

**NATIONAL SECURITY AGENCY
CENTRAL SECURITY SERVICE**



R222

INFOSEC Engineering

**Guidelines for Placing Biometrics in
Smartcards**

Version 1.0

September 15, 1998

**Prepared for the GSA Government Smart Card Group
by R223 Experimental System Architecture
September 22, 1998**

TABLE OF CONTENTS

Guidelines for Placing Biometrics in Smartcards

1.0	Scope	1
1.1	Purpose of this document	1
2.0	Background	1
2.1	Automated Information Systems (AIS)	1
2.2	Smartcards	2
2.3	Authentication Information (Biometrics)	3
2.4	Certificates	3
	2.4.1 Public Key Certificates	3
	2.4.2 Attribute Certificates	4
2.5	Public Key Infrastructure (PKI)	5
3.0	The Need to Protect Authentication Information	6
3.1	Why Protect AI?	6
3.2	Why Protect the Smartcard?	6
3.3	Why Protect Biometric Information?	6
3.4	Steps to Secure Authentication Information	7
4.0	Guidelines for Protecting Authentication Information	8
4.1	System Consideration Guidelines	8
4.2	SmartCard Guidelines	9
4.3	Biometric Guidelines	10
4.4	Certificate Guidelines	11
4.5	Public Key Infrastructure Guidelines	12
5.0	Conclusions	12
6.0	References	13

Appendix A: System Considerations

1.0	Background	1
1.1	Threat Definition	1
	1.1.1 Define the Data you are protecting	1
	1.1.2 Define Your Adversary	1
	1.1.3 Define the System Security Policy	2
1.2	Trust	2
1.3	User Confidentiality	3
1.4	Need To Know	4
1.5	Processing Time	4
1.6	System Architecture	4
1.7	Storage	4
1.8	Export Restrictions	4
2.0	System risks due to the smartcard	5
2.1	Defining the data on the smartcard	5
2.2	How can the data be lost?	5
2.3	What happens if the Government Smartcard is lost?	5
2.4	What can be done to prevent data from being lost or stolen?	6

Appendix B: SmartCards

1.0 Background	1
1.1 Vulnerabilities	1
1.1.1 Physical attacks on the smartcard	1
1.1.2 Attacks between the host and the smartcard	1
1.1.3 Steps to prevent the attacks	1
1.2 Multi-Application Smartcards	1
1.3 Cryptographic Algorithms	2
1.3.1 Cryptographic Co-processors	2
1.4 Overview of token based security mechanisms	2
1.4.1 Lock/Unlock	2
1.4.2 Mutual Authentication	4
1.5 Confidentiality through Encryption	6
2.0 Certificate Storage Requirements	7
3.0 Token Authentication Protocol (TAP)	8
3.1 Step 1	8
3.2 Step 2	8
3.3 Step 3	9
3.4 Step 4	9
4.0 Physical Protection	10
5.0 NIST Certification	11

Appendix C: Biometrics

1.0 Background	1
2.0 Selecting Biometrics	2
2.1 User Considerations	2
2.1.1 Public Acceptance	2
2.1.2 User Acceptance	2
2.1.3 Target Clientele Characteristics	3
2.1.4 User Difficulties	3
2.1.5 Ease of Use	4
2.2 Implementation Considerations	4
2.2.1 Enrolled Image Quality	4
2.2.2 False Acceptance/False Rejection	4
2.2.3 Uniform Testing	4
2.2.4 Circumvention	5
2.2.5 Cost	5
2.2.6 Continuous Verification	5
2.2.7 Template Storage	5
2.2.8 Data Channel Security	5
2.2.9 Computer Resources	6
2.2.10 Comparison Engine Location	6
2.2.11 Calibration	6
2.3 Product Considerations For Use With Smartcards	6
2.3.1 Smartcard Storage	6
2.3.2 Processing Time	6
2.3.3 Biometric Upgrades/Obsolescence	7
3.0 Placing Biometric Templates on the Government Smartcard	8

Appendix D: Certificates

1.0 Background	1
1.1 The X.509 Certificate	1
1.1.1 Attributes	3
1.2 Attribute Certificates	4
1.2.1 Attribute Certificate Advantages/Disadvantages	6
2.0 Using Certificates for Authentication	7
2.1 Encoding Rules	7
2.2 Signature Certificate Processing	7
2.3 Using Attribute Certificates	7
2.3.1 X.509 Attributes	7
2.3.2 Defining a new Attribute	8
2.3.2.1 Additional Authentication Methods	8
2.3.2.2 Additional Detailed Biometric Information	9
2.3.2.3 Processing Information	11
2.3.2.3.1 Registering Biometric Processes	11
2.3.2.3.2 Biometric Processing Parameters	11
2.3.2.4 Matching Information	12
2.3.2.4.1 Registering Biometric Matching Methods	13
2.3.2.4.2 Biometric Matching Parameters	13
2.3.3 ASN.1 Authentication Information (AI) Attribute Definition	14
2.3.4 ASN.1 Authentication Attribute Certificate definition	15
3.0 Approximate Certificate Data Size	16

Appendix E: Public Key Infrastructure

1.0 Background	1
1.1 PKI System Components	2
1.1.1 Enrollment Station	2
1.1.2 Certificate Authority	2
1.1.3 Attribute Authority	3
1.1.4 ID Station	3
1.1.5 Workstation	3
2.0 A PKI Example	4
2.1 System Enrollment	4
2.2 Local Enrollment	5
2.2.1 Workstation Access	6
2.2.1.1 Domain Verification	7
2.2.1.2 Workstation Verification	8

Guidelines for Placing Biometrics in Smartcards

1.0 Scope

This document provides guidance to any U.S. Government agency utilizing smartcards and biometrics for access control and/or user authentication.

To obtain a secure implementation of smartcards and biometrics, it is necessary to discuss several system and implementation concerns. Without this discussion, the security benefits of using biometrics and smartcards as an access control and/or user authentication mechanism may be negated.

Throughout the document, biometrics may be discussed in its more basic type called “authentication information” (AI). AI is defined as any information that can be used to authenticate an entity (i.e. a user). AI may include such information as passwords, memory phrases, physical keys, etc. Since this document is to be used as a guide for integrating biometrics into smartcards, only a discussion of Biometrics will be given.

1.1 Purpose of this document

These guidelines are offered to complement those provided in the “Government Smart Card Technical Interoperability Guidelines.” The intent is to offer guidelines which will in facilitate a uniform authentication mechanism for U.S. Government Smart Cards and U.S. Government Smart Card applications in a secure manner.

2.0 Background

Smartcards and biometrics by themselves each provide a considerable boost to the Identification and Authentication (I&A) mechanism of any system. Together, they can provide a comprehensive solution of “who you are,” “what you have,” and “what you know.”

A common understanding of the underlying technologies is required to fully grasp how each component contributes toward a comprehensive I&A solution.

2.1 Automated Information Systems (AIS)

The use of biometrics and/or smartcards must work in tandem with some form of automated information system to meet a minimum level of assurance. Whether it is a workstation on a desk or an embedded system within a vending machine, strong user

authentication is based on proper integration of the separate components. Use of these systems require that it is trusted to perform the operations desired and only those specific operations. An example would be that a vending machine is expected to only debit a stored value application within a smartcard and not attempt to digitally sign legal documents. "Trust" in an AIS is earned when the AIS's functionality is perceived to be correct with respect to an established security policy. Use of a robust multi-application smartcard with the appropriate security features can help mitigate risk when utilizing an AIS of questionable origin.

There are several ways to establish different levels of trust in an AIS. One method is to use a Trusted Operating System (TOS). A TOS has been verified to perform correctly and if a failure occurs, it will fail safely, so that no restricted information is compromised. Verification methods of this trust include testing and formal mathematical analysis. Other less stringent methods to gain trust in a system can include physical isolation (no network or dial-up connections), purchasing products through trusted vendors, and of course physical security to prevent tampering.

The level of trust in an AIS required for a specific application is dependent on the value of the information at risk. An AIS restricting access to a classified room should not be connected to the Internet. Ensure that the platform used for your application really is "trustworthy".

2.2 Smartcards

A very basic definition of a smartcard is: a credit card sized plastic card that contains a microprocessor and provides some quantity of computing power and storage capacity.

Commercially, many applications have been devised for these devices, ranging from public transportation to electronic commerce. The most common government applications all derive from identification of individuals; whether they are employees or recipients of services or benefits.

In an age when personal computers have hard drives exceeding 10 GigaBytes, memory over 100 MegaBytes, and processing power exceeding 100 MIPS (million instructions per second), smartcards could be labeled as having limited resources. Smartcards are considered "state of the art" as they approach 32 KiloBytes of storage, 1 KiloByte of memory, and processing power in the range of 10 KIPS (thousand instructions per second).

For all of its limitations, the smartcard has sufficient resources to perform many of the tasks we wish to assign it, such as storage of identification information and the capability to perform basic cryptographic operations for a multitude of applications. For a more complete description of smartcards and where the technology is heading, see Reference[9].

2.3 Authentication Information (Biometrics)

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 trusted authentication information (AI).

Biometrics are methods of measuring the inherent physical attributes of an individual. This is usually performed in order to identify an individual or to verify a claimed identity. In the first case, a "livescan" is provided, and a database of templates is searched to determine who the scan is associated with. In the second case, a template is provided with the livescan for a direct comparison.

There exists many different types of biometric attributes to identify users. They may be based upon fingerprints, hand or facial geometry, retinal or iris patterns, or even speech recognition. Each of these technologies can be obtained from multiple sources, with different algorithms and techniques for storing an individual's features and/or comparing a "livescan" of an individual's features to the previously stored record. The stored record is typically referred to as the "Biometric Template". Biometrics can be best characterized as an emerging technology.

These various methods and data formats provide a challenge for those who would like to use multiple biometric systems or prevent themselves from becoming dependent upon a single proprietary solution. One solution is to wrap these proprietary interfaces and data formats with a standard interface or data format, much in the same way that the Internet uses IP (internet protocol) to wrap all of the various application data into standardized packets to provide seamless connectivity worldwide. This is the approach proposed, using certificates as the standard format.

2.4 Certificates

Certificates are portable blocks of data, arranged in a standardized format, which are often used to provide identification. X.509 is the ISO (International Organization for Standardization) standard for certificates. X.509 certificates can be separated into two basic categories: public key certificates and attribute certificates.

