User Authentication Techniques
Using Public Key Certificates
Part 2: Authentication Information Including Biometrics

31 December 1997

Lawrence A. Reinert
Stephen C. Luther
## TABLE OF CONTENTS

1.0 Scope .........................................................1
2.0 Background ...................................................1  
  2.1 Related Standards ..........................................1  
    2.1.1 X.509 Attributes .....................................1  
    2.1.2 ECMA.219 Authentication Information Types ........2  
  2.2 An Overview of Biometrics .................................2  
  2.3 Terminology ...............................................4  
  2.4 Symbols and Abbreviations .................................6  
3.0 Reference Documents .........................................7  
4.0 Discussing Authentication Information .....................8  
  4.1 Requirements for Authentication Information ............8  
  4.2 Existing Authentication Information Standards - ECMA.219 .9  
    4.2.1 Passwords ...........................................10  
    4.2.2 Tokens ...............................................10  
    4.2.3 Biometric Information ...............................11  
    4.2.4 Trusted Third Party ................................11  
    4.2.5 Context ..............................................12  
    4.2.6 ECMA.219 Methods ..................................12  
5.0 Placing Authentication Information into X.509 Certificates ....14  
  5.1 X.509 Attributes .........................................14  
  5.2 Defining a new Attribute ................................14  
    5.2.1 Additional Authentication Methods ..................15  
    5.2.2 Additional Detailed Biometric Information ..........15  
      5.2.2.1 Processing Information ...........................18  
        5.2.2.1.1 Registering Biometric Processes ............18  
        5.2.2.1.2 Biometric Processing Parameters ..........18  
      5.2.2.2 Matching Information .............................19  
        5.2.2.2.1 Registering Biometric Matching Methods ....20  
        5.2.2.2.2 Biometric Matching Parameters ..........20  
6.0 Conclusions ..................................................21
1.0 Scope

The intent of the three part study is to investigate placement of user authentication information within public key certificates. Part 1 [10] examines the options within X.509 (and related standards) for the placement of user authentication information and how system requirements affect the decision process. This document (Part 2) investigates specific authentication information, such as biometrics and passwords, which should be considered for placement in a public key certificate prior to the development of a system. This document also considers the options for placement of this information in current standards. Part 3 [11] presents a case study of how to architect such a system through an example implementation.

2.0 Background

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. With more robust components, the capability to remove the threat of unauthorized access to computing resources is on the horizon.

2.1 Related Standards

Reports, studies, or specifications which deal with the use of authentication information contained within public key certificates are a rare find. There are, however, many documents which discuss the use of authentication information within a system (without the use of public key certificates).

Not all references made to user authentication imply the use of authentication information. As an example, X.501 Amendment 1 (equivalent to ISO/IEC 9594-2 DAM1) discusses the representation of access control information. Access control information is represented as a set of ACIItems. The ACIItem does have an authentication level which can specify to what level the system must authenticate the users, but it does not define the exact method or data to be used to verify them.

The following section discusses the few references found in open standards to authentication information.

2.1.1 X.509 Attributes

As discussed in Part 1 of this study, The Subject Directory Attribute Extension, the Attribute Certificate, and the PKCS Extended Certificate all make use of (registered) attributes. In all of these cases, the attributes can be used to provide additional information about the subject (user), including authentication information. Placing authentication information in any of these fields is a viable means for any system to implement authentication information.
2.1.2 ECMA.219 Authentication Information Types

The European Computer Manufacturers Association published standard ECMA.219 [1] entitled the “Authentication and Privilege Attribute Security Application with Related Key Distribution Function.” ECMA.219 defines three types of applications: an Authentication Application, a Privilege Application, and a Key Distribution Application. These are distributed security applications providing services concerned with authentication and access control, along with related key distribution information. The field of application of the ECMA standard is the design and implementation of distributed open systems that support: authentication of human users and software entities, secure access of users to applications, and secure access between distributed applications.

