate the original print; but instead the live scan is fed through the algorithm to create another template which can then be compared to the original template. Since templates and algorithms have not been standardized, any biometric information which is stored in a public key certificate should also include sufficient information to identify the method required to use the template. This may include manufacturer's name, version numbers, and/or algorithm names/versions.

Refer to Part 2 of this study for further discussion on Biometrics.

2.4 Introduction to Security Policies

One of the more important aspects of the design of a system is the drafting of a security policy. The security policy specifies who is to have access to a system, and what those accesses are to be. The security policy will often specify how these accesses are to be enforced. These requirements will almost always influence the design of the system.

The Department of Defense Trusted Computer System Evaluation Criteria (“The Orange Book”) has a requirement for a Security Policy. Refer to appendix I for details about the Orange book requirements. The security policy requirement is stated as follows:

Requirement 1 - SECURITY POLICY - There must be an explicit and well-defined security policy enforced by the system. Given identified subjects and objects, there must be a set of rules that are used by the system to determine whether a given subject can be permitted to gain access to a specific object. Computer systems of interest must enforce a mandatory security policy that can effectively implement access rules for handling sensitive (e.g., classified) information. These rules include requirements such as: No person lacking proper personnel security clearance shall obtain access to classified information. In addition, discretionary security controls are required to ensure that only selected users or groups of users may obtain access to data (e.g., based on a need-to-know).

This study will discuss the techniques which are useful for fulfilling the requirements of a typical security policy. It will not discuss specific security policies in detail.

2.5 The Importance of Risk Analysis

The system which utilizes the authentication information (AI) must formulate a procedure for capturing the AI and using the AI based upon a variety of factors including the security policy of the system, the system architecture, the protection level required, the value of the information being protected, the potential adversary, and the vulnerabilities of the system. Typically a risk analysis must be performed to analyze the factors involved with the use of AI. Refer to “An Introduction to the Risk Analysis Model (RAM)” (reference [14] of section 3.0) for further information on risk analysis.

A risk analysis has been performed on the system described in the case study found in part 3 of the study; however, it will not be included in this study.

2.6 Terminology

Several terms used throughout this document must be clarified before proceeding:

Authenticate: Establish or prove as: a. Conforming to a fact and therefore worthy of trust, reliance, or belief; b. Having an undisputed origin; c. validating a claimed identity.

Biometric: A measurable, unique physical characteristic or personal trait used to recognize the identity, or verify the claimed identity, of an enrollee.

Certify: To confirm formally as true, accurate, or genuine.

Certificate: A document testifying to accuracy or truth.

False Acceptance: When a biometric transaction results in the acceptance of an imposter (also known as a False Match or Type II Error).

False Rejection: When a biometric transaction results in the failure to recognize the identity, or verify the claimed identity, of an enrollee (Also known as a False Non Match or Type I Error).

False Acceptance Rate: The probability (expressed as a percentage) that a biometric verification device will fail to reject an imposter. It is also known as a Type II Error Rate and is calculated as $FAR = NFA/NIRA$ (or $NIVA$) $\times 100$ where FAR is the false acceptance rate, NFA is the number of false acceptances, NIRA is the number of imposter recognition attempts, and NIVA is the number of imposter verification attempts.

Identify: 1. To establish that the collective aspects of the characteristics by which a thing is distinctly recognizable or known. 2. To consider similar or identical: EQUATE.

Privilege: A special grant, immunity, right or benefit granted to an individual, class or caste.

Template: A data set representing the biometric measurement of an enrollee which is maintained on file and used by a biometric verification device for comparison against subsequently submitted biometric samples.

Token: A possession which shows the identity of its owner, such as an Identity Badge.

Verify: To prove the truth of by presenting evidence or testimony: SUBSTANTIATE.

Verification: The process of comparing a submitted biometric sample against the template of the single enrollee whose identity is being claimed in order to determine whether it matches the enrollee's template or not.

Verification Attempt: The submission of a biometric sample, accompanied by a claimed enrollee identity, to a biometric verification device for a verification decision.

2.7 Symbols and Abbreviations

This section contains symbols and abbreviations used in this document.

Table 1: Symbols and Abbreviations

Abbreviation	Meaning
AA	Attribute Authority
ABA	American Bankers Association
ASN.1	Abstract Syntax Notation
AI	Authentication Information
CA	Certification Authority
CRL	Certification Revocation List
DN	Distinguished Name
ECMA	European Computer Manufacturers Association
IDACert	Identification and Authentication Certificate
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
ITU	International Telecommunications Union
I&A	Identification and Authentication
I/O	Input / Output
PKCS	Public Key Cryptography Standard
PKI	Public Key Infrastructure
PrivCert	Privilege Certificate
RA	Registration Authority
RAI	Request Authentication Information
RDN	Relative Distinguished Name
RSA	Rivest Shamir Adelman algorithm
VAI	Validated Authentication Information
WD	Working Draft

3.0 Reference Documents

- [1] ABA. X9.55, "Public Key Cryptography For the Financial Service Industry: Extensions to Public Key Certificates and Certificate Revocation List."
- [2] ABA. X9.57-199x, "Public Key Cryptography For the Financial Service Industry: Certificate Management." Working draft, June 21, 1996.
- [3] Anderson, R. and M. Roe. "The GCHQ Protocol and its Problems." Cambridge University Computer Laboratory, <ftp://ftp.cl.cam.ac.uk/users/rja14/euroclipper.ps.Z>, undated.
- [4] DOD. DOD 5200.28-STD, "Department of Defence Trusted Computer System Evaluation Criteria ('The Orange Book')." December 1985.
- [5] ECMA. ECMA-219, "Authentication and Privilege Attribute Security Application with related key destruction functions." second edition, 1996.
- [6] ISO/IEC. ISO/IEC 8825, "Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)." first edition, 1995-10-15.
- [7] ISO/IEC. ISO/IEC 9594-2, "Information technology - Open Systems Interconnection - The Directory: Models." 1995.
- [8] ISO/IEC. ISO/IEC 9594-6, "Information technology - Open Systems Interconnection - The Directory: Selected attribute types." 1995-09-15.
- [9] ISO/IEC. ISO/IEC 9594-6, "Information technology - Open Systems Interconnection - The Directory: Selected attribute types AMENDMENT 2: Certificate extensions." draft edition, 1996-03-09.
- [10] ISO/IEC. ISO/IEC 9594-8, "Information technology - Open Systems Interconnection - The Directory: Authentication framework." 06/97.
- [11] Laurie, B. "A Supplementary Analysis of the Royal Holloway TTP-Based Key Escrow Scheme." <http://www.algroup.co.uk/crypto/rh.html>, Nov 16, 1996.
- [12] NIST. FIPS Pub 190, "Guideline for the use of Advanced Authentication Technology Alternatives." September 28, 1994.
- [13] NIST. FIPS Pub 196, "Entity Authentication Using Public Key Cryptography." February 18, 1997.
- [14] NSA R223. "O'er the RAMparts We Watch: An Introduction to the Risk Analysis Model (RAM)." 21 February 1996.
- [15] NSA R223. "TOKENNEER Operational Concept Description." Build 1: version 1.0, November 25, 1996.
- [16] NSA R223. "TOKENNEER Biometric Scanner and Verification Algorithm Study." Version 0.1, November 20, 1996.
- [17] Reinert, L. and S. Luther. "User Authentication Techniques Using Public Key Certificates Part2: Authentication Information Including Biometrics." 31 December 1998.
- [18] Reinert, L. and S. Luther. "User Authentication Techniques Using Public Key Certificates Part3: An Example Implementation." Feb 27, 1997.
- [19] RSA Laboratories. "PKCS#6:Extended Certificate Syntax Standard." Version 1.5, November 1, 1993.
- [20] RSA Laboratories. "PKCS#9:Selected Attribute Types." Version 1.1, November 1, 1993.

4.0 Examining Identification and User Authentication

Identification and Authentication (I&A) techniques are features included in a system only if there is a requirement to do so. In the credit card example in section 2.0, there is a requirement to identify the customer by charging the sale to the correct account so that:

- he can pay for his purchases,
- other customers are not contesting their bills,
- the company can make a profit.

There is also a requirement to authenticate the customer so that:

- the customer doesn't contest his bill,
- consumers have confidence in the system,
- fraud is minimized.