2.4.1 Public Key Certificates

In public key certificates, the primary piece of data being conveyed is a public key. The public key can be used for many things, such as mutual authentication, secure exchange of keys for setup of an encrypted link, and hashing to provide data integrity. There are many other data fields, including subject name, issuer, etc., but these are primarily to provide a context for the public key. Table 1 contains a minimized example of the contents of an X.509 public key certificate. Public key certificates are examined in more detail in Appendix D

Table 1: Contents of Public Key Certificate

Item	Description
version	The version of X.509 (currently equals 3)
serialNumber	The issuer provides each certificate it issues with a unique number
signature	The algorithm used by the issuer to sign the certificate
issuer	A description of the Authority who generated the certificate
validity	Start/Stop dates for this certificate
subject	Who was issued the certificate
subjectPublicKeyInfo	The public key
algorithmIdentifier	The algorithm associated with the public key
signatureValue	The issuer digitally signs the certificate to prove it is authentic

2.4.2 Attribute Certificates

Attribute certificates are much like public key certificates, with the difference being that instead of carrying a public key, the certificate's primary payload is some other form of data. One important detail is that the attribute certificate is linked to, or associated with, a public key certificate. A more complete description of attribute certificates can be found in Appendix D. Table 2 contains an example:

Table 2: Contents of Attribute Certificate

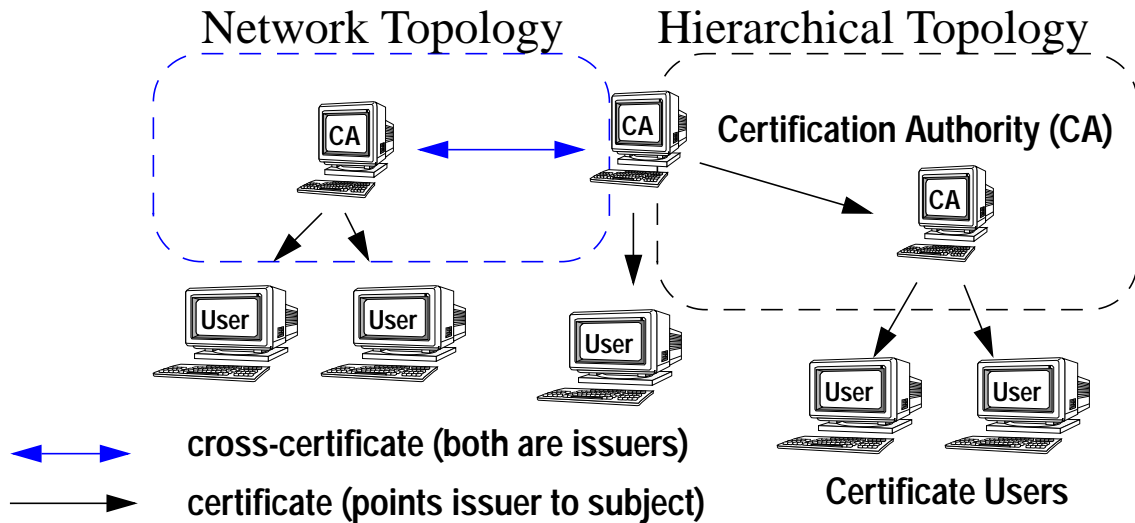
Item	Description
version	The version of the attribute certificate (currently equals 1)
owner (baseCertificateID)	Link to the public key certificate of the user
issuer (AA)	A description of the Authority who generated the certificate
signature	Identifies the algorithm used by the issuer to sign the certificate
serialNumber	The issuer provides each certificate it issues with a unique number
validity	Start/Stop dates for this certificate
Attributes	Data goes here
issuerUniqueID (Token Serial #)	This optional field can be used to tie the certificate to a smartcard by placing the card's serial number here.
algorithmIdentifier	The algorithm associated with the public key
signatureValue	The issuer digitally signs the certificate to prove it is authentic

2.5 Public Key Infrastructure (PKI)

The certificates introduced in Section 2.4, “Certificates,” were issued by an authority called an issuer. Who or what is an issuer? An issuer is a trusted entity within an organization acting in the role of a certificate authority (CA) or attribute authority (AA). A CA will issue public key certificates to users and/or subordinate CAs within its area of influence. An AA is subordinate to (or combined with) a CA and issues attribute certificates to users and/or systems within its area of influence. When these authorities are linked together in web or tree structures for interoperability, then there are the makings of an infrastructure. This infrastructure is based upon trust and strong cryptography.

Trust is transferred via certification paths in a PKI. A certification path is nothing more than a chain of certificates in which the signature on the previous certificate is verified by the public key of the next certificate until a certificate is reached which is trusted by the verifier. This is usually the CA of the verifier. This path will follow one of two topologies: hierarchical or network. In a hierarchical topology, all CA's are arranged under a 'root' CA that issues certificates to subordinate CA's, who may issue certificates to users or to subordinate CA's, and so on. This topology works best for government or military applications.

In a network topology, CA's cross certify each other. Each CA issues a certificate to the other. This means that each can place trust in the certificates issued by the other without both being subordinate to a common CA. This topology seems to be a better fit for the competitive marketplace.



3.0 The Need to Protect Authentication Information

The need to protect authentication information is a system-wide requirement. It is not limited to just the smartcard. Since smartcards will be lost or stolen, they are the most vulnerable component of the system. However, the transmission of the AI from the smartcard to the host, the memory inside the host, and any device/conduit which transmits and/or uses the AI needs to protect the information.

The following is a brief discussion of the vulnerability of the authentication information (AI). Appendices A through E contain further details on these discussion. The value of the information to be protected will drive which protection mechanisms need to be implemented in a specific application.

3.1 Why Protect AI?

As loss of information used to authenticate the user of the system (refer to Appendix A Section 2.3 on page 5 for more information) can result in an adversary (hacker, renegade employee, terrorist, etc.) being capable of the following:

- Signing information as the user.
- Decrypting sensitive information intended for the user.
- Viewing information in the authentication certificate.
- Gaining access to systems and resources which is being protected by the smartcard.

Clearly, the loss of AI can be detrimental to a system's security posture.

3.2 Why Protect the Smartcard?

If the smartcard is issued to every user of the system, there will be many more smartcards than any other component of the system. The smartcard will probably be carried with the user at all times. Lost and stolen smartcards are inevitable. The smartcard is therefore the easiest system component for an adversary to acquire. Since the smartcard will be the most accessible component of that system, it will most likely be targeted. Therefore there's a great need to protect the sensitive information on the card (i.e. the AI).

3.3 Why Protect Biometric Information?

If the biometric information (i.e biometric templates) placed in the smartcard is used as AI (for access to sensitive information), it needs to be protected.

The other main reason for protecting biometric information is public perception. The public may perceive biometric information as a confidential piece of information, much like a social security number. The use of biometric information as a means of access control and/or non-repudiation purposes is likely to enter into the privacy issue debate. Even if access to that information requires a lot of effort, it can lead to a public relations problem.

3.4 Steps to Secure Authentication Information

The guidelines beginning in Section 4.0 on page 8 will help protect authentication information. The highlights of those guidelines are:

- Define a system security policy that defines how users are to be certified to have access to the system.
- Define roles for the user.
- Define what information/functionality the roles allow users access to.
- Digitally sign the biometric template along with other identification, role, and privilege information to provide source authentication.
- Perform a mutual authentication protocol between the smartcard and the host prior to exchanging any authentication information.
- Encrypt the communication path between the host and the smartcard.
- Apply a protective coating to all smartcards.
- Encrypt all private keys stored in the smartcard.
- Seek NIST FIPS 140-1 certification on all smartcards used.
- Verify the signature on all certificates loaded into the smartcard which are used for authentication purposes.
- Enforce mutual authentication and biometric verification through a smartcard prior to granting access to a system. The system should only grant privileges based upon role information located within verified certificates.

4.0 Guidelines for Protecting Authentication Information

In Section 2.0, “Background,” three important technologies were presented: smartcards, biometrics, and certificates. The combination of these three can provide a comprehensive I&A capability which is not limited by specific vendor implementations of cards or biometrics. However, several specific guidelines should be met for protecting the authentication information.

4.1 System Consideration Guidelines

The security of any given system is only as good as its weakest link. Several steps can be taken to better define how the system acts and determine where the weakest link exists. Protecting the authentication information from the time its generated to the time its utilized should be the goal. The specific guidelines are as follows:

Table 3: System Guidelines

Reference Marker	Brief Description	Found in Appendix	Section
SYS1	The system security policy needs to establish roles and access privileges for those roles	Appendix A	2.4
SYS2	A risk analysis should be performed in order to determine the weak points of the system	Appendix A	2.4

These guidelines will help protect the information on the smartcard as well as the path between the smartcard and its AIS. Protecting the information while it is outside of the smartcard and securing the communications with the AIS is an important system concern.

4.2 SmartCard Guidelines

The Government Smart Card is responsible for storing and issuing the authentication information. Several steps should be taken to ensure that any sensitive information is protected while stored within the card as well as during any communications with an AIS:

Table 4: Guidelines For Biometric Capable Smartcards

Reference Marker	Brief Description	Found in Appendix	Section
SC1	A single cardlet (Smartcard Authentication Cardlet) should be developed to provide a universal authentication mechanism for the Government Smart Card.	Appendix B	1.2
SC2	A public key capable smartcard should be used.	Appendix B	1.3
SC3	At least 4.5 Kilobytes of EEprom space must be reserved for certificates. Note: This value is an estimate: refer to Appendix B for details.	Appendix B	2
SC4	The Token Authentication Protocol should be implemented to mutually authenticate the token to the AIS and to provide an encrypted channel between the AIS and the token	Appendix B	3
SC5	Protective coatings should be applied to the Government Smart Card to prevent physical attacks.	Appendix B	4
SC6	FIPS 140-1 certificate should be obtained on Government Smart Cards to a level 3 minimum	Appendix B	5
Cert2	The smartcard (Smartcard Authentication Cardlet) must verify the signature of all certificates loaded into it which are used for authentication purposes.	Appendix C	2.2
Cert3	The Smartcard Authentication Cardlet must be capable of PER decoding certificates	Appendix C	2.2

The intention is to utilize a smart card with enough resources to adequately implement the cryptographic and procedural functions necessary to authenticate users and protect the information used. Standardization on the cardlet and the certification of the card will help make the protection uniform between different vendors.