Most of the application elements discussed in ECMA.219 are not applicable to this study. However, it does contain a useful discussion on Authentication Information that can be used as a basis for further discussion. Annex B of ECMA.219 gives examples of specific Authentication Information. It categorizes the information in terms of several authentication principles and discusses how they can be combined. This information will be used as a basis for the structure of the Authentication Information.

2.2 An Overview of Biometrics

Biometrics incorporate 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 system incorporating biometric verification (as illustrated in figure 1) usually has a biometric scanner which takes a biometric sample or “image” of the individual. The biometric device may be a microphone, video camera, fingerprint reader, etc. The other components of a biometric system consist of a processing algorithm and a matching algorithm. These may be hardware or software components, but most implementations generally utilize software algorithms running on a workstation.

Figure 1: Biometric Components.
The user enrollment process (figure 2) consists of taking a sample or biometric “image” which is specific to the type of biometric sample being utilized (i.e. a video image for facial recognition systems, voice recording for voice print systems, fingerprint image for fingerprint verification systems, etc.). The biometric image is usually processed using a proprietary processing algorithm. The processing algorithm extracts specific information from the image (wavelengths, minutia points, eye location, etc.) and stores this information in a data object called a “biometric template.” The biometric template is used for comparisons in user verification. The template cannot usually be used to recreate the original image, but instead the live scan is fed through the algorithm to create another digital template which can then be compared to the original template.

*Figure 2: Biometric Enrollment*

The user verification (figure 3) takes place when the system requires biometric authentication for some process (typically during each login process). A live scan of the user is taken, processed into a biometric template, and compared against the template taken during the enrollment process. The comparison is performed by a “matching algorithm.” The matching algorithm typically gives a measurement of the degree of certainty of which the user was verified (i.e. 0 to 100 where 100 is an exact match and 0 is none at all). The measurement of certainty is typically referred to as the “Matching Score”. Since most biometric matching takes place from images that are taken at different points in time, no two images can be identical, and consequently no match is ever an exact match. The system has to determine what level it will accept as a match. The higher the level, the more false rejects the system will experience (i.e. legitimate users that are rejected because a poor sample was taken). Systems having thresholds set on a per user basis allow certain users with poor biometric qualities to pass at a lower level without sacrificing system security.
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.

### 2.3 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. To establish the validity of a claimed identity

**Biometric:** A measurable, unique physiological characteristic or behavioral trait used to recognize the identity, or verify the claimed identity, of an enrollee (i.e. a subject or user).

**Biometric Image:** The raw (unprocessed) output of the biometric scanner. The biometric image is generally fed into a biometric extraction algorithm.

**Biometric 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. The biometric template is generally the output of biometric extraction routine(s), such as a Minutia Extraction Algorithm.

**Certify:** To confirm formally as true, accurate, or genuine.
**Certificate:** A document testifying to accuracy or truth.

**Enrollment:** The process of collecting biometric samples from a person and the preparation and storage as a template of that person’s identity

**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).

**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.

**Imposter:** A person who submits a biometric sample in either an intentional or inadvertent attempt to pass him/herself off as another person who is an enrollee.

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

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

**Response time:** The time period required by a biometric verification device to complete a biometric transaction.

**Security Policy:** The set of laws, rules, and practices that regulate how an organization manages, protects, and distributes sensitive information

**SSO:** Site Security Officer

**Subject:** An active entity, generally in the form of a person, process, or device that causes information to flow among objects or changes the system state.