A more complex set of requirements is necessary for improving computer security. The Department of Defense Trusted Computer System Evaluation Criteria has a requirement for subject (i.e. user) identification. Refer to appendix I for details about these requirements. The identification requirement is stated as follows:

Requirement 3 - IDENTIFICATION - Individual subjects must be identified. Each access to information must be mediated based on who is accessing the information and what classes of information they are authorized to deal with. This identification and authorization information must be securely maintained by the computer system and be associated with every active element that performs some security-relevant action in the system.

4.1 User Identities

Users' identities are verified using one or more of three generic methods: something they know (PINs, passwords, memory phrases, etc.) something they have (a physical token such as a magnetic stripe card, a physical key, a smart card, etc.), or something they are (i.e. biometric verification).

While most current security policy requirements can be met by any one of these different methods, systems using two or more methods will have greater security. Systems using only one method of authentication may be more vulnerable to compromise of the authenticator, thus multiple methods are preferred.

New advances in biometric identification and token-based public key cryptography have opened the door for improving existing computer security requirements. The main improvement comes by verifying a claimed identity based upon some physical characteristic, which can be sampled at any specific time and verified against a tamper-proof known good sample (preferably a sample

that is electronically signed by a trusted source). When the biometric verification (the “something you are”) is combined with memory phrases (the “What you know”) and/or physical possession of a verifiable token (the “What you have”) the authentication of a claimed individual becomes extremely difficult to spoof.

4.1.1 Authenticating a User’s Identity with Public Keys

A public key certificate, such as that specified in X.509, provides a unique “claimed” identity for the individual person to whom the key was issued. Stealing the public key (contained in the certificate) is useless to an adversary trying to masquerade as the individual, since the adversary does not possess the secret component of the key. The binding between the claimed identity of the public key and the individual is “linked” by simple possession of the secret portion of the key.

A user wishing to verify his “claimed” identity will accept a known value, encrypt the value with the private key of the “claimed” identity, and then return the encrypted value along with the public certificate of the “claimed” identity (or sufficient information for the system to correctly retrieve the public certificate) to the system performing the verification. The verification system must then decrypt the encrypted value using the public key in the certificate and obtain the known value. If the decryption reveals the known value, the user is verified as the “claimed” identity. If the known value is not retrieved, the user is verified as being someone other than the “claimed” identity.

Typical systems will allow the individual user access to the secret component of the key by verifying a password. The password is used to “authenticate” the individual user. Clearly the user authentication can be greatly enhanced by implementing more advanced techniques to provide access control to the private component of the key.

4.1.2 Why Place Authentication Information in a Certificate?

Reasons for placing authentication information (AI) in an X.509 certificate include:

1. **Universal Acceptance:** Many systems are being developed to interact with the X.509 model. The dominance and acceptance of the X.509 certificates by the Internet community, in the area of public keys, makes it a prime candidate for holding authentication information.
2. **No Connectivity Constraints:** Placing AI in an X.509 certificate would provide everything that a local terminal would need to authenticate an individual. If reliable, continuous connectivity does not exist with a trusted authentication server, then the X.509 certificate is the next best mechanism. Refer to section 4.2.3.1 “Trusted Third Party Architectures” on page 14 for further discussion.
3. **Existing Infrastructure:** The trusted components of the infrastructure required to create, distribute, and manage the certificates are in place and would not have to be significantly altered

to provide the extra information.

4. **Simpler Implementation:** Using an existing standard and infrastructure to add authentication information is clearly the easiest means of implementing an enhanced security feature.

5. **Data Integrity:** The PKI hierarchy and use of digital signatures defined in X.509 provides beneficial data integrity for the AI. This applies even to data that is not exchanged (e.g. a database). Data can be authenticated and checked for external manipulation if the certificate is validated prior to use.

Reasons against placing Authentication information in an X.509 certificate include:

1. **Information Compromise:** System vulnerabilities and/or a weak security policy could compromise the confidentiality of the AI.

2. **Increased Exposure:** Greater exposure of AI could require further improvements to biometric false acceptance rates or the use of stronger encryption algorithms since this information literally contains the “keys to the kingdom”.

3. **Certificate Size:** AI could increase the size of a standard certificate to ranges deemed unacceptable by the Internet community. The AI data added to the certificate may increase the certificate processing time.

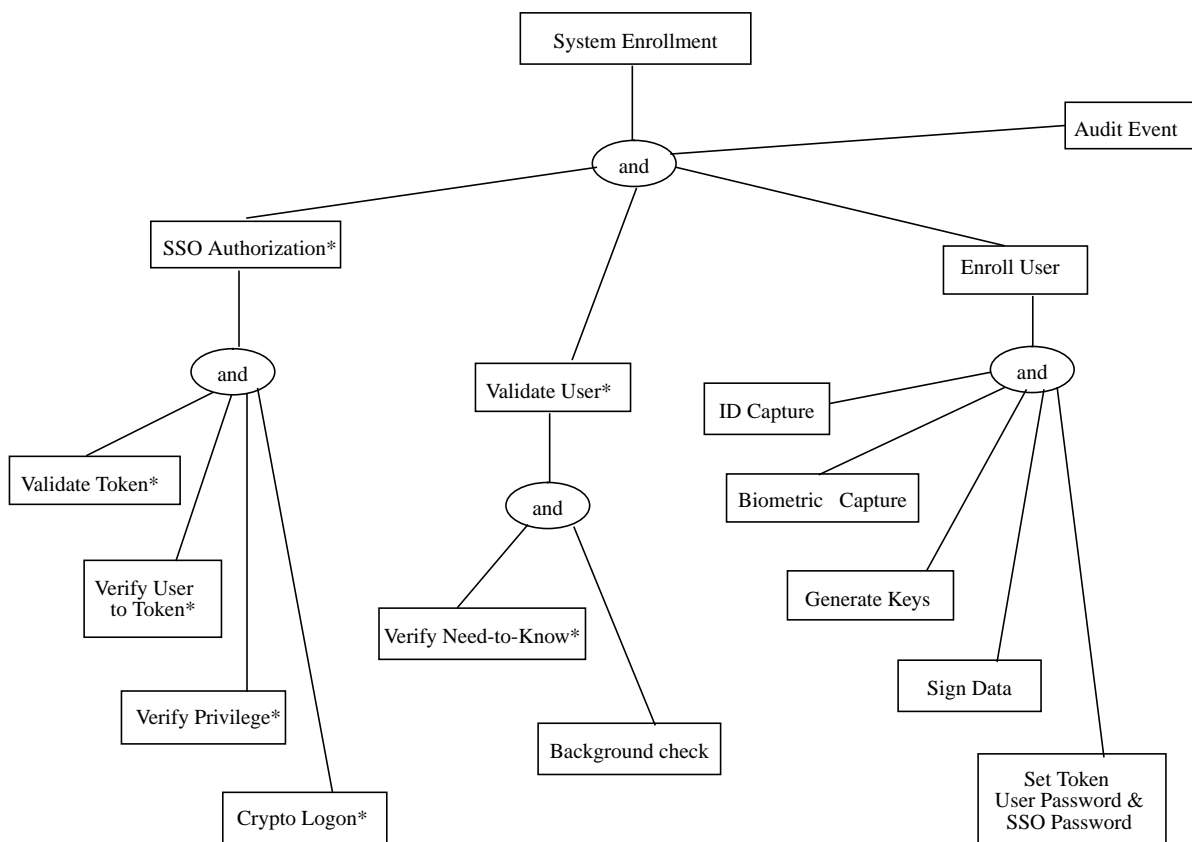
4.2 The Effect of System Architectures on I&A

Any system which includes the functions of I&A must perform two high level tasks: registration of users into the system and subsequent user services at system access points. Each will be examined in turn followed by the description of several alternate architectures for the provision of user services and the effect each has on how I&A is provided.

4.2.1 Registration of Users

In order for any system to correctly provide I&A services, it must contain a trusted method to enroll its users. This usually entails the user providing evidence of who he is (e.g. driver’s license, birth certificate). The registration process will usually conclude by providing something to the user to gain access. This something is usually a badge, a password, or a combination of the two. A sample enrollment process is shown below. The station that the user provides his information to may be just a registration system or it could assume all of the functions of a Certificate Authority and/or an Attribute Authority.