4.3 Biometric Guidelines

Each system designer must ultimately choose which standards-based or proprietary biometric products it will support. The guidelines given in this section are in support of making that choice. There is no criteria given in this section that will include or exclude a specific product. Refer to appendix C for a detailed discussion on making a choice of a biometric device. the guidelines for choosing a biometric device(s) are as follows:

Table 5: Biometric Guidelines

Reference Marker	Brief Description	Found in Appendix	Section
BIO1	Template Size: The size of the user’s biometric template must be small enough to fit into a smart-card. 500 bytes or less is the recommended target size.	Appendix C	2.3.1
BIO2	Multiple Templates: In order to reduce the False Rejection Rate (FRR), some biometric devices will require more than one template.	Appendix C	2.3.1
BIO2	The maximum processing time to scan, process and image and verify against a biometric template should be calculated for the target system. A value of 1 second is suggested.	Appendix C	2.3.2
BIO3	Because biometric products will change over time, upgrades should be automated over a period of time.	Appendix C	2.3.3

Keep in mind that biometric verification technology, as well as smartcard technology, is rapidly changing. Rejecting a biometric device solely because it has a template 50 bytes larger than another may be premature since the next version of the product could reduce that size or the next smartcard may have 8K bytes more EEprom.

4.4 Certificate Guidelines

The authentication information (AI) used by the system which utilizes the Government smartcard should require the AI (including biometric templates) to be placed in an authentication attribute certificate as described in Appendix C section 2.3.4 . This certificate should be placed in the smartcard when the user is enrolled in the system and issued the smartcard. The certificate can be retrieved by any system component (at a later time) to authenticate the user by verifying the signature of the certificate then proceeding to verify the authentication information via the means specified in the certificate (such as a fingerprint matching routine).

Table 6: Guidelines for an Authentication Attribute Certificate.

Reference Marker	Brief Description	Found in Appendix	Section
Cert1	All certificates used on the Government Smart Card should be encoded using Packed Encoding Rules (PER Encoding)	Appendix D	2.1
Cert4	The Authentication Attribute Certificate shall be encoded and processed in according with Appendix C and placed in the smartcard during enrollment	Appendix D	2.3.4

The intention is to create a definition of a single authentication certificate to be used by systems which utilize the Government SmartCard. The authentication certificate should be flexible enough to hold multiple vendor specific biometric templates (and/or any other type of AI) and associated controlling parameters in a single certificate. The certificate contains a digital signature, signed by an attribute authority at the time of enrollment, to provide an authentication mechanism for verifying the source of the original Biometric template used for verification.

4.5 Public Key Infrastructure Guidelines

The Public Key Infrastructure (PKI) will be considered by this section to include all system component which have a stake in creating, processing, or utilizing certificates used for authentication. The includes a certificate authority for signing public keys, an attribute authority for signing attribute certificates, an enrollment station for enrolling users of the system, and an identification station for verifying users of a system. The guidelines for the PKI are as follows:

Table 7: Guidelines For Public Key infrastructures

Reference Marker	Brief Description	Found in Appendix	Section
PKI1	There needs to be a PKI in place to support the Government Smart Card	Appendix E	1
PKI2	Enrollment of a user, including initialization of a smartcard should occur at a trusted (dedicated) workstation	Appendix E	1.1.1
PKI3	An attribute authority must be established to support the creation/maintenance of authentication certificates	Appendix E	1.1.3
PKI4	Verification of a user should occur at a trusted (dedicated) workstation	Appendix E	1.1.4

The unknown risk of running other applications on these system component leads to the recommendation of dedicated workstations. Since there are probably few certificate authorities, attribute authorities, and enrollment stations, it is probably a reasonable guideline for those components. Since it is unknown how many identification stations will be used, the impact this guideline will have on the total cost of a given system cannot be given.

5.0 Conclusions

The secure linkage sought between smartcards and biometrics is made possible through the use of attribute certificates and public key cryptography. The utilization of a vendor-neutral data interchange format improves the opportunity for widespread adoption of these technologies in a manner that allows for systems to upgrade with technology. It also provides the opportunity of choosing between multiple vendors irrespective of previous product purchases.

6.0 References

- [1] U.S. General Services Administration Office of Government Policy. "Government Smart Card Technical Interoperability Guidelines",
- [2] ISO. WD15782-2. "Banking - Certificate Management Part 2: Attribute Certificates." October 28, 1997.
- [3] ISO/IEC. ISO/IEC 9594-6, "Information technology - Open Systems Interconnection - The Directory: Selected attribute types." 1995-09-15.
- [4] ISO/IEC 7816-4, "Identification Cards - Integrated circuit(s) cards with contacts - Part 4: Interindustry commands for interchange." 1 September 1995.
- [5] Luther, S. and L. Reinert. "TOKENNEER Authentication Protocol for Smartcards," version 1.0, January 23, 1998.
- [6] Reinert, L. and S. Luther. "User Authentication Techniques Using Public Key Certificates Part 1: Certificate Options." December 24, 1997
- [7] Reinert, L. and S. Luther. "User Authentication Techniques Using Public Key Certificates Part 2: Authentication Information Including Biometrics." December 31, 1997.
- [8] Reinert, L. and S. Luther. "User Authentication Techniques Using Public Key Certificates Part 3: An Example Implementation." December 10, 1998.
- [9] Bury, Lawrence J. "Smartcards: A Technology Forecast." July 1998.
- [10] Anderson, Ross and Markus Kuhn. "Tamper Resistance - a Cautionary Note." *The Second USENIX Workshop on Electronic Commerce Proceedings*. Oakland, CA November 18-21 1996, pp1-11, ISBN 1-880446-83-9
- [11] NIST. FIPS Pub 196, "Entity Authentication Using Public Key Cryptography." February 18, 1997.
- [12] NSA. R223. "O'er the RAMparts We Watch: An Introduction to the Risk Analysis Model (RAM)." 21 February 1996.
- [13] DOD. DOD 5200.28-STD. "Department of Defense Trusted Computer System Evaluation Criteria ('The Orange Book')." December 1985.
- [14] NIST. FIPS Pub 140-1, "Security Requirements For Cryptographic Modules." January 11, 1994.
- [15] Weldon, Tommy D. "Biometrics: A Technology Forecast." July 1998.
- [16] ISO/IEC. ISO/IEC 9594-8, "Information technology - Open Systems Interconnection - The Directory: Authentication framework." 06/97.
- [17] ISO/IEC. ISO/IEC 9594-2, "Information technology - Open Systems Interconnection - The Directory: Models." 1995.
- [18] ECMA. ECMA-219, "Authentication and Privilege Attribute Security Application with related key destruction functions." second edition, 1996.
- [19] NIST. FIPS Pub 190, "Guideline for the use of Advanced Authentication Technology Alternatives." September 28, 1994.
- [20] Willis. David. "Let Your Fingers Do the Logging In", *Network Computing Online*, July 6, 1998.

Note: references 4 - 9, 12, 15 are available from the National Security Agency's Office of INFOSEC Research and Technology.

Appendix A: System Considerations

1.0 Background

1.1 Threat Definition

Before attempting to discuss the details of implementation, it is important to define the data the system is trying to protect, the usage of that data, and from whom you are trying to protect it against. This helps the system designers in making critical trade-offs between security and other factors such as cost, complexity, and system performance.

One method which assists in making the security trade-offs decisions is the use of the Risk Analysis Model (RAM)[12]. The RAM explores vulnerabilities by creating “event trees” with probabilities of success assigned to each event. By doing this, one can play “What if” scenarios to determine how the adversary’s probability of success changes with system protection mechanism. This gives the system designer a powerful tool in justifying security costs by showing decline in the adversary’s probability of success. The RAM does require a significant amount of effort, but the results are well worth it.

The following definitions are required for the RAM analysis. Even if a RAM analysis is not performed, the definitions help in the consideration of applying security measures to any given system.

1.1.1 Define the Data you are protecting

It is important to define the data you are protecting, and its value to you and your adversary. The value of the data to the adversary will determine how many resources he will be willing to place against obtaining the data. The value to you will determine how much you will pay to protect the data. To determine the value, ask the following questions:

- What happens if the data is stolen?
- What happens if the data is lost?
- What happens if the data is duplicated by an untrusted source?
- What happens if access is given to the data by an unauthorized entity?

1.1.2 Define Your Adversary

The following definitions are the currently accepted taxonomy for attackers (the adversary) [10]:

- Class I (clever outsiders): They are often very intelligent but may have insufficient knowledge of the system. They may have access to only moderately sophisticated equipment. They often try to take advantage of an existing weakness in the system, rather than to create one.
- Class II (knowledgeable insiders): They have substantial specialized technical education and experience. They have varying degrees of understanding of parts of the system but potentially access to most of it. They often have highly sophisticated tools and instruments for analysis.

- Class III (funded organizations): They are able to assemble teams of specialists with related and complementary skills backed by great funding resources. They are capable of in depth analysis of the system, designing sophisticated attacks, and using the most advanced analysis tools. They may use Class II adversaries as part of the attack team.

1.1.3 Define the System Security Policy

The Security Policy is defined by the Orange Book as “The set of laws, rules, and practices that regulate how an organization manages, protects, and distributes sensitive information”[13]. The system targeted by the adversary can be ranked by the robustness of the policy and how well the users of the system adhere to the policy. The ranking is as follows:

- **HIGH:** A security policy exists which limits the vulnerabilities created by users of the system. The policy is well maintained and updated on a periodic basis. Users of the system receive training and update information about the policy. Certain users may require certification and background checks as needed. Automated techniques are used to enforce and verify that the policy is adhered to. Components of the system must be located in physically protected facilities (i.e. a building with an electronic security system with 24 hour guards posted at the entrance) such that no unauthorized people have physical access to the system.
- **MEDIUM:** A security policy exists which was created at the time the system was installed, but not very well maintained. User may have knowledge of the policy but are not trained and do not receive updates. User adherence to the policy is ad-hoc and not enforced.
- **LOW:** There exist no security policy for the system, or users have no knowledge of it. Only default security measures are installed in the system and the users only use these when they are forced to. Security measures can be disabled by the user if they are a burden, and are usually set off.

1.2 Trust

The following definitions are found in the “Glossary of Computer Security Terms”, reference [13].