**Zero Knowledge:** a protocol used to establish the identity of a user by proving knowledge of a secret but without revealing that secret. An example is a challenge-response protocol.
2.4 Symbols and Abbreviations

This section contains symbols and abbreviations used in this document.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASN.1</td>
<td>Abstract Syntax Notation</td>
</tr>
<tr>
<td>AI</td>
<td>Authentication Information</td>
</tr>
<tr>
<td>AUC</td>
<td>Authentication Certificate</td>
</tr>
<tr>
<td>DAM</td>
<td>Draft Amendment</td>
</tr>
<tr>
<td>ECMA</td>
<td>European Computer Manufacturers Association</td>
</tr>
<tr>
<td>GUC</td>
<td>Generalized User Certificate</td>
</tr>
<tr>
<td>ID</td>
<td>Identifier</td>
</tr>
<tr>
<td>IEC</td>
<td>International Electrotechnical Commission</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>NSA</td>
<td>National Security Agency</td>
</tr>
<tr>
<td>PAC</td>
<td>Privilege Attribute Certificate</td>
</tr>
<tr>
<td>PIN</td>
<td>Personal Identification Number</td>
</tr>
<tr>
<td>PKCS</td>
<td>Public Key Cryptography Standard</td>
</tr>
<tr>
<td>SSO</td>
<td>Site Security Officer</td>
</tr>
<tr>
<td>UIC</td>
<td>Unified INFOSEC Criteria</td>
</tr>
</tbody>
</table>
3.0 Reference Documents


4.0 Discussing Authentication Information

Users’ identities are verified using one or more of three generic methods (types): something they know (PINs, passwords, memory phrases, etc.), something they have (a physical token such as a magnetic stripe card, a physical key, a smartcard, etc.), or something they are (biometric verification). If this information is gathered by a trusted process, verified, and then signed by a trusted authority, it can be considered as Authenticated Authentication Information.

The following sections discuss the requirements imposed upon Authentication Information and explore applicable standards.

4.1 Requirements for Authentication Information

The Unified INFOSEC Criteria (UIC), reference [9], offers the following set of requirements for authentication and authentication information:

1. Users’ claimed identities should be verified using more than one type of authentication.

2. The path over which the identifiers and authenticators are passed should be provided adequate source authentication, data integrity, and/or access control.

3. The authentication component should be provided an adequate level of assurance

4. Authentication, after initial session/connection, should be re-verified at least periodically, if not on a continual basis.

5. Authentication should be two way (i.e. user to host/resource and host/resource to user).

In addition the following common sense requirements should be applied:

6. Solutions should be generic and allow for a wide variety of implementations.

7. Solutions should be open and standardized. This implies that current standards should be utilized, if possible.

8. The authentication information transfer mechanism should allow for multiple values of different types. That is to say, more than one biometric template of each type should be allowed, as well as multiple types of biometrics (i.e. it should allow storage for multiple fingers of a given subject to be enrolled, as well the hand geometry, retina image, and facial image).

These requirements, if possible, should be used to drive the solution for placing authentication information in X.509 certificates. The existing requirements for the implementation of X.509 certificates should already satisfy requirements 2, 3 and parts of 5, 6, and 7(assuming the system security policy upholds these requirements).
4.2 Existing Authentication Information Standards - ECMA.219

ECMA.219 defines three types of applications: an Authentication Application, a Privilege Application, and a Key Distribution Application. In support of the Authentication Application, the specification defines several authentication information types. The information types are categorized in terms of the basic principles which govern the authentication methods. The principles, as defined in Annex B, are as follows:

Principle A - something known
Principle B - something possessed
Principle C - immutable characteristics
Principle D - trusted third party
Principle E - context (location)

The definitions given in Annex B are for methods (functions) not data. In ECMA.219 Annex E, ASN.1 abstract service, a structure is defined which binds the method with parameters, including data. The definition of the structure is as follows:

AuthenticationInfo ::= SEQUENCE {
  authenticationMethod [0] AuthenticationMethod -- defined above
  exchangeAI [1] AuthMparm
}

AuthenticationMethod ::= OBJECT IDENTIFIER

AuthMparm ::= CHOICE {
  printableValue [0] Printable String,
  integerValue [1] INTEGER,
  octetValue [2] OCTET STRING,
  bitStringValue [3] BIT STRING,
  otherValue [4] ANY } -- defined by authenticationMethod

This specification has many places where the authentication information can be placed, since they are defined as object identifiers. These include the ECMA defined certificate structures: the Authentication Certificate (AUC), the Privilege Attribute Certificate (PAC) and the Generalized User Certificate (GUC). However, the definition of authentication information and the object identifiers are exportable to other definitions as needed (i.e. they can be referenced or used as “imports” to other specifications). Therefore, this specification may provide a useful, existing definition to build upon as needed.