Example System Enrollment Process



Note*: This Branch must be completed successfully prior to the second branch

Once the user is registered, there must be a path from the system used for registration to the access points of the system so that services or resources can be provided to the user. In many current systems, the registration system is physically connected (often through a network) to the access points. This can introduce vulnerabilities. Through the use of public key enabled tokens, there is the capability to isolate the registration system from the access points while still providing all of the information necessary to those points for making identification and authentication decisions. The access points also have complete confidence that the information was provided by the trusted registration system. The token becomes the carrier of all information necessary (except perhaps for information regarding revoked privileges, which would need to use another channel) for the access point to make a determination on granting of privilege. This information is typically packaged into a digitally-signed structured list commonly referred to as a certificate.

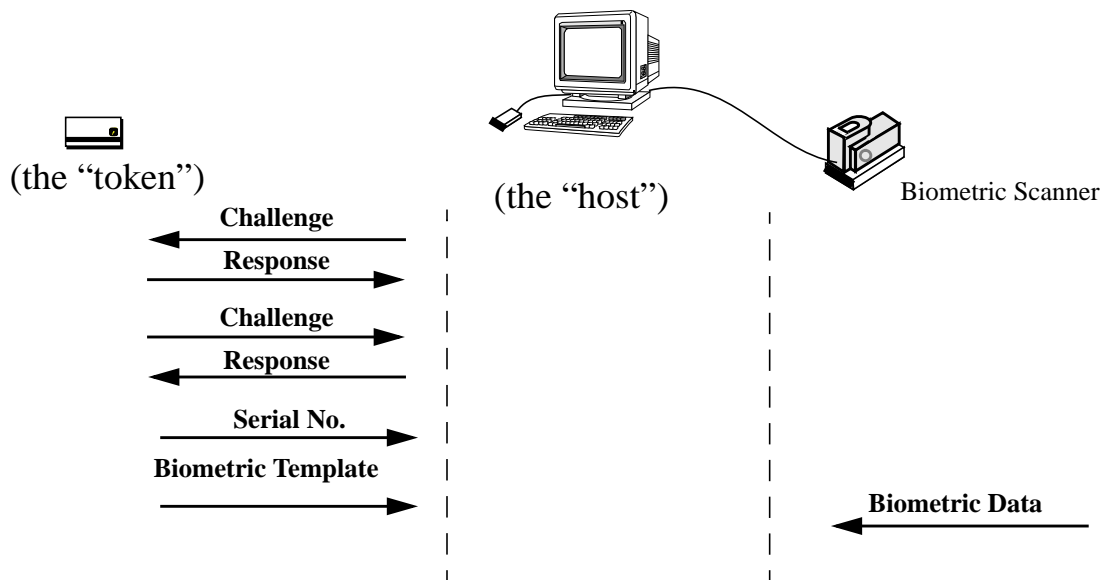
4.2.2 Services at System Access Points

The system access point must make a determination of whether to grant or withhold privilege from the requester. The first step in this process is to determine who is claiming to request service. This may be accomplished in many ways. The most traditional method is to enter a user id at a login prompt. Another method is to present a unique token which provides an identity. Yet another is to present a biometric to be compared against a database of known users.

After the claim of identification, proof must be presented that the requestor is who he claims to be. This may be done by entering a password or personal identification number. It may also be done by presenting a biometric to be matched against a template known by the system to belong to the identity claimed.

In the case of a token based system using biometrics, the token can provide the connection between registration of the user and all subsequent access decisions. The certificate will contain the user's claimed identity and all the templates which have been registered for biometric verification. The system will then prompt the user to prove that he is the individual who was registered. This may be done by entering the password to the token, providing a fingerprint or other biometric for comparison, or a combination of the two. Access areas requiring higher identification confidence will use the more stringent requirements, effectively trading ease of use for security. The example shown below utilizes a smartcard and a biometric, but no passphrase.

The Local User Verification



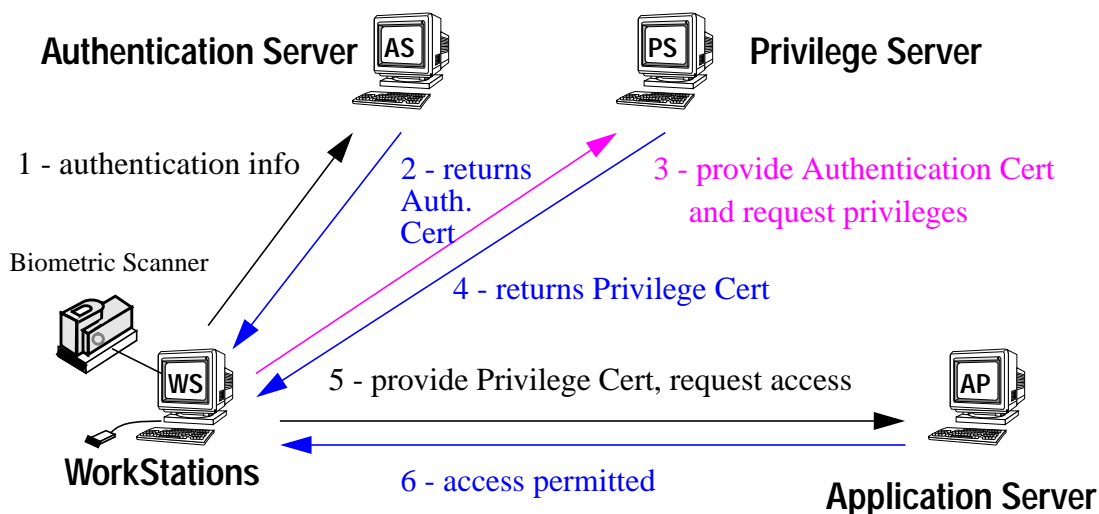
4.2.3 Examples of System Architectures

The following is an introduction to two systems which use authentication information and how they use the information. Each authentication function can be performed by different components of the system. By distributing the functions to different components of the system, the sensitive functions can be handled by “trusted entities” or shared by several components in order to reduce the risk of attack on the system.

4.2.3.1 Trusted Third Party Architectures

A trusted third party architecture assumes that there is at least one entity in a system which can be trusted to provide authentication. The trusted entity, which is called an authentication server (AS), is responsible for holding the validated authentication information (VAI) of each individuals enrolled into the system. All users and services can authenticate each other by sending the request authentication information (RAI) to the AS. The authentication server compares the RAI to the VAI to verify the requestor. That same server, or another called a privilege server, is trusted to provide the user’s privileges to the application server, which runs the service that is requested by the user. As can be seen below, this architecture can provide an extremely distributed system.

Trusted Third Party Architecture



Examples of Trusted Third party efforts are:

Kerberos: A trusted third party service which uses user passwords and symmetric key encryption to provide authentication. If the authorization is approved, the server issues a

ticket to the requester which can be used by the service being requested to authenticate the requester.

Secure European System for Application in a Multi-vendor Environment (SESAME): A trusted third party system based upon Kerberos and European Computer Manufacturers Association (ECMA) standards which added asymmetric cryptography and other improvements over Kerberos.

The Royal Holloway TTP-Based Key Escrow Scheme: A trusted third party system for electronic mail based upon Diffie-Hellman and the Message Security Protocol. This system is sponsored by the UK government.