Trusted Computing Base (TCB): The totality of protection mechanisms within a computer system, including hardware, firmware, and software, the combination of which is responsible for enforcing a security policy. A TCB consists of one or more components that together enforce a unified security policy over a product or system. The ability of a TCB to enforce correctly a unified security policy depends solely on the mechanisms within the TCB and on the correct input by system administrative personnel of parameters (e.g., a user's clearance level) related to the security policy.

Trusted Path: A mechanism by which a person at a terminal can communicate directly with the TCB. This mechanism can only be activated by the person or the TCB and cannot be imitated by untrusted software.

Trusted Process: A process whose incorrect or malicious execution is capable of violating system security policy.

“Trust” will be used in this document to denote that an entity, to which the term is being applied, can be expected to, within normal operating procedures, perform its given tasks without a significant risk of manipulation by a malicious source. This implies that the operations which it performs are well known at the time of its design and that its design has passed an evaluation by an accredited security analysis.

An entity can be trusted as a whole, or sub-portions of the entity can be trusted. Whole trusted components must be special purpose entities since their operations must be evaluated prior to use. If only portions of the entity are trusted, then only operations which are performed on those trusted components can be trusted.

System designers must specify which entities within the system are trusted and which are not. Biometric and smartcard enrollment and verification must be performed on a trusted platform, or the integrity of those operations can be impaired.

1.3 User Confidentiality

The X.509 certificate is 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 public 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.

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 [11] for further information on mutual verification.

1.4 Need To Know

Sensitive information is information to which access is based on a “need to know” concept. Only system components which have a need to know should have access to that information. Certain sensitive information (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.

1.5 Processing Time

The amount of time it takes to process the signature verification may be of concern to the system. The 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.

In systems where I/O throughput is a factor (especially in smartcard based system where I/O may be limited to 9600 baud/second), the data size of the certificate may be a factor. 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 become a detriment to overall system performance if multiple certificates are required.

1.6 System Architecture

The system components needed to manage the certificates may increment 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.

1.7 Storage

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).

1.8 Export Restrictions

If you plan to send these systems overseas, then you must consider export restrictions that fall under the U.S. State Department's national security and criminal control statutes.

2.0 System risks due to the smartcard

2.1 Defining the data on the smartcard

The specific sensitive data placed on the Government Smartcard will consist of:

- Private signature key of the user.
- Private key exchange key of the user
- The authentication certificate
- Other sensitive information such as account balances, security codes, etc.

2.2 How can the data be lost?

For the private keys:

- Physical attack on the smartcard
- Incorrect implementation of an algorithm
- Back doors and implementation flaws due to poorly designed/test implementation of the smartcard.

Most smartcards do not allow private keys to be obtained directly from the interface. The most likely way of obtaining the private key is via a physical attack. Refer to Appendix B section 1.1.1 for more information on smartcard attacks.

For the AI certificate:

- A physical attack on the smartcard
- Placing a “Trojan Horse” application in the host PC to capture I/O information
- Tapping the line between the host and the smartcard
- Providing a bogus host to capture the information from the smartcard.

2.3 What happens if the Government Smartcard is lost?

If the adversary gains access to the smartcard, it is possible for them to:

- Sign information as the user
- Decrypt sensitive information intended for the user
- View information in the authentication certificate
- Gain access to the system and its resources which is being protected by the smartcard.

Since the keys placed on the smartcard will be used to protect other information within the system, the effect of an adversary exploiting the data on the smartcard could potentially be detrimental. The adversary could gain access to system resources resulting in financial loss, military intelligence, proprietary data, etc.

Although the signature on the authentication certificate will prevent the adversary from easily

changing data, the authentication certificate can still be used to masquerade as the user if the authentication mechanism is easily circumvented with the information in the certificate (i.e. if the AI was a password the adversary is “in”). The use of biometrics as AI increases the difficulty of exploiting the information, however they have varying degrees of difficulty in circumvention. More than likely, if this information is easily obtainable, the result will be a loss of confidence by the user. Biometric information may be sensitive in terms of privacy issues. The loss of confidence in the system to provide confidentiality may be more detrimental than an actual success measured by an adversary.

2.4 What can be done to prevent data from being lost or stolen?

The following steps should be taken by the system designers to prevent the theft of data from the smartcard:

- Create a security policy with roles defined and access privileges for those roles. This system policy should address all the points in this section. [SYS1]
- Define each component of the system and determine if it is trusted. Define all interactions between the smartcard and each component of the system.
- Define all the data stored on the smartcard and the its association for information used by the system.
- Perform a risk analysis on the system to determine the weak points and define possible solutions[SYS2].

The following steps can be taken to prevent data from being stolen from the smartcard:

- Certify the smartcard to FIPS140-1 level 3 minimum. This will screen out implementation flaws and provide necessary security features (refer to Appendix B section 5.0 “NIST Certification”).
- Protective coatings on the smartcard (refer to Appendix B section 4.0 “Physical Protection”).
- Perform a mutual verification between the host and the smartcard prior to any other operations (refer to Appendix B section 1.4.2 “Mutual Authentication”).
- Encrypt sensitive data while the card is inactive (refer to Appendix B section 1.4.1 “Lock/Unlock”).
- Encrypt I/O between host and the smartcard (refer to Appendix B section 3.0 “Token Authentication Protocol (TAP)”).

Appendix B: Smartcards

1.0 Background

1.1 Vulnerabilities

Millions of users will be given the government smartcard. Lost or stolen smartcards falling into the hands of a knowledgeable adversary are almost a certainty. These adversaries will be doing everything possible to extract the contents of the smartcard (private keys and/or authentication certificates) in order to exploit the system for financial, political, or military gain.

1.1.1 Physical attacks on the smartcard

Several physical attacks (attacks which require physical possession of smartcard by the adversary and generally require physical access to the smartcard chip) have been publicly documented [10] and are available on the internet. These attacks include reading the contents of the EEPROM on the smartcard chip, applying reverse engineering techniques, and use of a focused ion beam. Most commercially available smartcards are susceptible to these attacks.

1.1.2 Attacks between the host and the smartcard

Probing the communications path between the host and the smartcard can give the adversary knowledge of all unencrypted variables. Even public information (such as public keys) can give the adversary necessary data to perform attacks. Securing the path between the host and the smartcard is clearly important, especially for contactless cards since the information transmitted is vulnerable to passive eavesdropping.

1.1.3 Steps to prevent the attacks

Several steps can be taken to resist an attack on a smartcard. These steps should contain:

- Applying a physical coating to the smartcard chip to prevent various physical attacks.
- Using tamper-evident card to hold the ICC.
- Using a token authentication protocol which mutually authenticates the host to the smartcard and encrypts the communication path between the host and the smartcard

1.2 Multi-Application Smartcards

The future of smartcards revolves around the “multi-application” smartcard. This may lead to different “cardlets” (applications running on the smartcard) utilizing different authentication methods. For that reason, it is suggested one cardlet be developed to provide session based authentication methods[SC1]. Other aspects of multi application cards to consider are:

- A certification method such as FIPS 140-1 certification.
- Signature verification on all downloaded cardlets. This prevents unauthorized cardlets from being downloaded to the smartcard.
- Secure file sharing capabilities.

1.3 Cryptographic Algorithms

Protection of the communication path between the host and the smartcard requires encryption. Public key algorithms offer a mechanism to provide an encrypted channel without the exposure of a private key. Therefore the smartcard should contain a public key capability via a key exchange algorithm[SC2]. This document will not specify or endorse algorithms.

1.3.1 Cryptographic Co-processors

Smartcards with cryptographic co-processors are generally required for most public key based algorithms. However, with the advent of elliptic curve cryptography, co-processors may no longer be required for adequate performance. The selection of a smartcard with a cryptographic co-processor is a system specific decision which takes into the account the choice of the algorithm, the desired system performance, and the cost of the smartcard.

1.4 Overview of token based security mechanisms

The smartcard must be trusted to perform correctly and to be relatively impervious to physical (tamper) attack. As smartcard vendors do not disclose the anti-tamper protection provided to their products, there is a level of risk associated with the use of these products. The Lock/Unlock protocol uses a cryptographic technique to prevent an adversary from obtaining a user's private key via tamper attacks. This method will be described briefly, but for a more detailed explanation refer to "TOKENNEER Authentication Protocol for Smartcards" [5]. The token should only share sensitive user information with hosts which are authorized to receive it. Through mutual authentication, the token can be assured it is dealing with an authorized host while the host can be assured it is dealing with a properly registered token. Finally, for the purpose of securing the (possibly vulnerable) link between the token and the host, the session can be protected by the secure exchange of a traffic encryption key.

1.4.1 Lock/Unlock

The primary purpose of the Lock/Unlock protocol is to protect the user's private signature key (which never leaves the token) from attack. This is accomplished by splitting the private key into two values, and then encrypting one of the two values with the public key of another (trusted) entity. The private key can then only be reclaimed by taking the token to that entity to have the one half of the key decrypted. The key will be split using a random number generator, thereby preventing the trusted entity from obtaining insight into the value of the key. Once the token has reconstructed the key, it generates another split so that an attacker could not perform a replay attack. For this scheme to be successful, there must be a requirement that the private key is never stored in the token's persistent storage (such as EEPROM or Flash memory). It must only be stored in RAM so that it will not remain present once the token is removed from the reader. For a complete description of this protocol refer to [5].

The simplified protocol without any nested enclaves is presented in Table 1 on page 3. As can be seen, the Lock/Unlock protocol requires the additional storage of the public key of the root key at

all times and at various times the public key of each enclave entered. (Note: a designated authority (possibly the corporate Certificate Authority (CA)) is the root of the lock/unlock hierarchy. This is not the same entity as the root of the CA hierarchy). This places additional storage pressures on the token. Also note that a significant number of objects are identified as residing in RAM only. This may cause significant difficulties in locating a token with sufficient RAM, especially if a commercial smartcard based solution is desired.