ECMA.219 states that the detailed description of the authentication information is beyond the scope of the document. Therefore, the following sections will build upon the definitions given in ECMA.219 Annex B. Further detail will be given to better utilize and try to standardize the basic concept illustrated in ECMA.219.
4.2.1 Passwords (Principle A)

ECMA.219 provides seven methods for authenticating human users with a password. A password can take three forms, or the password can be used to form a cryptographic key under which exchange-AI is encrypted. The method identifiers are as follows:

- `password ::= {id-am 0}`
- `encryptedPassword ::= {id-am 1}`
- `encryptedRepPassword ::= {id-am 2}`
- `hashedPassword ::= {id-am 3}`
- `hashedRepPwd ::= {id-am 4}`
- `passwordAsKey ::= {id-am 5}`

Where the associated password authentication information is defined as:

- `PasswordAI::= BIT STRING`

ECMA.219 also supplies definitions for the associated authentication information but they will not be described here. The `encryptedPassword` suffers the same limitation as do all other encrypted attributes. That limitation is the need to possess the key at the time the certificate is signed. Encrypting fields in a certificate, at the time the certificate is signed, limits the information to those who hold the specific key. It is for that reason that the use of encrypted fields within a certificate is discouraged.

X.509 does define a password attribute. This is to be used where the X.509 defined “simple authentication procedure” is used. If the system uses only a password, then the other information discussed in this section may not be necessary.

4.2.2 Tokens (Principle B)

There are several authentication methods available for the “something possessed” principle (principle B). The methods are identified by the object identifiers below:

- `passiveToken ::= {id-am 6}`
- `activeToken ::= {id-am 7}`

The authentication information associated with these methods is described as:

- `DeviceAI::= BIT STRING`

A useful interpretation of the `DeviceAI` would be the serial number of the token. If the authentication information is to be used strictly with a specific token, then the serial number of the token would be a useful deterrent to copying the information into another token (assuming the serial number would be verified by the host communicating with the token).
If the token is an active token, then further verification can be performed (such as a challenge/response procedure).

### 4.2.3 Biometric Information (Principle C)

Principle C defined the “immutable characteristics” of humans. The description distinguishes between some methods identified by the following identifiers:

- `voicePrint ::= {id-am 8}`
- `signature ::= {id-am 9}`
- `fingerPrint ::= {id-am 10}`
- `retinaPattern ::= {id-am 11}`

The methods which operate under principle C (immutable characteristics) have authentication information (AI) with the following syntax:

- `voicePrintAI ::= BIT STRING`
- `signatureAI ::= BIT STRING`
- `fingerPrintAI ::= BIT STRING`
- `retinaPatternAI ::= BIT STRING`

Unfortunately, the designers of ECMA.219 did not take into consideration the possibility that there might be multiple approaches to implementing biometric systems which rely on the same physical features, or multiple biometrics of the same type (e.g. several fingerprints are used). To properly support the each of these systems, a further refinement of the authentication information syntax is required. The section labeled “Additional Detailed Biometric Information” on page 15 is an attempt to provide this refinement.