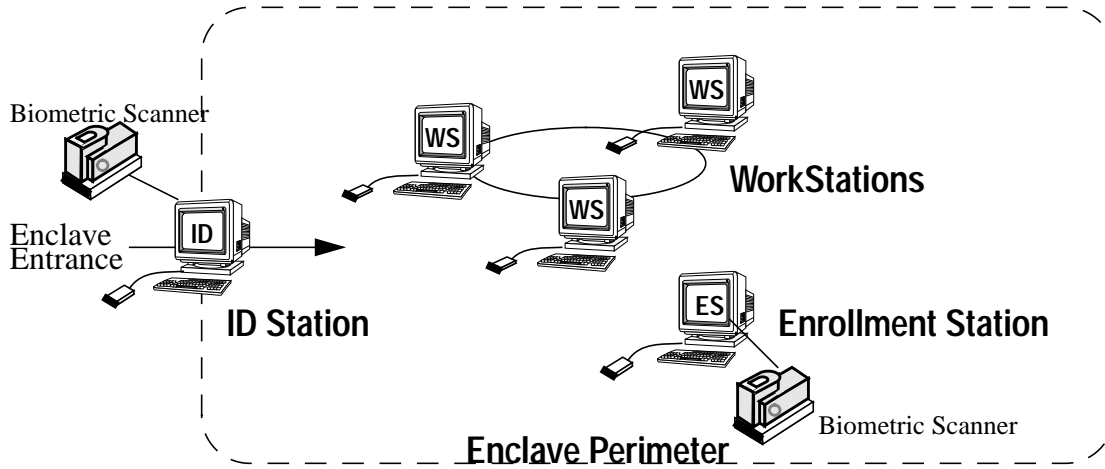
In the trusted third party example, the VAI may not necessarily need to be placed in the public key certificate since the authentication server holds the validation information and is accessed every time a request for authentication is made. This assumes that all components of the system have connectivity to and immediate access to the authentication server. It also assumes that the workstation is somewhat trusted to gather the RAI and send it securely to the authentication server without risk of alteration by an adversary.

4.2.3.2 TOKENEER Architecture

The TOKENEER system uses a nested approach to authentication based upon the premise that immediate connectivity to a trusted authentication server (AS) is not always possible. It utilizes a token to provide validated authentication information (VAI) to service access points. It also assumes that the workstation is not trusted to perform the complete authentication of the user. The I&A function of the TOKENEER architecture is divided between the ID station and the workstations to achieve three factor authentication. The biometric and token factors are used at the ID station (which is placed at the enclave entrance), while the token and a password are used

at the workstation.

TOKENEER Architecture



The Tokeneer System will be used as the example system in Part 3 of this study. Refer to Part 3 for further details on the Tokeneer system, and how the AI is utilized.

5.0 Examining Certificate Standards

The implementation of any system should not be conducted without first examining the availability of existing products and/or standards for its use. In the area of public key certificates, both the standards community and industry have focused their energies upon the X.509 certificate (as defined in ISO/IEC 9594-8), with several competing provisions for extensions (or additions) including the extensions and attribute certificates defined in version 3 of the X.509 standard, (equivalent to ANSI X9.57), and extended certificates defined in RSA Data Security Inc.'s PKCS#6. A brief description of the X.509 standard and some distinguishing features of each proposed extension to the standard follow.

5.1 The X.509 Certificate

The X.509 standard is an international standard for security and authentication services supporting security frameworks for electronic information distribution. The term "X.509 certificate" has become the defacto name for public key certificates in use today. This specification defines the main data structure (i.e. the "certificate") used for performing these services and addressing the handling of keys.

The X.509 certificate is primarily used to hold public key information. In addition to public key information, X.509 also describes certificate revocations lists (CRLs) which are signed lists of certificate serial numbers which have been compromised. A large portion of the system which utilize public key certificates may be focused on the managing of the certificates via CRLs. The creation, distribution, and utilization of CRLs can become quite complex. Systems which utilize multiple certificates can greatly increase the complexity of the system if different CRLs are used to control each type of certificate. When designing the architecture of a system, the management burden to support the system must be considered. The focus of this study, however, will be on how to utilize certificates to provide authentication techniques, not CRL management.

X.509 is a recommendation by the International Telecommunications Union (ITU). X.509 is equivalent to the International Standards Organization (ISO) / International Electrotechnical Commission (IEC) 9594-8 specification. The X.509 specification and ISO/IEO 9594-8 document are identical. X9.55, originated by the American Bankers Association (ABA) and adopted by ANSI, contains an equivalent description from the perspective of the banking industry. ISO/WD-15782 is the international version of the banking industry's certificate management descriptions, which borrows from X9.57 and X9.55.

Information Contained in the X.509 Version 3 Certificate is as follows:

Table 2: X.509 Version 3 Fields

Field	Description
version	This identifies which version of the X.509 standard applies to this certificate. This affects what information can be specified in it. Version 1 has been available since 1988, and is widely deployed. (Note: a value of 0 indicates Version 1, 1 indicates Version 2, and 2 indicates Version 3)
serial Number	The issuer creates a unique serial number and places it here. It is intended to distinguish this certificate from all others created by the issuer.
subject	The name of the entity (usually a human) whose public key the certificate identifies. This name uses the X.500 standard's Distinguished Name (DN), and is intended to be unique. The DN is comprised of such information as the country, region, and given name of the entity.
subject Unique Identifier	The unique id of the subject (optional for version 3 and after). Added in version 2, this field provides a location to specify a bit string to uniquely identify the entity's X.500 DN, in the event that the same DN has been assigned to more than one entity over time.
subjectPublicKeyInfo	The Public Key field consists of: The ID of the Algorithm the public key is to be used with, and the public component of the key which belongs to the entity specified by the Subject Name. This value is used to encrypt information to be sent to the entity.
issuer	The name of the entity that is signing the certificate. The entity is usually a Certificate Authority (CA). This name uses the X.500 standard (the Distinguished Name), and is intended to be unique. The CA can sign its own certificate.
issuer Unique Identifier	The unique id of the issuer (optional) Added in version 2, this field provides a location to specify a bit string to uniquely identify the Issuer X.500 DN, in the event that the same DN has been assigned to more than one CA over time.

Table 2: X.509 Version 3 Fields

Field	Description
validity	This specifies when a certificate is valid. This period is described by a start date and time and an end date and time as follows: notBefore: The start time that the Certificate is valid. notAfter: The end time that the Certificate is valid.
extension(s)	Additional fields (Optional-see following section)
signature	The Signature field consists of: Identifier of the Algorithm used to create the signature and the output of the Signing function (i.e. the signed Hash value of the data in this certificate). This data is used to verify the data in the certificate.

5.1.1 The Relative Distinguished Name (RDN)

The subject field in the X.509 certificate refers to a “name” whose definition is imported from X.501 (X.501 is equivalent to ISO/IEC 9594-2). The name in X.501 consists of sequence of relativeDistinguishedNames. Each RelativeDistinguishedName is a set of 1 or more attribute types and values.

Name::=RDNSequence

Where RDNSequence is defined as:

RDNSequence::=SEQUENCE OF RelativeDistinguishedName

And RelativeDistinguishedName is defined as:

RelativeDistinguishedName::= AttributeTypeAndValue

Where the Attribute Type and Value is specified in X.520 (X.520 is equivalent to ISO/IEC 9594-6). Only one attribute is required to be used. Multiple uses of the same attribute can also be specified (i.e. there can be 3 organizationalUnitNames). It is left to the individual system to dictate which attributes will be used and how.

5.1.2 The DistinguishedName (DN)

The DistinguishedName deserves some discussion because it is intended to create a unique identifier for the entity which owns the certificate. It is created from the sequence of the Relative Distinguished Names (RDN) of the entity and each of its superior entities. Each RDN only contains enough information to distinguish it from its superior. The DN (being a combination of all its superior RDNs) creates a unique name (assuming that its superior named it uniquely) and illustrates its hierarchy. The following diagram (taken from X.501) illustrates the naming of RDNs and DNs.

Determination of Distinguished Names

	RDN	Distinguished Name
Root 		{ }
	C=GB	{C=GB}
	O=Telecom	{C=GB, O=Telecom}
	{OU=Sales, L=lpswitch}	{C=GB, O=Telecom, (OU=Sales, L=lpswitch)}
	{CN=Smith}	{C=GB, O=Telecom, (OU=Sales, L=lpswitch), CN=Smith}

5.1.3 Attributes

The following is a discussion on Attributes that can be used in X.509 certificates. Attributes are defined in X.501 (reference [7]) as "Information of a particular type." The Attribute is meant to describe a characteristic of the object to which it is associated with. Refer to X.501, reference [7] for further information about Attributes.

The AttributeTypeAndValue referenced by the RelativeDistinguishedName (RDN) is defined as follows:

```
AttributeTypeandValue ::= SEQUENCE {
    type ATTRIBUTE.&id ({SupportedAttributes});
    value ({ATTRIBUTE.&Type({SupportedAttributes})}@type)}
```

Where the ATTRIBUTE.&id is the object identifier assigned to it and the ATTRIBUTE.&TYPE is the it attribute syntax (An ASN.1 type such as BIT STRING, INTEGER, etc.). Both the id and Type are defined in a class named ATTRIBUTE.