Table 1: Lock/Unlock

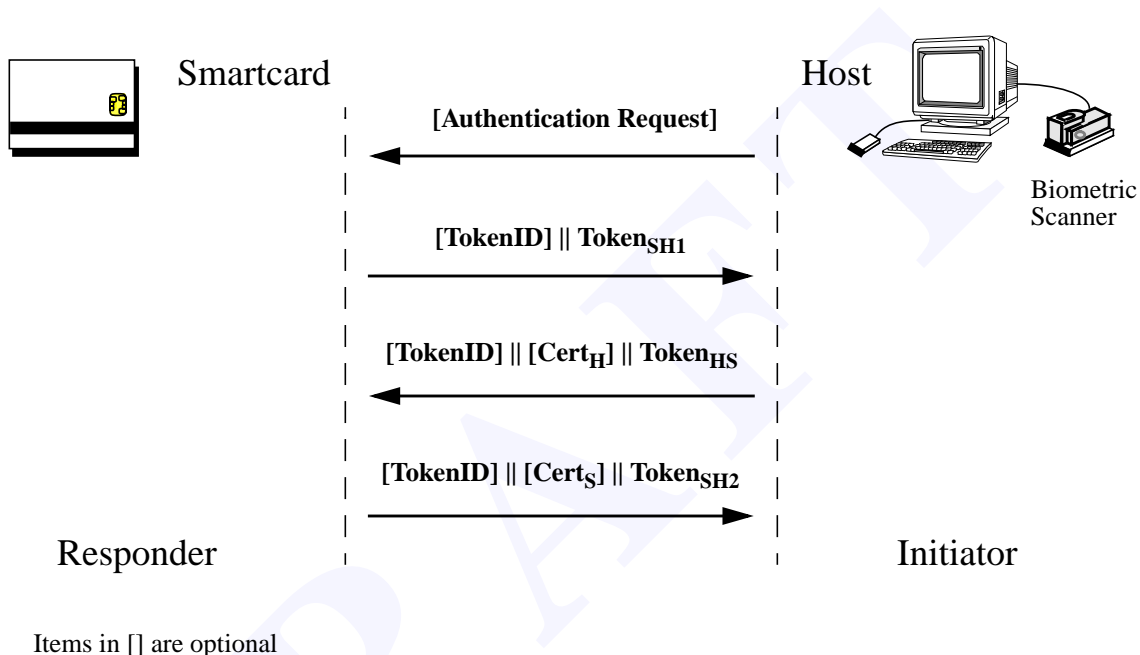
Event	Token		Exchange Data	Host
	EEPROM	RAM		
Outside protected boundary	$\alpha_1, [\beta_1]^{root}, Y_{root}$	--	--	--
Enter enclave	$\alpha_1, [\beta_1]^{root}, Y_{root}$ $\alpha_2, [\beta_2]^{enclave}, Y_{enclave}$	$x = \alpha_1 + \beta_1$ $\alpha_2 = \alpha_1 + \delta(\alpha_1) + \delta(\beta_1)$ $\beta_2 = \beta_1 + \delta(\alpha_1) + \delta(\beta_1)$ $[\beta_2]^{enclave} =$ $PKE(\beta_2, Y_{enclave})$	--> $[\beta_1]^{root}$ <-- β_1 $Y_{enclave}$	$X_{root}, Y_{enclave}$ $\beta_1 = PKD([\beta_1]^{root}, X_{root})$
Inside enclave	$\alpha_1, [\beta_1]^{root}, Y_{root}$ $\alpha_2, [\beta_2]^{enclave}, Y_{enclave}$	--	--	--
Log-on at workstation	$\alpha_1, [\beta_1]^{root}, Y_{root}$ $\alpha_2, [\beta_2]^{enclave}, Y_{enclave}$	α_3 $x = \alpha_2 + \beta_2$ $\alpha_2 = \alpha_2 + \delta(\alpha_2) + \delta(\beta_2)$ $\beta_2 = \beta_2 + \delta(\alpha_2) + \delta(\beta_2)$ $[\beta_2]^{enclave} =$ $PKE(\beta_2, Y_{enclave})$	--> $[\beta_2]^{enclave}$ <-- β_2	$X_{enclave}$ $\beta_2 = PKD([\beta_2]^{enclave}, X_{enclave})$
Exit workstation, inside enclave (optional)	$\alpha_1, [\beta_1]^{root}, Y_{root}$ $\alpha_2, [\beta_2]^{enclave}, Y_{enclave}$	--	--	--
Exit enclave, outside protected boundary	$\alpha_1, [\beta_1]^{root}, Y_{root}$	--	--	--

PKD(A, B) = public key decrypt A with key B PKE(A, B) = public key encrypt A with key B
 $[\beta_1]^{root} = \beta_1$ is encrypted with root's private key Y_{root} = root's public key X_{root} = root's private key
 x = the token's private key α_1 & β_1 = components used to recover the token's private key

1.4.2 Mutual Authentication

NIST FIPS 196 [11] specifies two challenge-response protocols by which entities may authenticate their identities to one another. The first protocol, for unilateral authentication, will not be addressed. The second protocol, for mutual authentication, does meet the needs of this system as it provides the token, the trusted and untrusted hosts with the confidence they require to know with whom they are communicating.

Mutual Authentication Protocol



The mutual authentication protocol consists of up to six steps. The first (optional) step is for the entity deemed the initiator (the Host in the diagram above) to send a request for authentication to the responder. The format of the request is not defined in the standard. In step 2, the responder (the Smartcard in the diagram above) then determines whether it will initiate (if step one is not used), continue or terminate the exchange. If it initiates or continues, it creates a data packet consisting of the following elements:

- **R_S**: Random number challenge
- **Text₁**: Other data to be appended to the random number challenge and considered to be a part of the challenge token itself, denoted **Token_{SH1} = R_S || [Text₁] [optional]**
- **TokenID**: a token identifier, which is information used to identify the challenge token and assist token processing by the recipient [optional]

In step 3 of the protocol, the initiator will receive the data packet, and perform the following steps:

1. use **TokenID** to determine what is being received [optional]
2. retrieve information from the optional data field [optional]

3. generate a new random number challenge: R_H
4. create an identifier for the responder: S [optional]
5. generate an authentication token to send to the responder: $Token_{HS}$

$Token_{HS}$ contains of the following items:

- R_S, R_H, S
- $Text_2$ and $Text_3$: $Text_2$ is a subset of $Text_3$, including cases where $Text_2$ is NULL and $Text_2 = Text_3$.

$Token_{HS}$ is generated by concatenating this data and generating a digital signature: $Token_{HS} = R_H \parallel [R_S] \parallel [S] \parallel [Text_3] \parallel s_{S_H}(R_H \parallel R_S \parallel [S] \parallel [Text_2])$ This value is then [optionally] concatenated to $TokenID$ and the Certificate or entire certificate path (denoted $Cert_H$) of the initiator.

Upon receiving the $Token_{HS}$ transmission, in step 4 the responder will:

1. use $TokenID$ to determine what is being received [optional]
2. if R_S is a part of $Token_{HS}$, verify that R_S is the same value that was originally sent out
3. verify that the identifier for the initiator has been obtained through either: $Cert_H, Token_{HS}$, or the step 1 authentication request. For this system, the identifier MUST contain the initiator's public key
4. verify the initiator's certificate or certificate chain
5. verify the initiator's signature in $Token_{HS}$
6. generate an authentication token to send to the responder: $Token_{HS}$
7. retrieve data from the $S, Text_2$ or $Text_3$ fields. This is not addressed further in the standard

At the successful completion of step 4, the Host has been authenticated to the Smartcard. In step 5, the Smartcard prepares a data packet so that it can be verified by the Host by completing the following actions:

8. create an identifier for the initiator: H [optional]
9. generate an authentication token to send to the responder: $Token_{SH2}$

$Token_{SH2}$ contains the following items:

- R_H, R_S, H
- $Text_4$ and $Text_5$: $Text_4$ is a subset of $Text_5$, including cases where $Text_4$ is NULL and $Text_4 = Text_5$.

$Token_{SH2}$ is generated by concatenating this data and generating a digital signature: $Token_{SH2} = [R_S] \parallel [R_H] \parallel [H] \parallel [Text_5] \parallel s_{S_S}(R_S \parallel R_H \parallel [H] \parallel [Text_4])$ This value is then [optionally] concatenated to $TokenID$ and the certificate or entire certificate path (denoted $Cert_S$) of the responder.

Upon receiving the **Token_{SH2}** transmission, in step 6 the initiator will:

1. use **TokenID** to determine what is being received [optional]
2. if **R_A** is a part of **Token_{SH2}**, verify that **R_H** is the same value that was originally sent out
3. if **R_B** is a part of **Token_{SH2}**, verify that **R_S** is the same value that was received in **Token_{SH1}**
4. verify that the identifier for the initiator has been obtained through either: **Cert_H**, **Token_{SH1}**, or **Token_{SH2}**. For this system, the identifier **MUST** contain the responder's public key
5. verify the initiator's certificate or certificate chain
6. verify the initiator's signature in **Token_{SH2}**
7. retrieve data from the **S**, **Text₄** or **Text₅** fields. This is not addressed further in the standard

At the successful completion of step 6, the Smartcard has been authenticated to the Host, and the entities have successfully mutually authenticated.

1.5 Confidentiality through Encryption

Systems will be sharing potentially sensitive information with the hosts it attempts to authenticate. To prevent this information from being compromised while in transit between the token and host, the information can be encrypted. Commercial smartcard's DES secret key encryption can be used. However, there must be a method to securely transmit the key to be used. The secure exchange of a traffic encryption key can be performed through a public key algorithm such as RSA public key encryption. RSA algorithms are typically available on commercial smartcards.

2.0 Certificate Storage Requirements

The smartcard will be required to hold and process a signature certificate for each Certificate Authority (CA) in the public key infrastructure (PKI) hierarchy, and each of the certificates associated with its user. The following assumptions are going to be used for the storage estimate. Refer to Appendix C section 3.0 “Approximate Certificate Data Size” for details on the certificate component sizes used.

- A 512 bit public key algorithm is used
- The average biometric template is 500 octets
- PER encoding is used
- 4 levels of CA exist in the PKI hierarchy

Table 2:

Certificate	Use	Approximate size	Notes
User PK Signature Certificate	Used by other entities to verify the signature created by the users smartcard	579 octets	
User Key Exchange Certificate	Used by other entities to encrypt the data for the user.	579 octets	
User Authentication Information Certificate	Used by other entities to verify the user.	838 octets	Could be used by the smartcard if it performs biometric authentication
Root CA Signature Certificate	Used by the smartcard to verify other certificates.	579 octets*4= 2316 octets	If there is a hierarchy of CAs less then 4 reduce this estimate
Total		4312 octets	This approximate amount can change if the number of CA increases or a different algorithm is used.