### 4.2.4 Trusted Third Party (Principle D)

If a trusted third party has established authentication, then some information authorized by it may be passed as exchange AI. The methods described in ECMA.219 are limited to zero-knowledge or a user certificate. The methods are identified by the following identifiers:

- `userCertificate ::= {id-am 12}`
- `zeroKnowledgeMeth1 ::= {id-am 13}`

The certificate method authentication information has the following form:

`DirectoryAI ::= SEQUENCE
  certificate Certificate, -- Defined in X.509
  token Token, -- Defined in X.501`
The ZeroKnowledge authentication information is described as:

ZeroKnowledgeAI ::= CHOICE{
    challenge [0] BIT STRING
    response [1] BIT STRING
}

4.2.5 Context (Principle E)

The context as defined in ECMA.219 is the physical location of the user. Hence, the method is denoted ‘location’ and will typically be some form of address of the equipment used by the user. The identifier of the method is defined as:

location ::= {id-am 14}

The authentication information associated with the method is defined as:

LocationAI ::= BIT STRING -- Some form of address

4.2.6 ECMA.219 Methods

ECMA.219 defined and registered the methods used to provide the authentication functions. The methods were registered as “methods” not attributes. The methods are assigned an object identifier type of id-am.

The object identifiers below are found in Annex F of ECMA.219.

passwords ::= {id-am 0}
encryptedPassword ::= {id-am 1}
encryptedRepPwd ::= {id-am 2}
hashedPassword ::= {id-am 3}
hashedRepPwd ::= {id-am 4}
passwordAsKey ::= {id-am 5}
pasiveToken ::= {id-am 6}
activeToken ::= {id-am 7}
voicePrint ::= {id-am 8}
signature ::= {id-am 9}
fingerPrint ::= {id-am 10}
retinaPattern ::= {id-am 11}
userCertificate ::= {id-am 12}
zeroKnowledgeMeth1 ::= {id-am 13}
location ::= {id-am 14}
passwordBits ::= {id-am 15}
symmetricCrypto::= {id-am 16}
asymmetricCrypto::= {id-am 17}
applicationName::= {id-am 18}

All the methods which operate under principle C (immutable characteristics) have authentication information (AI) with the following syntax:

voicePrintAI::= BIT STRING
signatureAI::= BIT STRING
fingerPrintAI::= BIT STRING
retinaPatternAI::= BIT STRING

ECMA.219 groups the methods and data for the method using a construct for one of its services. The construct is as follows:

AuthenticationInfo::=SEQUENCE{
    authenticationMethod[0] AuthenticationMethod -- defined above
    exchangeAI[1]AuthMparm
}

AuthenticationMethod::=OBJECT IDENTIFIER
AuthMparm::=CHOICE{
    printableValue [0] Printable String,
    integerValue [1] INTEGER,
    octetValue[2] OCTET STRING,
    bitStringValue[3] BIT STRING,
    otherValue [4] ANY} -- defined by authenticationMethod (i.e. the AI)

However, the authentication information is not assigned an object identifier. This makes it unusable outside of the context in which it was created (as with an X.509 certificate). The authentication method is assigned an object identifier and can be exported as needed.
5.0 Placing Authentication Information into X.509 Certificates

ECMA.219 described the methods used to provide the authentication functions. The Subject Directory Attribute Extension, the Attribute Certificate, and the PKCS Extended Certificate (as described in Part 1 of this study) all make use of (registered) attributes. The ECMA defined methods and currently registered attributes could be used without modification and be placed in an X.509 certificate or be placed directly in the locations described Part 1 of this study, as long they are the only authentication techniques that the system employing the authentication information needs. If the system utilizes AI such as a biometric, then more information is needed.

5.1 X.509 Attributes

X.509 imports the attribute definition from X.501.

The X.501 defined attribute (that is AttributeTypeandValue) is as follows:

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

All attributes are assigned an identifier using an object type of id-at. Any registered attribute, assigned a unique identifier by an ISO recognized standards body, can be used. X.520 is a source for ISO defined attributes; however, many other standards body have registered attributes which may used.

There as several authentication information attributes already defined. Among them are:

- userPassword OBJECT IDENTIFIER ::= {id-at 35}
- userCertificate OBJECT IDENTIFIER ::= {id-at 36}
- x121Address OBJECT IDENTIFIER ::= {id-at23}

These attributes can be placed directly in the locations described in Part 1 of this study as long they are the only authentication techniques the system needs.