The Attribute construct utilized by X.509 related constructs is defined as follows:

```
Attribute ::= SEQUENCE {
    type ATTRIBUTE.&id ({ SupportedAttributes}),
    values SET SIZE (0.. MAX) OF ATTRIBUTE.&TYPE ({ SupportedAt-
        tributes}@type}),
    valuesWithContext SET SIZE (1 .. MAX) OF SEQUENCE {
        value ATTRIBUTE.&Type ({SupportedAttributes}){@type})
        OPTIONAL,
    contextList SET SIZE (1 .. MAX) OF Context} OPTIONAL}
```

Contexts are properties of the attribute which are used to determine the applicability of the attribute. As an example, contexts can be used to associate a particular language, time, or locale.

The Context is defined by X.501 as follows:

```
Context ::= SEQUENCE {
    contextType CONTEXT.&id ({SupportedContexts}),
    contextValues SET SIZE (1..MAX) OF CONTEXT.&Type ({SupportedCon-
    texts}@contextType}),
    fallback BOOLEAN DEFAULT FALSE}
```

All Attributes must be standardized (i.e. registered) with an ISO recognized standards body. Some are registered with international organizations, such as the ISO, and other are registered at the national level organizations, such as ANSI. There is no single source or document that lists all registered attributes.

Some Types are listed in X.520 along with a description of the value (BIT STRING, INTEGER, etc.). Even with the description given in X.520 there is still room for interpretation of the fields. There is not always a real standard as to the semantics and exact meaning of each of the elements in the RDNSequence. As an example, some name creating bodies will use the CommonName and others will use a combination of surName and givenName.

X.521 (equivalent to ISO 9594-7) defines several groupings of attributes or “classes.” The X.509 certificate can make use of these classes.

Refer to ITU X.509 or ISO/IEC 9594-8 for further details on the X.509 Certificate definitions or X.501 and X.520 for further definitions of the Distinguished Name.

5.1.4 Encrypted Attributes

There is an option for an encrypted Attribute within Certificates, as specified in X.501. The syntax for the encrypted attribute is as follows:

```
EncryptedAttributeSyntax {AttributeSyntax} ::= SEQUENCE {
    keyInfo          SEQUENCE ::= SEQUENCE OF KeyIdOrProtectedKey
    encAlg           AlgorithmIdentifier,
    encValue         ENCRYPTED {AttributeSyntax}}
```

```
KeyIdOrProtectedKey ::= SEQUENCE {
    keyIdentifier    [0] KeyIdentifier OPTIONAL,
    protectedKeys   [1] ProtectedKeys OPTIONAL}
```

-- At least 1 key identifier or protected key must be present.

Where AttributeSyntax is the cleartext syntax of the attribute.

The KeyIdentifier may be the identifier of a certified public key as held in the Subject Public key Identifier extension field defined in X.509 and ISO/IEC 9594-8 Clause 12 or the identifier of a symmetric key and associated security control information.

The AlgorithmIdentifier is the identifier of the algorithm used to encrypt the attribute.

The information specified for the attribute is encrypted and placed in the encValue field. The Key Identifier tells the recipient on how to decrypt the information. Refer to X.501 for more details on

the Encrypted Attributes.

One limitation for the use of encrypted attributes in certificates is the need for having the key at the time of the Authority signing the certificate. This restricts the use of encrypted attributes because the user of the encrypted attribute cannot re-encrypt the attribute after the certificate is created without invalidating the signature on the certificate. The recipient which utilizes the encrypted attribute must have possession of the key (specified by the KeyIdOrProtectedKey) prior to processing the encrypted attribute.

5.2 X.509 Version 3 Extensions

Version 3 of X.509 introduced the extension mechanism. The mechanism allows anyone to register an extension with the appropriate authorities (e.g., the ITU or ISO). Each extension contains the following information:.

Type	Criticality	Value
------	-------------	-------

Where:

Type: The Type is used as an identifier for extension.

Criticality: This (single bit) flag is used to denote whether the information in this extension must be processed. If this flag is set and the application cannot process the extension type, then the application must reject the certificate. Note: An application can still require that a non-critical extension be present in order to process. This flag is intended to insure that all applications process this extension because it is of great importance. Most extensions will be non-critical.

Value: Data for the extension. This data will be processed in accordance with the description of the extension.

5.2.1 Standard Extensions

X.509 Version 3 defined several extensions when it released the specification. These extensions were categorized under four basic sections. A detailed description of the extensions is found in X.509, reference[10]. The following is an overview of categories of extensions described in X.509.

5.2.1.1 Key and Policy Information Extensions

These certificate and CRL extensions convey additional information about the certificate subject

and certificate issuer keys, such as key identifiers and indicators of intended key usage and restrictions on key usage. They also convey key policy. They facilitate the implementation of public-key infrastructures and allow administrators to limit the purposes for which certificates and certified keys are used. This grouping would not be appropriate for authentication information.

5.2.1.2 Certification Path Constraints Extensions

These extensions allow constraints specification to be included in CA-c certificates (i.e. certificates for CA's issued by other CA's), to facilitate the automated processing of certification paths when multiple certificate policies are involved (e.g. when policies vary for different applications in an environment or when inter operation with external environments occurs). The constraints may restrict the type of certificates which can be issued by the subject CA or which may occur subsequently in a certification paths. This grouping would not be appropriate for authentication information.

5.2.1.3 Certificate Subject and Certificate Issuer Attributes Extensions

These certificate and CRL extensions support alternative names, of various name forms, for certificate subject and certificate or CRL issuer. They can also convey additional attribute information about the certificate subject, to assist a certificate user in being confident that the certificate subject is a particular person or entity.

5.2.1.4 CRL Extensions

The CRL extension allows a CRL to include indications of revocation reasons, to provide for temporary suspension of a certificate, and to include CRL-issue sequence numbers to allow certificate users to detect missing CRLs in the sequence from one CA.

The CRL distribution points and delta CRLs allow the complete set of revocation information from one CA to be partitioned into separate CRLs, and to support the use of partial CRLs indicating only changes since the preceding CRL issue. This grouping would not be appropriate for authentication information.

5.3 PKCS #6: Extended Certificate Syntax Standard

RSA Data Security Inc. has introduced the “extended” certificate which envelopes the X.509 certificate. The extension allows an existing X.509 certificate to be embedded into a structure which adds additional information and signs the entire structure. This allows a X.509 certificate to be extracted from the extended certificate for backward compatibility.

The extended Certificate structure is as follows:

Table 3: X.509 Version 3 Fields

Field	Description
version	This identifies the version number of the extended certificate for compatibility with future revisions of the standard. Current version is 0.
certificate	An X.509 Certificate
attributes	A set of attributes which give additional information about the subject of the certificate. Some attribute types are defined in PKCS#9.
signature	The Signature filed consists of: Identifier of the Algorithm used to create the signature and The output of the Signing function (i.e. the signed Hash value of the data in this certificate). This data is used to verify the data in the extended certificate.

The Extended Certificate is discussed in detail in PCKS#6 with some attributes defined in PKCS#9.

5.4 Attribute Certificates

Attribute certificates are used to convey a set of attributes along with a public key certificate identifier (i.e. a serial number and a public key certificate issuer name) or entity name. The attributes were placed in a separate structure to maintain conformance with existing international standards (X.509). An entity may have multiple attribute certificates associated with each of its public keys certificates.

X9.57, originated by the American Bankers Association (ABA) and adopted by ANSI, also defines an attribute certificate which is complementary to the X.509 certificate. ISO/IWD-15782-2 also describes the attribute certificate, with added detail for banking applications.

There is no requirement that the same authority create both the public key certificate and the attribute certificate; in fact, role separation should frequently dictate otherwise. The generation of an attribute certificate may be requested by an entity other than the subject of the attribute certificate. The X9.57 specification does not define the messages between an entity and the attribute authority (AA) dealing with the generation of the attribute certificate.

X9.57 defines an attribute as information, excluding the public key, which is provided by an entity or an AA, and certified by the AA in an attribute certificate. Attributes are bound to a public key certificate or entity name by the signature of the AA on the attribute certificate.