This implies that a little over 4K [SC3] bytes of storage is needed just for the certificates used by the smartcard.

Cardlet space is not included in this estimate. There is currently no estimation methods for determine the size of the cardlets used. The individual system will need to determine the storage requirements for the cardlets it intends to utilize and the data storage required for them and add that number to certificate storage requirements.

3.0 Token Authentication Protocol (TAP)

In order to maximize the efficiency of the processing power and I/O throughput of the smartcard, it is necessary to create an adaptation of the FIPS Pub 196 defined Mutual Authentication Protocol to utilize public key and secret key encryption versus digital signatures [SC4]. This will avoid the natural repetition which occurs if each protocol is implemented separately. The optimization has produced one potential vulnerability: The encrypted β (the split of the smartcard's private key) is shared with the host prior to the host completing its authentication to the smartcard. If this issue requires further resolution, the logical solution would be to insert an additional challenge and reply after Step 1 of the protocol.

The Token Authentication Protocol (TAP) combines the necessary aspects of Lock/Unlock, FIPS Pub 196, RSA public key encryption, and DES secret key encryption. The protocol has been generalized so that it is adaptable to meet various architectures and alternative cryptographic algorithms.

The protocol is as follows:

3.1 Step 1

The host, H, assembles a data packet (called the **Authentication Request**) which consists of its own public key certificate (**Cert_H**) and a list of identifiers for each attribute certificate type (**ACertID₁ || ... || ACertID_N**) required for the user to be authenticated. This request is forwarded to the smartcard, S, for verification.

$$\text{Authentication Request} = \text{Cert}_H \parallel \text{ACertID}_1 \parallel \dots \parallel \text{ACertID}_N$$

NOTE: The Enrollment station and ID Station will request the Privilege Certificate and IDA Certificate. The workstation will request the Privilege Certificate only.

3.2 Step 2

The smartcard examines **Cert_H** and determines if it has a corresponding beta so that the card may be unlocked and the private key recovered. If there is no match, the card returns an error message of SW1 = 69 (command not allowed) SW2 = 82 (security status not satisfied). If there is a match, the card will then generate a random number, **Rand_S**, store it to static memory, and encrypt it with the host's public key. The card will then evaluate the list of attribute certificate identifiers, and compile a list of those which it is willing to share with the host, H. It assembles a data packet, **Token_{SH1}**, and sends it to the host. **Token_{SH1}** consists of:

$$\text{Token}_{SH1} = \text{RSA}[\text{Rand}_S]^H \parallel \text{Cert}_S \parallel \text{RSA}[\beta]^H \parallel (\text{ACertID}_i \parallel \dots \parallel \text{ACertID}_j)$$

3.3 Step 3

The host, H, verifies the card's public key. If verification is successful, H examines the list of attribute certificates the card is willing to share and determines if it will continue the exchange. If it does, H decrypts \mathbf{Rand}_S and stores it for safekeeping. It then decrypts $\mathbf{RSA}[\beta]^H$. The host generates its own random number, \mathbf{Rand}_H , and encrypts it with the smartcard's public key. If the host is an ID station, it will locate the public certificate for the next lower domain, \mathbf{Cert}_X . H will then generate the next data packet, \mathbf{Token}_{HS} , to be sent to the card:

$$\mathbf{Token}_{HS} = \mathbf{RSA}[\mathbf{Rand}_H]^S \parallel \beta \parallel \mathbf{Cert}_X$$

3.4 Step 4

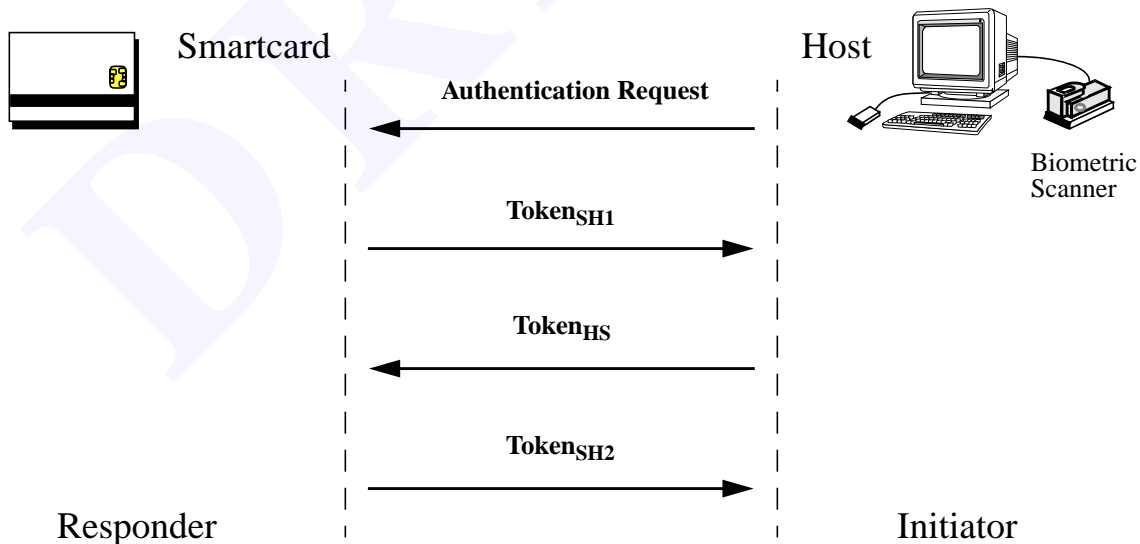
The smartcard combines β with α to generate its private key. If this operation is successful, the host has successfully identified itself to the card. Next, the card creates new β and α values and encrypts all β 's with the public key from the appropriate certificate (\mathbf{Cert}_X or \mathbf{Cert}_H). The card then decrypts \mathbf{Rand}_H and combines it with \mathbf{Rand}_S to create the DES key. The card then sends the attribute certificates to the host (encrypted) as shown:

$$\mathbf{KEY} = \mathbf{Rand}_S + \mathbf{Rand}_H$$

$$\mathbf{Token}_{SH2} = \mathbf{DES}[\mathbf{Rand}_H \parallel \mathbf{ACert}_1 \parallel \dots \parallel \mathbf{ACert}_N]^{\mathbf{KEY}}$$

If the host is able to decrypt \mathbf{Token}_{SH2} and extract \mathbf{Rand}_H , then the smartcard has been successfully verified to the host.

Token Authentication Protocol



$$\mathbf{Authentication\ Request} = \mathbf{ACertID}_1 \parallel \dots \parallel \mathbf{ACertID}_N \parallel \mathbf{Cert}_H$$

$$\mathbf{Token}_{SH1} = \mathbf{RSA}[\mathbf{Rand}_S]^H \parallel \mathbf{Cert}_S \parallel \mathbf{RSA}[\beta]^H \parallel (\mathbf{ACertID}_1 \parallel \dots \parallel \mathbf{ACertID}_j)$$

$$\mathbf{Token}_{HS} = \mathbf{RSA}[\mathbf{Rand}_H]^S \parallel \beta \parallel \mathbf{Cert}_X$$

$$\mathbf{Token}_{SH2} = \mathbf{DES}[\mathbf{Rand}_H \parallel \mathbf{ACert}_1 \parallel \dots \parallel \mathbf{ACert}_N]^{\mathbf{KEY}}$$

$$\mathbf{KEY} = \mathbf{Rand}_S + \mathbf{Rand}_H$$

4.0 Physical Protection

Physical protection for smartcards can be placed on the chip itself (ICCs) [SC5]. The coatings can prevent certain physical attacks on a lost or stolen card. These coatings can be applied in a cost effective means prior to mounting it on the card. Such coatings can help a smartcard achieve FIPS 140-1 level 2 certification.

An example of such a coating is Dow Corning's ChipSeal. It is a commercially available process for applying a protective coating to the bare die that is mounted in the smartcard.

DRAFT

5.0 NIST Certification

The National Institute of Standards and Technology (NIST) published the “Security Requirements for Cryptographic Modules” (FIPS PUB 140-1 [14]). This publication describes 4 security levels:

Table 3:

Level	Brief description
1 (lowest level)	Use of a NIST approved algorithm Most smartcards can meet this requirement
2	Adds tamper evident coatings or seals Adds role based authentication requirement Most smartcards can meet this with a physical coating on the chip
3	Enhanced tamper protection Adds identity based authentication requirement Requires protection of output ports Allows for multi user time shared system Most smartcards can meet this if the Token Authentication Protocol is implemented.
4 (Highest Level)	Adds tamper detection protects against environmental conditions for voltage and temperature.

It is recommended that all smartcards which utilize authentication certificates be certified to level 3. The rationale for a level 3 certification requirement is as follows [SC6]:

- A system is only as strong as its weakest link. Without any certification policy to enforce physical protection, the potentially large numbers of vendors could produce one or two easily exploitable implementations (via a design/lack of testing flaw). Standard validations may prevent this.
- Smartcards are easily lost or stolen. Physical attacks to extract private keys and authentication information can be easily obtained without protective measures. Level 4 provides strong protection against such attacks.
- Smartcards are susceptible to voltage/temperature attacks such as the theoretical Bell labs microwave attack. Level 4 certification may prevent such attacks.
- Protecting the link between the host and the smartcard is critical in providing confidentiality of the authentication information between the host and the smartcard. This is especially true of contactless cards where the data can be easily captured via undetectable means. Level 3 requires such protection.

Appendix C: Biometrics

1.0 Background

Biometrics has long been recognized as an effective and highly accurate way of determining the identity of individuals. Already used by government agencies and others for physical security, immigration control, prisoner/parolee control, and to minimize fraud in benefits programs, biometrics are now being developed and marketed as computer/network security technologies. If the biometrics industry is to grow significantly beyond its 1997 U.S. market value of approximately \$25 million, it will do so in the area of computer/network security.

There are many different biometrics in use today and several others under investigation or development for identification and authentication. Each of these physiological or behavioral characteristics has its own set of strengths and weaknesses. Some are too intrusive to be accepted by the general user population while others may not afford the high degree of accuracy required in some applications. The biometrics which seem best suited for the new information security applications are fingerprinting, voice recognition, and face recognition. Either singularly or in combination, these biometrics are likely to dominate the portion of the computer/network identification and authentication devices market captured by the biometrics industry.