5.2 Defining a new Attribute

To utilize the authentication information in an X.509 or related certificate, the authentication information would have to be defined as an attribute. If the Authentication Info construct, as defined in ECMA.219 (and discussed in the section labeled “Existing Authentication Information Standards - ECMA.219” on page 9), is given the attribute syntax, the following attribute is the result:
TOKENEER

authenticationInfo ATTRIBUTE::={
  WITH SYNTAX AuthenticationInfo,
  ID id-at-TBD}

AuthenticationInfo ::= SEQUENCE{
  authenticationMethod[0] AuthenticationMethod, -- defined in ECMA.219
  exchangeAI[1]AuthMparm, -- defined in ECMA.219
  biometricInfo BiometricInfo OPTIONAL -- defined in section 5.2.2 of this document}

Since all the attribute placement options discussed in Part 1 of this study can have multiple attributes per use, it is suggested that only one information object be placed in an attribute, if multiple objects are used. This implies that if the application is using five fingerprints per user, each fingerprint would be placed in a separate AuthenticationInfo attribute. This will allow the application using the information to scan the attributes and select the appropriate one.

5.2.1 Additional Authentication Methods

As technology advances, additional authentication methods will be developed. If the methods are not listed in ECMA.219, then the authentication method should be registered with the appropriate registry authority. Such additional methods may include:

Handwriting Recognition: An individual is prompted to write random phrases on a touchpad. This data is then compared to samples in a database.

Hand Geometry: An immutable characteristic which utilizes the bone structure of the hand to verify individuals.

Facial Recognition: An immutable characteristic which utilizes facial characteristics to verify individuals.

5.2.2 Additional Detailed Biometric Information

There are many different processes capable of performing biometric verification under a specific method (i.e. there can be different processes for performing fingerprint verification). In order to allow for the continual evolution of biometric products, a data structure must be defined which can hold varying formats of data and information needed to identify the information for correct processing. This data structure must be capable of evolving with the biometric technology and it must be capable of customizing by the applications which may utilize a subset of the information.
Such a structure may contain the following fields:

**Table 2: Biometric information fields**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProcessingInfo</td>
<td>These (optional) fields are used to provide information to the process(es) which are used to create a livescan biometric template to compare against the biometric template found in this structure.</td>
</tr>
<tr>
<td>MatchingAlgorithmInfo</td>
<td>These fields specify which algorithm(s), and their respective versions, the biometric template is compatible with. They also provide any algorithm specific information.</td>
</tr>
</tbody>
</table>

Where each of the information fields will consist of the following data:

**Table 3: Biometric information fields**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Each biometric process or matching algorithm must be registered by the manufacturer with the appropriate standards body. This will insure a unique identifier for application to determine compatibility.</td>
</tr>
<tr>
<td>Parameters</td>
<td>Each manufacturer that registers its process or matching algorithm must document the parameters which are used with it and can be placed in a certificate.</td>
</tr>
<tr>
<td>Version</td>
<td>The version of the component used to create the information. Note: the decision to have compatibility with the information will be left to the application.</td>
</tr>
</tbody>
</table>

The following diagram is meant to illustrate how the biometric processing and matching parameters would be utilized during a biometric verification process:
Figure 4: Using an X.509 Certificate with Detailed Biometric Information

The certificate used to store the biometric information would be transferred to the entity performing the verification (from a database, smartcard, disk, etc.). The entity would verify the signature on the X.509 certificate to detect alteration and to prove the validity of the biometric template. The processing parameters are fed to the biometric processing function which converts the livescan image to a livescan biometric template. The livescan biometric template, the biometric template from the X.509 certificate and the matching algorithm parameters from the X.509 certificate are fed into the matching algorithm for verification of the user.
The ASN.1 Syntax of the Biometric information would be as follows:

\[
\text{BiomtricInfo} ::= \text{SEQUENCE}\{
\hspace{1cm} \text{processingInfo} \hspace{1cm} \text{ProcessingAlgorithmInfo OPTIONAL,}
\hspace{1cm} \text{matchingInfo} \hspace{1cm} \text{MatchingAlgorithmInfo}\}
\]

Where the ProcessingInfo and MatchingInfo are defined in the following sections.