The Information Contained in the Attribute Certificate is as follows:

Table 4: X9.57 Attribute Certificate Fields

Field	Description
version	This identifies the version of the attribute certificate.
serial Number	This field uniquely identifies this certificate among all those issued by the AA. (if the AA is also a CA, the serial number space is thus shared by the public key certificates and the attribute certificates.)
owner	An attribute certificate may be linked to either a particular entity, or one of that entity's public key certificates. The mechanism to be used is specified by the application or standard which uses the attribute certificate.
issuerName	This field contains the name of the issuer of the attribute certificate (an AA). The syntax is defined in appendix III
Issuer Unique Identifier	This field uniquely identifies the issuer, in the case where the issuer name is not sufficient.
validity	This specifies when a certificate is valid. This period is described by a start date and time and an end date and time as follows: notBefore: The start time that the certificate is valid. notAfter: The end time that the certificate is valid.
Attributes	The attributes are information concerning the entity, or the certification process. They may be supplied by either the entity, a third party entity, or the AA depending upon the application.

Table 4: X9.57 Attribute Certificate Fields

Field	Description
extension(s)	The extensions field allows addition of new fields to the attribute certificate without modification of the ASN.1 definition.
signatureAlgorithm	This field identifies the algorithm used to sign the certificate.
signature	The signature field consists of: The output of the Signing function (i.e. the signed hash value of the data in this certificate). This data is used to verify the data in the certificate.

The AttributeCertificate matching rule was created to allow more complex matching than the certificateExactMatch (a matching rule defined in X.509). It allows Comparison to the issuer's serialNumber, the owner, the issuerName, and the validity. Refer to X.509 for further information on the matching rules.

A new attribute type of AttributeCertificate is defined to bind the attribute certificate to an X.509 certificate or directly to a subject name. The AttributeCertificateAttribute can be used with the certificateExactMatch to verify the binding.

6.0 Considering System Characteristics

Determining the best approach for the placement of attributes to be used for authentication requires the careful study of the proposed characteristics and/or requirements of the system to be implemented and the possible impact each requirement may have upon another. A discussion of those requirements which would most directly impact the certificate structure decision follows.

6.1 Attribute Identification

The minimum set of information necessary to achieve the authentication requirements of the system must be identified. Is all of this information going to remain stable for the entire validity period of the certificates? The stable data must be separated from the transient data. Effort must then be placed into mapping the information into existing X.509 attributes and identifying any deficiencies. Identification of attributes which need to be defined could potentially lead to an object identifier registration or even eventual modification of the standard.

6.2 Attribute Confidentiality

The X.509 certificate was designed to be a public resource and is commonly stored in a public depository (such as in a mail server) where anyone can have access to it. Some attributes (such as a clearance level) may be considered sensitive and therefore its placement in a publicly accessible location is not recommended.

Providing confidentiality (i.e. encryption) of the specific attribute may be necessary. The encrypted attribute may be a possible alternative. However an encrypted attribute may not be very useful since it requires a known key at the time the certificate is signed. Refer to the section 5.1.4 "Encrypted Attributes" for information on this subject.

Encryption of the entire certificate may be an alternative. For this reason, splitting the attributes into public and private (confidential) attribute groups may be a useful technique. The confidential attributes can be placed in a separate certificate and encrypted prior to sending to another system component. Choosing this option does require that the solution support multiple certificates (i.e. a public key certificate and an attribute certificate). Performing matching functions on the confidential certificate may be difficult to attain if the information to be matched upon is encrypted.

An alternative to encrypting the information within the directory may be to encrypt at a lower level (i.e. at the transport layer) after a mutual verification between the entities exchanging the information is performed. If the confidential information is kept in a separate certificate, the public key certificate can be used to encrypt the confidential certificate. The information is only protected over the I/O channel between the two entities. For some applications this may be sufficient. Refer to [13] for further information on mutual verification.

6.3 Need To Know

Sensitive attribute information can be further parsed into categories which are based on a “need to know” concept. Only system components which have a need to know should have access to certain information. Certain sensitive attributes, which can be placed in the certificate as an attribute (such as a password), may only be valid for use by one or two system components, and should not be propagated to other system components which have no need of the information.

6.4 Certificate Storage

Each certificate will require some amount of memory to store it. In the case where the certificate is stored in a token (such as a smartcard), storage area may be a critical factor. Each certificate requires at least one signature. The size of the signature depends on the exact signature algorithm being used (128 bytes in the case of a 1024 bit RSA signature). Adding other data fields and ASN.1 encoding overhead, each certificate can be on the order of several hundred bytes. Using a single certificate may be beneficial to those systems in which certain components have limited storage capabilities.

6.5 I/O Throughput

In Systems where I/O throughput is a factor (especially in smartcard based systems where I/O may be limited to 9600 baud/second) the data size of the certificate may be a concern. Separating the certificate into several certificates will be beneficial only if transfer is limited to one certificate per service request. Separating attributes into several different certificates may also be detrimental to overall system performance if multiple certificates are required.

6.6 Certificate Processing Time

The amount of time it takes to perform the signature verification may be critical to the overall system performance. This time may vary within a system depending upon the placement of components. If the processing time to verify a set of signatures is of concern, then it may be beneficial to reduce the number of certificates, and therefore signatures, requiring verification.

6.7 Certificate Architecture

The system components needed to manage the certificates may increase with the number of certificates. If an attribute certificate is used, the system may plan an attribute authority which is distinct from the certificate authority. This may indicate that certain attribute certificates require separate management functions from the certificate. This may imply a separate set of certificate revocation lists for attributes, and a separate overt function to align certificate and attribute certificate status. The number of components needed to support the system depends upon the need

for separation of roles decided upon by the system designers.

6.8 Certificate Management Complexity

The creation, maintenance, and invalidation of certificates should be of great concern to system designers. An increase in the number of certificates issued per individual user implies the management complexity increases greatly, especially where issuance is spread among multiple authorities (as with attribute authorities).

7.0 Exploring the Alternatives for Authentication Information

The definitions given in the X.509 and the three related standards provide some flexibility and freedom in interpretation when placing new authentication information in several different locations. While a new extension could be proposed for adding authentication information, the time and effort necessary to get a new extension approved may be unnecessary if there exists an approved extension which could handle such information.

The following sections will discuss the advantages and disadvantages of using the X.509 version 3 extensions, the PKCS#6 extended certificate, and X.509 attribute certificates.

7.1 Using X.509 Version 3 Extensions

Of the basic categories of extensions that have been defined for version 3 of X.509, only the “Certificate Subject and Certificate Issuer Attributes Extensions” category is appropriate for the insertion of more information about the subject. The requirements for that section include the following statement:

A certificate user may need to securely know certain identifying information about a subject in order to have confidence that the subject is indeed the person or thing intended.

The only field specified in this section which provides for the placement of a generic attribute is the `subjectDirectoryAttributes` field. The `subjectDirectoryAttributes` field, as defined in ITU-T X.509 (also ISO/IEC 9594-8:1995-1) is as follows:

This field conveys any desired Directory attribute values for the subject of the certificate. The following ASN.1 type defines this field:

```
subjectDirectoryAttributes EXTENSION ::= {  
    SYNTAX AttributeSyntax  
    IDENTIFIED BY {id-ce subjectDirectory Attributes}}
```

`AttributeSyntax ::= SEQUENCE SIZE (1..MAX) OF Attribute`

This extension is always non-critical. This implies that an application can ignore this extension if it doesn't have the capability to process it.

If there is a directory attribute defined that could hold the authentication information, then this extension would be a prime candidate for holding the information.

7.1.1 Advantages

The Advantages of using the subjectDirectoryAttribute field would be:

1. Using an already defined extension to implement authentication validation information would not require long approval cycles.
2. Since the extension is non-critical, this information could be ignored by those applications which do not need it.
3. The field utilizes an approved extension to an international standard. Acceptance of this standard is well established.

7.1.2 Disadvantages

The Disadvantages in using the subjectDirectoryAttribute field would be:

1. It is unknown how many application currently using X.509 certificates can handle version 3 certificates.
2. Information in the X.509 certificate is meant to be public. Any attribute placed in this certificate cannot be kept confidential.
3. The information in the certificate is tied to the life span of the certificate itself.