Fingerprint recognition technology is already a key player in the information security device market. The most noteworthy characteristics of fingerprint biometric technology is the template size is typically under 500 bytes. The adaptability of fingerprint recognition hardware to the computer keyboard and mouse makes it a viable alternative to the workstation password. New optical components like molded lens and single chip fingerprint imaging devices have lowered hardware costs from the thousands dollars per unit to under fifty. Improved techniques under development by a number of different companies are designed to eliminate many of the shortcomings of current technology and will likely lead to even better performance by finger scanning devices.

Which biometric will eventually gain preeminence in the computer/network identification and authentication devices market will depend on a number of factors including cost, technological improvements that provide for improved performance, and consumer acceptance. Hand geometry devices have template sizes in the 10 - 20 byte range while newer facial recognition algorithms feature template sizes as low as 750 bytes which are very feasible for network / smartcard integration. Mapping the application requirements with all the attributes of the specific biometric will determine optimal biometric technology.

No single biometric, or single security device for that matter, can deliver guaranteed 100 percent security. Most customers that purchase information security identification and authentication devices want the best possible security solution that is affordable, easy to implement, yet is unilaterally accepted by the intended user population. The answer to these needs will most likely be a combination of security techniques to include multiple biometrics.

Growth in the biometrics industry has led to an ever increasing number of vendor products available to the prospective buyer. As in most new and emerging technology, there are many small vendors, many large product claims, providing an atmosphere that is difficult for customers to make strong differentiation amongst products.

This appendix raises the implementor's awareness of issues surrounding the best choice of a biometric for their system. This will include issues to utilize the biometric device with a smartcard.

2.0 Selecting Biometrics

To choose the right approach to biometric authentication, you must understand the application, the user base and the characteristics of the biometric device itself. You also must consider the conditions under which it will be used and how fallback authentication methods, such as passwords or tokens, will be instituted when biometrics are not available. Here are some factors to consider before choosing a system.

There are multiple levels of requirements in industry on recognition systems. It can be broken down into two classes, authentication and verification. The obvious choice when combining a token such as a smartcard with biometrics is using verification. A verification (matching the live fingerprint with the template stored on the smartcard) typically requires a more simplistic algorithm and has better performance than authentication (matching one to many). Both verification and authentication require more math computations than the present day smartcard microprocessor can perform in a reasonable period. It is for this reason that verification processing is done on the host rather than by the smartcard. Care must be taken with the system architecture to insure obvious system vulnerabilities don't negate the added protection of adding the biometric / smartcard authentication.

2.1 User Considerations

2.1.1 Public Acceptance

The public may perceive biometric information as a confidential piece of information, much like a social security number. The use of biometric information as a means of access control and/or non-repudiation purposes is likely to enter the privacy issues debate. Even if access to that information requires a lot of effort, it can be a public relations problem.

2.1.2 User Acceptance

The retinal eye scanner had many desirable attributes like a small template, simplistic matching algorithm, fast processing, and very low FAR / FRR (refer to section 2.2.2 "False Acceptance/False Rejection" for details on the FAA/FAR). The device worked by measuring the vein pattern from a circular scan on the back side of a user's cornea. A safe low powered laser diode was used for illumination. However, users' perception of a laser as being dangerous or harmful, virtually shunned any acceptance of this biometric into systems.

Some biometrics, such as fingerprints, may be perceived as associated with the criminal element or an invasion of personal privacy. Vendors are careful to point out that they are not linked with the FBI's fingerprint recognition system or database. In addition, most fingerprint systems devices do not store a raw fingerprint image but a minutiae mapping of image. Given the minutiae mapping or minutiae template, the fingerprint image cannot be reconstructed. General intrusiveness can be another factor affecting user acceptance of some devices.

2.1.3 Target Clientele Characteristics

Some biometric verification products may have better characteristics depending upon the makeup of the users of the system. For example some factors that may effect fingerprint verification may be:

- race and gender. Generally speaking, caucasians have the easiest prints to read, with the prints becoming less defined in other races. Women tend to have more finer print details than men, however, women typically smaller print area which make alignment more critical.
- occupation does have a significant impact. For white collar workers, the problem is not so acute. For manual workers like in IC / PC board production facilities, the system designer might consider using a fingerprint reader using ultra sound which looks beneath the skin surface for fingerprint details.
- age will affect the fingerprint verification systems and will induce higher errors in the system. A combination of more robust variable threshold software matching algorithms and surface pallet coatings should minimize these problems.

One of the factors effecting iris verification are:

- color of eyes. Dark colored irises have presented problems when there isn't enough definition between the pupil and the iris.

It would be advisable to describe the environment and the users to a biometric test facility to see if the FAA or FAR are altered by their characteristics.

2.1.4 User Difficulties

The most common user difficulties deal with alignment in image capture area. If the image is translated or rotated excessively, the verification algorithm has difficulty in matching the user's live image to the stored template. The majority of these problems are corrected the first few times a person uses the scanner. It may be important for the system to provide users with feedback on translation or rotation.

Some populations have difficulty using biometric devices. People with light ridge definition in their fingers may have difficulty using fingerprint-recognition systems. Those who work with abrasive substances--construction workers or even people who handle large volumes of paper--can have their ridges worn down. There also are substantial physical differences based on age, gender and ethnicity[20].

Users with excessively dry, wet or dirty hands have experienced problems with finger- and palm-recognition systems. People wearing gloves generally can't use these systems; however, the ultra-sound-based systems have had limited success detecting prints through thin latex gloves.

Many systems may be tuned to do less strict checking at the expense of weakening the security provided by the system. Administrators have to balance false acceptances versus false rejects, the possibility of fraud versus user convenience[20].

Individual thresholds may be used to lower the threshold for only clients which have poor biometric characteristics. The threshold should still be within acceptable limits, as to not allow a “anyone can pass with my card” scenario.

2.1.5 Ease of Use

One area which has been addressed by biometrics vendors is the ease of use. This factor can be determined by the scanning method, false rejects, and speed of the product. A product that is less intrusive (e.g. facial recognition system) may be less desirable if the false rejection rates are higher than a fingerprint. However, the facial recognition may be better if the user will be opening a door with their hands full.

2.2 Implementation Considerations

2.2.1 Enrolled Image Quality

Enrollment quality is very important to achieve high operational performance. Some enrollment applications have advanced feedback dialog messages which provide useful information about poor quality scans, be it fingerprint, facial or speech. There should be a good balance between the feedback mechanism of the enrollment software and the understanding of acceptable quality by the enrollment officer.

2.2.2 False Acceptance/False Rejection

The False Acceptance Rate (FAR) is the rate at which an intruder can be recognized as a valid user. Many vendors quote the false acceptance rates of their devices, typically generated through mathematical extrapolation of field trial data. As a result, it's difficult to compare these technologies based on vendors' quoted FAR numbers. But it's important to remember that during user verification (a one-to-one match), false acceptance is based on imposter attempts, not on the total number of attempts by valid users. If the FAR is 1 percent, that means one out of 100 users trying to break into the system will be successful[20].

The False Reject Rate (FRR) is the rate at which a valid user is rejected by the system. A 1% FRR would imply the average user would fail every hundredth time. However, it is more likely that only a few individuals may fail a lot more often. These individual may be conduits for a secondary verification mechanism. Many systems, such as the fingerprint-recognition devices, may be tuned to do less strict checking at the expense of opening the system. Administrators have to balance false acceptances versus false rejects, the possibility of fraud versus user convenience.

One method for reducing the false rejects is to use more than one template for verification. The ability to use different fingers for verification can be simply achieved by storing multiple user fingers on the smartcard.

2.2.3 Uniform Testing

Recent US government efforts have been made by the “National Biometric Test Center” or the Commercial Biometrics Developers Consortium (CBDC) to create a consistent testing mechanism for all biometric verification products. Testing vendors products using a uniform or standard database to provide valuable information to consumers of hardware scanner and algorithmic per-

formance. These can be useful in determining how many users will (on the average) be falsely rejected. Keep in mind that if the technology is altered, improved, or changed to be utilized with smartcard applications, that these numbers may not reflect the rates experienced by the resultant system. In those cases the technology should be retested.

2.2.4 Circumvention

The benefits of biometric verification can be nullified if the system is easily fooled by a synthetic device (i.e. rubber finger attack on fingerprint systems). Liveliness testing can be a desirable feature to help prevent such occurrences.

2.2.5 Cost

The cost of implementing the whole system is a factor and importance. The architecture and initial assumptions of the system (i.e using one biometric scanner at the enclave entrance rather than at each workstation) can impact the overall cost of the system greatly. Although cost for biometric verification product have come down sharply over the years it is still not clear whether a business case will develop to have biometric verification at every workstation. It is also not clear, how the cost of these products reflects the infrastructure development expense. Modularity at the interface permits interchange of commercially developed hardware components in the system to maximize the steadily falling commercial biometrics product market.

2.2.6 Continuous Verification

There is an obvious benefit of continuous verification in the computer security application. A facial recognition system which not only logs the user in but randomly checks the login user is still in front of the terminal could be highly desirable. The biometric scanner would need to be at every workstation. The least intrusive product would probably be the most suited to meet this requirement.

2.2.7 Template Storage

Although the biometric template typically cannot be used to create a image or physiological attribute of the user, the template still is sensitive data. The digital representation of what the reader detects--should be encrypted where it's stored, and protected storage locations such as smartcards can improve overall security. The size of the template may be a factor. Most fingerprint and iris templates require between 256 bytes and 1 KB per user, though some systems need up to 8 KB. Face-recognition systems can require up to 3.5 KB per user--too large for some smartcards. Architectures based on a separate database engine that runs in tandem to the operating system may require additional host resources and administration[20].

2.2.8 Data Channel Security

The security of the connectivity between the device and host, as well as the host and any back-end verification engine, is crucial to avoid wire snooping and playback attacks. Many devices that use standard cameras, microphones and other equipment should encrypt or sign packets on the wire. In a Windows NT domain environment, communications between client and server PCs also should be encrypted.

2.2.9 Computer Resources

Biometric matching algorithms should have reasonable performance characteristics using a workstation with medium range processor. Most low-cost devices rely on the processing power of the host and can require up to the power of a Pentium processor and at least 32 MB of RAM.