### 5.2.2.1 Processing Information

Biometric processing is the function which takes a biometric sample (typically a video image or an audio sample), extracts information from the sample (such as the location of the minutia in the fingerprint), and creates a output file (typically called a biometric template). The processing information field would be used to provide processing algorithm specific information which may be used to personalize the process for the individual. The algorithm used to create the biometric template is specified by the processingAlgorithmID. processAlgorithmParams is used to provide process specific parameters.

The syntax for this field would be defined as:

\[
\text{ProcessingInfo} ::= \text{SEQUENCE}\{
\hspace{1cm} \text{processingID} \hspace{1cm} \text{OBJECT IDENTIFIER,} \hspace{1cm} \text{Defined by implementation}
\hspace{1cm} \text{processingParms} \hspace{1cm} \text{AuthMparm OPTIONAL,} \hspace{1cm} \text{Defined in ECMA.219}
\hspace{1cm} \text{processingVersion} \hspace{1cm} \text{Version} \hspace{1cm} \text{Defined in X.509}\}
\]

Version is defined in X.509 as follows:

\[
\text{Version} ::= \text{INTEGER\{v1(0)\}} \hspace{1cm} \text{Add versions as needed}
\]

### 5.2.2.1.1 Registering Biometric Processes

The need to identify this process is minimal from a certificate’s point of view, unless the process creating the livescan sample to compare against the certificate requires some customizing in regard to the individual who is being sampled. If this is the case, then the individual process creating the template needs to be registered by a recognized standards body. The OBJECT IDENTIFIER assigned during the registration process will be used to associate the parameters defined below.

### 5.2.2.1.2 Biometric Processing Parameters

As stated above, biometric processes only need to be registered if there are associated parameters that need to be sent. The individual processing parameters do not have to be registered as long as they are defined and maintained by the organization which registered the process. The processing parameters are associated with that particular OBJECT IDENTIFIER.
Examples of processing parameters might be:

Minimal Acceptable Quality: A minimum quality that the sample must have to be accepted for further processing (useful if the particular biometric can obtain preliminary quality ratings on a sample). This may relieve the need for users with poor biometric characteristics (such as a scarred finger) to re-enter a biometric sample several times for verification.

Number of Samples: The number of samples that should be taken of the user which meet the MinimumAcceptableQuality threshold. This will also help users with poor biometric characteristics to avoid re-entering a biometric sample several times.

The processing parameters could follow the established by ECMA.219 AuthMparm or define an entirely new syntax using the ANY option.

The application would be responsible for determining compatible versions. If the versions are incompatible, then the processing information may have to be rejected, and therefore the authentication process would have to fail.

Such parameters should, and could, be standardized upon to reduce the overhead for systems which want to incorporate multiple biometric devices. This is likely to happen in the future as biometric technology matures.

5.2.2.2 Matching Information

Biometric matching is the function (algorithm) which takes two biometric templates and compares them for similarities. The output of the matching function is typically a matching score representing the amount of similarity found between the two templates.

The biometric templates are generally designed to work with a specific biometric matching algorithm. The application can reference the ID of the MatchingInfo in this field to determine compatibility.

Parameters for the matching algorithm may be supplied to provide matching algorithm specific information to personalize the matching process for the individual. The algorithm used to match the biometric template is specified by the matchingAlgorithmID. processAlgorithmParams is used to provide process specific parameters (such as matching thresholds).