7.2 Using PKCS#6 Extended Certificates

PKCS#6 contains an ExtendedCertificateInfo which has the following ASN.1 syntax:

```
ExtendedCertificateInfo ::= SEQUENCE {  
    version Version  
    certificate Certificate  
    attribute Attributes }
```

The PKCS#6 uses the standards from X.509 and attributes to create an envelope around the certificate. This envelope can hold the useful authentication attributes discussed in this document, without making any changes to the existing standard.

Since subject authentication will only be needed by a few functions (such as logon, large monetary transactions, etc.) this method has a clear advantage of being able to remove the additional (possibly large) amount of data suggested for authentication needs.

7.2.1 Advantages

Using a PKCS#6 extended certificate would have the following advantages:

1. The extended certificate leaves the original X.509 certificate intact and envelops it with additional signed information. This permits the X.509 certificate to be extracted for compatibility with other standards or existing applications. Applications in which the speed is critical, and which do not require the information in the extended certificate, do not have to process the excess information.
2. The X.509 certificate and the extended certificate can be verified with a single public-key operation, since they are signed together by the same certificate issuer.
3. Only the additional information needed would be placed in the extended certificate. Applications which utilize the extended certificate would have access to all the information in the X.509 certificate.
4. Separate CAs could be used for certificate creation and extended certificate creation. This would imply the use of two separate signatures, however, it would allow the user enrollment function to be delegated to other CAs.

7.2.2 Disadvantages

The disadvantages of using a PKCS extended certificate would be:

1. PKCS is not an international standard. The acceptance of an extended certificate may be more difficult to obtain.
2. Two signatures would have to be created by the CA: one for the X.509 certificate and one for the extended certificate. This may reduce the efficiency of the CA.
3. By adding the extended certificate, the system becomes more complex and key management issues associated with the extended certificate become a complex task.
4. If two separate signatures are used, verifying two separate signatures will take more processing time and resources by the authenticators.

7.3 Using Attribute Certificates

Attribute certificates are essentially X.509 certificates without public key information. (alternatively one can perceive them as extended certificates without the X.509 certificate embedded into them.) They are intended to complement the X.509 certificate with additional information about the user (subject). This would give the same advantages and disadvantages as the PKCS#6 certificate with the additional benefits and disadvantages listed below:

7.3.1 Advantages

1. Mutual verification, via a challenge response can be performed between the holder of the attribute certificate and the user authenticator prior to sending the attribute information.
2. The attribute information can be encrypted, providing access to the confidential information to verified authenticators only.
3. Information can be separated into as many attribute certificates as needed by the system. This may be useful in meeting the “need to know” requirement of many systems.
4. Anonymity can be accommodated if the Distinguished Name (DN) of the user’s X.509 certificate is a reference, not an actual identity (i.e. a user number, database lookup, etc.). The DN can be used to match attribute certificates with X.509 certificates.
5. Attribute certificates are becoming standardized (as with X.509).

7.3.2 Disadvantages

1. Introducing multiple attribute authorities into the system architecture makes the system more complex. Key management issues may become the primary concern.
2. User authentication processing time may be an issue if two signatures must be verified, and the attribute certificate needs to be decrypted.

8.0 Conclusions

This document (Part 1) has discussed three different alternatives for incorporating authentication information into X.509 and related certificates. The determination of which certificate alternative to use must be based upon system requirements. The following sections summarize the system requirements and how to match attribute placement with those requirements.

8.1 Seven Steps to Analyze the Certificate Authentication Information

There are several system characteristics which must be evaluated prior to deciding the placement of the authentication information. Each system must determine the level of protection, the cost of implementation, and the performance factors involved. Technologically, the critical factors are:

1. What authentication information is required by the system?
2. What information should be kept private and what data can be made public? Any information in the X.509 certificate is public. Anything that needs to be confidential should be placed in an extended or attribute certificate.
3. Will the information that is planned to be placed in the certificate be stable during the validity period of the certificate? If it is not, then a certificate may not be the appropriate place for the information.
4. Is certificate storage a problem? Each certificate can add several hundred bytes in storage requirements to the system. This may be of great concern to a system which utilizes tokens (such as smartcards) to hold the certificates.
5. Is system throughput (transferring the data) a concern? Each certificate which needs to be transferred can add several hundred bytes of data I/O to be transferred. This may be of great concern to systems which utilize tokens (such as smartcards) to store the certificates.
6. Is processing time a factor? Each additional signature that needs to be verified can add hundreds of milliseconds to seconds onto the processing time already needed to verify the original signature, not to mention the processing time on the rest of the data.
7. Where will the information be used in the system? If only a small part of the information is used by most components of the system, and throughput/ processing time is a concern, then it makes sense to split the data up. Trust may also be a factor if certain components of the system can be trusted to process particular information and other components may not.

8.2 Mapping the System to the Right Solution

Now that the system is analyzed, its needs must be mapped to the strengths of the certificate alternatives. Each of the three alternatives discussed in this paper have certain advantages over the others. Refer to section 5.0 “Examining Certificate Standards” on page 17 for an overview of X.509 and a description of the alternatives. Refer to the section labeled section 7.0 “Exploring the Alternatives for Authentication Information” on page 31 for a description of the advantages and disadvantages of each alternative. The following is a summary of that section:

X.509 public key certificates

- Simplest certificate management (fewer certificates to manage)
- Based upon an international standard (and well accepted by industry)
- Fastest signature processing on ALL data (only one signature to verify)
- All certificate data is sent to all system components which need a certificate.
Note: this implies that all system components must have a “need to know” of certificate information. (all certificate information is public)
- Extensions are marked Non-Critical, implying that entities which don’t have the ability to process the information don’t have to.

PKCS#6 Extended X.509 Certificates

- Only one certificate to manage (with extended data)
- Can be signed by the same CA as the X.509 certificate.
- X.509 certificate can be “stripped out” for X.509 backward compatibility.

X.509 Attribute Certificate:

- Allows attributes to be categorized, placed in separate certificates, and sent only to entities that have a “need to know”.
- Attribute certificates can be encrypted after the recipient has been verified via a challenge/response mechanism.

The following table was created in order to convey how the alternatives described in this document support the concepts described in section 6.0 “Considering System Characteristics” on page 28 (which address the seven steps in section 8.1).

“Good” implies that the method can support the concept, “Average” implies the method could be made to support it with some complexity, and a “Poor” implies greater difficulty in making the method support the concept.

The System Weight column was added to place a system specific weight factor for each of the desired characteristics. The system designers should review each characteristic and place a weight factor (i.e. 1 to 10 where 10 implies that the characteristic is of great concern to the system and 1

implies the characteristic is of little concern to the system).

Table 5: Comparison Chart

Desired System Characteristic	X.509 Subject Directory Extension	PKCS#6 X.509 Extended Certificate	X.509 Attribute Certificates	System Weight
Attribute Identification	Good	Good	Good	TBD
Attribute Confidentiality	Poor	Poor	Good	TBD
Need To Know	Poor	Average	Good	TBD
Small Certificate Storage (assuming all certificates need to be stored)	Good	Average	Poor	TBD
Low I/O throughput (all certificates sent)	Good	Average	Poor	TBD
Low I/O throughput (one certificate sent)	Poor	Poor-Average	Good	TBD
Small Processing Time (all certificates processed)	Good	Average	Poor	TBD
Small Processing Time (one certificate processed)	Poor	Poor-Average	Good	TBD
Low Architecture Complexity	Good	Average ¹	Poor ²	TBD
Low Management Complexity	Good	Average ¹	Poor ²	TBD
Score				

1 Approximately the same as Subject Directory Extension when one signature is used for both components.

2 For the special case of only one public attribute certificate, complexity is equivalent to that of the PKCS#6 extended certificate.

To use this chart first determine the System Weight for each of the requirements. Next, place a numeric equivalent for High, Medium and low (i.e. High = 10, Medium=5, Low-Medium=3, Low = 1) in each cell of the certificate option columns. Multiply each of those cells by the System weight in that row. Add each of those results with other results in that column. Place the sum of the values in the score box in that column. The high score should indicate which option best fits the system.

There may be other requirements which have to be added to this chart for the system under design. They can be added as needed. This chart is meant to demonstrate some of the more obvious design considerations, not provide a complete and concrete methodology. Alterations or changes can and should be made as needed.

8.3 The Bottom Line

None of the three alternatives for incorporating authentication information into X.509 certificates has a clear advantage as a general purpose solution. The determination of which certificate alternative to use must be based upon system requirements. Several techniques were discussed for determining the best fit for your architecture. Refer to Part 3 of this study for an example implementation which illustrates these techniques.

APPENDIX I Orange Book Requirements

The Department of Defence Trusted Computer System Evaluation Criteria (“The Orange Book”) has six Fundamental Computer Security Requirements outlined in its introduction section. The fundamental requirements are as follows:

Policy:

Requirement 1 - SECURITY POLICY - There must be an explicit and well-defined security policy enforced by the system. Given identified subjects and objects, there must be a set of rules that are used by the system to determine whether a given subject can be permitted to gain access to a specific object. Computer systems of interest must enforce a mandatory security policy that can effectively implement access rules for handling sensitive (e.g., classified) information. These rules include requirements such as: No person lacking proper personnel security clearance shall obtain access to classified information. In addition, discretionary security controls are required to ensure that only selected users or groups of users may obtain access to data (e.g., based on a need-to-know).

Requirement 2 - MARKING - Access control labels must be associated with objects. In order to control access to information stored in a computer, according to the rules of a mandatory security policy, it must be possible to mark every object with a label that reliably identifies the object’s sensitivity level (e.g., classification), and/or the modes of access accorded those subjects who may potentially access the object.

Accountability:

Requirement 3 - IDENTIFICATION - Individual subjects must be identified. Each access to information must be mediated based on who is accessing the information and what classes of information they are authorized to deal with. This identification and authorization information must be securely maintained by the computer system and be associated with every active element that performs some security-relevant action in the system.

Requirement 4 - ACCOUNTABILITY - Audit information must be selectively kept and protected so that actions affecting security can be traced to the responsible party. A trusted system must be able to record the occurrences of security-relevant events in an audit log. The capability to select the audit events to be recorded is necessary to minimize the expense of auditing and to allow efficient analysis. Audit data must be protected from modification and unauthorized destruction to permit detection and after-the-fact investigations of security violations.

Assurance:

Requirement 5 - ASSURANCE - The computer system must contain hardware/software mechanisms that can be independently evaluated to provide sufficient assurance that the system enforces requirements 1 through 4 above. In order to assure that the four requirements of Security Policy, Marking, Identification, and Accountability are enforced by a computer system, there must be some identified and unified collection of hardware and software controls that perform those functions. These mechanisms are typically embedded in the operating system and are designed to carry out the assigned tasks in a secure manner. The basis for trusting such system mechanisms in their operational setting must be clearly documented such that it is possible to independently examine the evidence to evaluate their sufficiency.

Requirement 6 - CONTINUOUS PROTECTION - The trusted mechanisms that enforce these basic requirements must be continuously protected against tampering and/or unauthorized changes. No computer system can be considered truly secure if the basic hardware and software mechanisms that enforce the security policy are themselves subject to unauthorized modification or subversion. The continuous protection requirement has direct implications throughout the computer system's life-cycle.

The IDENTIFICATION requirement can be met in various ways. At the time of the creation of the orange book, Passwords were the only viable means of authenticating the identity of "Subjects" (i.e. users of the system). With the recent availability of low cost Tokens (primarily smart cards) and low cost biometric verification devices, improvements can be made to system which can meet the basic IDENTIFICATION requirements and greatly reduce the probability of a security violation because of poor practices(i.e. writing down passwords, choosing easy to guess passwords, etc.)

The SECURITY POLICY requirement is what "Glues" the IDENTIFICATION with MARKING requirement (i.e. only user with appropriate authorization are allowed to access to specific information). The Security policy must strike a balance between usability (i.e. user friendliness) and security control measures in order for the system to gain acceptance. This implies that different systems will create different security policies depending upon many factors (What data is protected, who is it being protected against, what is the value of the information, etc.). Flexibility in any standards used by the security policy must be made in order to adapt it to different systems.

Information about the IDENTIFICATION of the user must authenticated and assessable for use by the ACCOUNTABILITY requirement. Access to Audit information can be maintained and made accessible to individuals granted Auditor privileges by the system. The Auditor (and other privilege roles) can be biometrically verified via the IDENTIFICATION requirement.

Biometric IDENTIFICATION and token Authentication can add functionality which can make improvements which meet the ASSURANCE and CONTINUOS PROTECTION requirements. If the IDENTIFICATION improvements are utilized by the functions which maintain the

ASSURANCE and CONTINUOUS PROTECTION requirements, then the level; at which these requirements are met can be greatly enhanced.

The main goal of this document is to illustrate a means of providing a mechanism to meet or exceed the IDENTIFICATION requirement with biometric Identification and Authentication information placed in a certificate. When the improvements are made to the IDENTIFICATION function, and if the functions meeting the other requirements utilize these improvements, then the overall security of the system is greatly enhanced.

APPENDIX II X.509 Certificate ASN.1 Syntax

```

Certificate ::= SIGNED { SEQUENCE {
    version [0] Version DEFAULT v1,
    serialNumber CertificateSerialNumber,
    signature AlgorithmIdentifier,
    issuer Name,

    validity Validity,
    subject Name,
    subjectPublicKeyInfo SubjectPublicKeyInfo,

    issuerUniqueIdentifier [1] IMPLICIT UniqueIdentifier OPTIONAL,
        -- if present, version must be v2 or v3

    subjectUniqueIdentifier [2] IMPLICIT UniqueIdentifier OPTIONAL
        -- if present, version must be v2 or v3

    extensions [3] Extensions OPTIONAL -- If present, version must be v3 -- }}
Version ::= INTEGER { v1(0), v2(1), v3(2) }

```

CertificateSerialNumber ::= INTEGER

```

AlgorithmIdentifier ::= SEQUENCE {
    algorithm ALGORITHM.&id ({ SupportedAlgorithms }),
    parameters ALGORITHM.&Type ({ SupportedAlgorithms } { @algorithm })
OPTIONAL }

```

```

-- Definition of the following information object set is deferred, perhaps to standardized
-- profiles or to protocol implementation conformance statements. The set is required to
-- specify a table constraint on the parameters component of AlgorithmIdentifier.
-- SupportedAlgorithms ALGORITHM ::= { ... }

```

```

Validity ::= SEQUENCE {
    notBeforeTime,
    notAfterTime }

```

```

SubjectPublicKeyInfo ::= SEQUENCE {
    algorithm AlgorithmIdentifier,
    subjectPublicKey BIT STRING }

```

```

Time ::= CHOICE {
    utcTime UTCTime,
    generalizedTime GeneralizedTime }

```

Extensions ::= SEQUENCE OF Extension

Extension ::= SEQUENCE {
 extnId EXTENSION.&id ({ExtensionSet}),
 critical BOOLEAN DEFAULT FALSE,
 extnValue OCTET STRING
 -- contains a DER encoding of a value of type &ExtnType
 -- for the extension object identified by extnId -- }

ExtensionSetEXTENSION ::= { ... }

EXTENSION ::= CLASS {

 &id OBJECT IDENTIFIER UNIQUE,
 &ExtnType }
WITH SYNTAX {
 SYNTAX &ExtnType
 IDENTIFIED BY &id }

APPENDIX III X509 Attribute Certificate ASN.1 Syntax

AttributeCertificate ::= SIGNED { AttributeCertificateInfo }

AttributeCertificateInfo ::= SEQUENCE {
 versionVersion DEFAULT v1,
 subject CHOICE {
 baseCertificateID[0]IssuerSerial, -- associated with a Public Key Certificate
 subjectName [1] GeneralNames }, -- associated with a name
 issuer GeneralNames, -- CA issuing the attribute certificate
 signature AlgorithmIdentifier,
 serialNumberCertificateSerialNumber,
 attrCertValidityPeriodAttCertValidityPeriod,
 attributes SEQUENCE OF Attribute,
 issuerUniqueIDUniqueIdentifier OPTIONAL,
 extensions Extensions OPTIONAL }

IssuerSerial ::= SEQUENCE {
 issuer GeneralNames,
 serial CertificateSerialNumber,
 issuerUID UniqueIdentifier OPTIONAL }

AttCertValidityPeriod ::= SEQUENCE {
 notBeforeTimeGeneralizedTime,
 notAfterTimeGeneralized Time }