The syntax for this field is defined as follows:

MatchingInfo ::= SEQUENCE {
    matchingID OBJECT IDENTIFIER, -- Defined by implementation
    matchingParm AuthMparm OPTIONAL, -- Defined in ECMA.219
}
Version is defined in X.509 as follows:

Version ::= INTEGER { v1(0) } -- Add versions as needed --

5.2.2.2.1 Registering Biometric Matching Methods

The currently available biometric related Software Development Kits (SDK) offer company/device specific routines and data. Although there may be standardized Application Programming Interfaces (APIs) developed in the future, there are enough existing proprietary devices and methods in circulation to justify the need to differentiate between the various implementations. If a system utilizes more than one biometric device, there must be a way to choose which software to use. Registering the biometric matching method would be one means of doing so.

The individual process which matches the biometric template placed in the X.509 certificate against the livescan biometric template needs to be registered by a recognized standards body. The OBJECT IDENTIFIER assigned during the registration process will be used to associate the parameters defined below.

5.2.2.2.2 Biometric Matching Parameters

As mentioned above, the matching identifier can be used to determine compatibility between the X.509 certificate and the system being accessed. If there are any parameters required for the matching process, then the matching parameters can be used to provide this information securely.

An example of a matching parameter might be:

- Minimal Acceptable Matching Threshold: This may be used to adjust the sensitivity of the matching process for individual with poor biometric characteristics.

- Template Identifier: This may be used to distinguish the biometric template from other found in certificate. It may be used to for such distinctions as which finger the template represents (index finger, thumb, etc.).

The matching parameters would be defined and maintained by the organization which registered the matchingID. The application would be responsible for determining which versions it is compatible with. If the versions are incompatible, then the matching information may have to be rejected, and therefore the authentication process would have to fail.

As with the biometric processing parameters, these parameters should and could be standardized upon to reduce the overhead for systems which want to incorporate multiple biometric devices.
6.0 Conclusions

There currently exists no attribute definition that can be used in a X.509 or related certificate if a system requires multiple types of authentication information. By using some of the Authentication Information definitions provided by ECMA.219, we can construct a new attribute for placing the information in an X.509 certificate. The complete syntax of the attribute would be as follows:

```plaintext
authenticationInfo ATTRIBUTE ::= {
    WITH SYNTAX AuthenticationInfo,
    ID id-at-TBD
}
```

```plaintext
AuthenticationInfo ::= SEQUENCE {
    authenticationMethod[0] AuthenticationMethod, -- defined in ECMA.219
    exchangeAI [1] AuthMparm, -- the data, as defined in ECMA.219
    biometricInfo BiometricInfo OPTIONAL -- defined in section 5.2.2 of this document
}
```

```plaintext
BiometricInfo ::= SEQUENCE {
    processingInfo ProcessingInfo OPTIONAL,
    matchingInfo MatchingInfo
}
```

```plaintext
ProcessingInfo ::= SEQUENCE {
    processingID OBJECT IDENTIFIER, -- Registered by implementation
    processingParms AuthMparm OPTIONAL, -- Defined in ECMA.219
    processingVersion Version} -- Defined in X.509
```

```plaintext
MatchingInfo ::= SEQUENCE {
    matchingID OBJECT IDENTIFIER, -- Registered by implementation
    matchingParm AuthMparm OPTIONAL, -- Defined in ECMA.219
    matchingVersion Version} -- Defined in X.509
```

--- ECMA.219 definitions

```plaintext
AuthMparm ::= CHOICE {
    printableValue [0] Printable String,
    integerValue [1] INTEGER,
    octetValue [2] OCTET STRING,
    bitStringValue [3] BIT STRING,
    otherValue [4] ANY} -- defined by authenticationMethod (i.e. the AI)
```

--- X.509 Definitions

```plaintext
Version ::= INTEGER {v1(0)} -- Add versions as needed --
```
Each implementation of a biometric matching algorithm and biometric processing function would require registration with an appropriate standards body. The processing functions would only have to register identifiers if they need to send parameters to the processing function which takes a livescan for comparison.

For standardization, the AuthenticationInfo attribute must be registered along with the provided, or a similar, definition. The registration of this attribute is beyond the scope of this study.