2.2.10 Comparison Engine Location

If verification is not performed as part of a network that stores and retrieves biometric data, it will be necessary to enroll each new user at every potential location. In a Windows NT environment, for example, client-only verification is much more limited than domain-based verification, which gives users access at any properly equipped workstation.

2.2.11 Calibration

Early fingerprint-recognition systems required careful calibration by a trained expert. And some systems today still require periodic adjustment to assure correct reading. Likewise, voice-recognition systems can require considerable user training, especially when used at more secure sensitivity settings.

If the calibration between the enrolled scanner and the verification scanner effects the FRR or the FAR then this becomes an issue. An auto calibration mechanism may help reduce this problem or eliminate all together.

2.3 Product Considerations For Use With Smartcards

2.3.1 Smartcard Storage

The amount of storage room on the smartcard to store a template (or certificate) may be limited. As a result, the template size and whether multiple templates per user will be supported should be taken into account.

Memory space is always critical for smartcards. Typical template sizes for fingerprint algorithms are 1K or less. The smallest commercial fingerprint templates are 50 bytes; however, the most common size is around 500 bytes[Bio1]. Hand geometry typically have the smallest (15 - 20 byte) template size. A system designer must weigh the template size along with the other factors of choosing a biometric when making a decision on which biometric devices to support.

User alignment problems can more difficult with some systems like facial recognition when storing only one user template. To reduce the FRR the facial biometric product may require multiple templates stored in the smartcard [Bio2]. Trade-off with number of templates stored and amount of memory is available in the smartcard will determine performance.

Refer to Appendix B section 2.0 "Certificate Storage Requirements" for more details on this subject.

2.3.2 Processing Time

Due to some limitations such as data rate, the extraction of the biometric template from the smartcard may take a considerable amount of time. The processing time required to scan a live image, process the data into a template, and verify the result should be calculated to insure that overall

time does not exceed a target goal of 2 seconds. The target time for a biometric product to grab an image, process it and compare templates should be less than 1 second in order to give the smart-card enough time for its processing[Bio3].

2.3.3 Biometric Upgrades/Obsolescence

Products will change/improve over time especially with emerging technologies like smartcards and biometrics. Not all changes are transparent. Incompatibility between versions and flaws in older versions may require significant resources for managing different versions in the field. This is especially true for smartcard implementations. Biometric versions will have to be phased out over time, since updating mobile users may be difficult. Automated means of updating biometrics on a smartcard may be a desirable, however risky, feature (i.e. create a new biometric template and store it on the smartcard during the next successful verification).

DRAFT

3.0 Placing Biometric Templates on the Government Smartcard

The Government Smartcard should hold the authentication certificate which contains the authentication information (AI) for the user which the smartcard represents. This information should be created at the time of enrollment at an enrollment station, sent to an attribute authority for signing, and placed in the smartcard by the enrollment station. The biometric template is placed in the ID authentication certificate as an authentication information attribute. For a complete description of how to place the biometric template in the authentication certificate refer to Appendix D section 2.3.2 “Defining a new Attribute”. For a more detailed description on how the system should create certificates for enrollment, refer to appendix E section 2.0 “A PKI Example”.

DRAFT

Appendix D: Certificates

1.0 Background

1.1 The X.509 Certificate

The X.509 standard is an international standard for security and authentication services supporting security frameworks for electronic information distribution [16]. 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 document, 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 1: 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.

Table 1: X.509 Version 3 Fields

Field	Description
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.
subject PublicKeyInfo	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.
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)	Add on fields (Optional-see following section)
signature	The signature field consist 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.

1.1.1 Attributes

The following is a discussion on attributes that can be used in X.509 certificates. Attributes are defined in X.501 (reference [17]) as “information of a particular type”. The attribute describes a characteristic of an associated object. Refer to X.501, reference [17] 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 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 ({ SupportedAttributes } @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 ({ SupportedContexts } @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/IEO 9594-8 for further details on the X.509 Certificate definitions or X.501 and X.520 for further definitions of the Distinguished Name.

1.2 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 are 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 complimentary to the X.509 certificate. ISO/IWD-15782-2 [2] also describes the attribute certificate, with added details 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 dictates 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 2: 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).
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. The 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.
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.[3]

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.

1.2.1 Attribute Certificate Advantages/Disadvantages

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 compliment 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:

Advantages:

- 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.
- The attribute information can be encrypted, providing access to the confidential information to verified authenticators only.
- 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.
- 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.
- Attribute certificates are becoming standardized (as with X.509).

Disadvantages:

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

2.0 Using Certificates for Authentication

2.1 Encoding Rules

The smartcard will be required to verify the signature on certificates loaded into it for authentication purposes. Since smartcards are limited in storage, processing power, and other resources, it is suggested that Packed Encoding Rules (PER) [Cert1] be used for all certificates from the certificate authorities and user certificates loaded into the smartcard.

2.2 Signature Certificate Processing

The smartcard will be required to store all CA certificates required for authentication. These certificates will be used to verify certificates loaded into the smartcard [Cert2]. This includes any CA certificates, user signature and/or key exchange keys, and authentication certificates. This implies that the smartcard must be capable of decoding each certificate loaded into itself and verify signatures [Cert3].

2.3 Using Attribute 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[6] 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 reference [6] 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.

2.3.1 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 be used.

There are 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 reference [6] as long they are the only authentication techniques the system needs.

2.3.2 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 information construct, as defined in ECMA.219, is given the attribute syntax, the following attribute is the result:

```
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 2.3.2.2 of this document}
```

Since all the attribute placement options discussed in [6] 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 if the application is using five fingerprints per user, that each fingerprint be placed in a separate AuthenticationInfo attribute. This will allow the application using the information to scan the attributes and select the appropriate one.

2.3.2.1 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 register 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.

2.3.2.2 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 3: Biometric information fields

Field Name	Description
ProcessingInfo	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.
MatchingAlgorithmInfo	These fields specify which algorithm(s), and their respective versions, the biometric template is compatible with. They also provide any algorithm specific information.

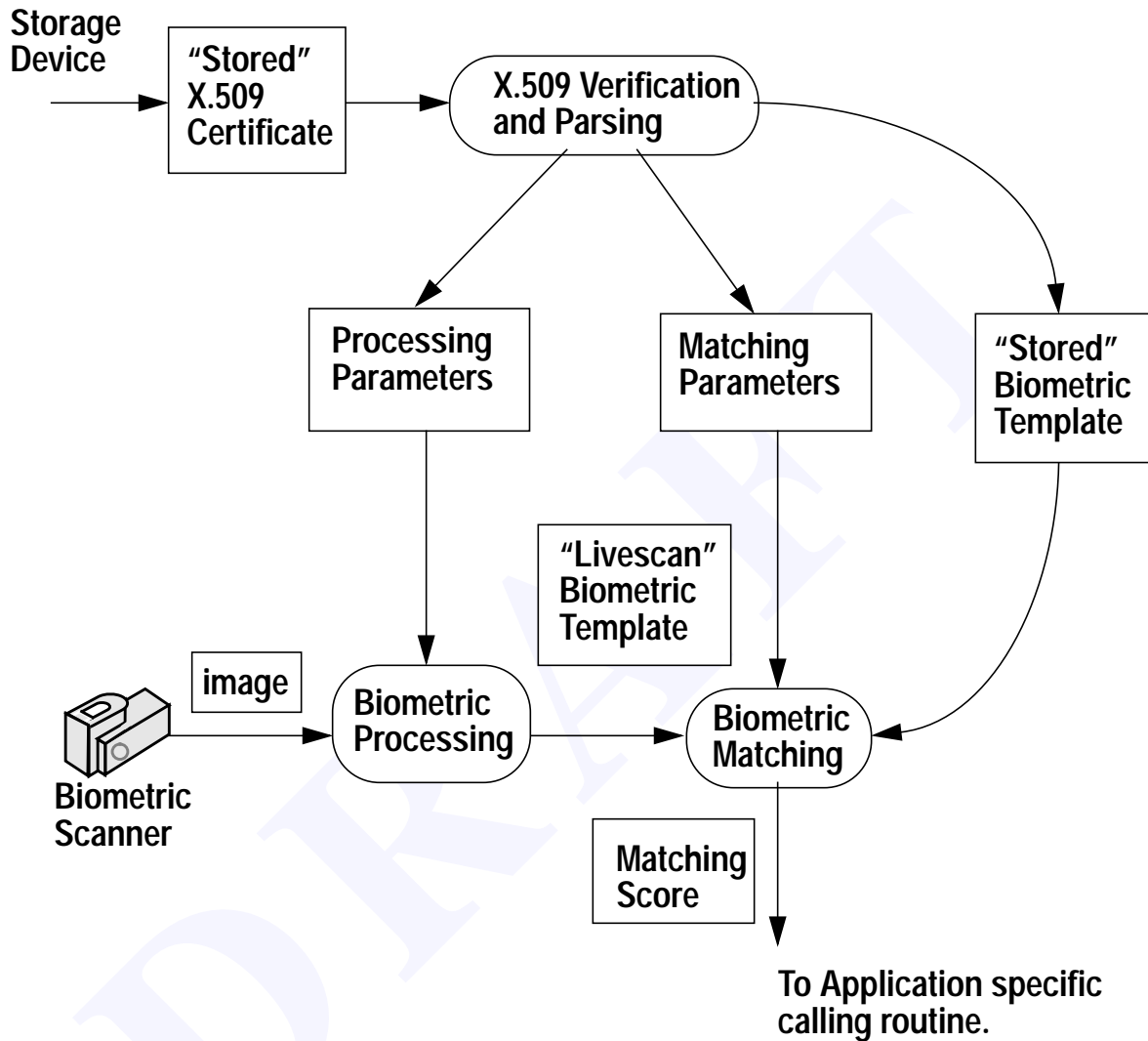
Where each of the information fields will consist of the following data:

Table 4: Biometric information fields

Field Name	Description
ID	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.
Parameters	Each manufacture that registers its process or matching algorithm must document the parameters which are used with it and can be placed in a certificate.
Version	The version of the component used to create the information. Note: the decision of compatibility with the information will be left to the application.

The following diagram illustrates how the biometric processing and matching parameters would be utilized during a biometric verification process:

Figure 1: 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:

```
BiometricInfo ::= SEQUENCE {
    processingInfo          ProcessingAlgorithmInfo OPTIONAL,
    matchingInfo           MatchingAlgorithmInfo }
```

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

2.3.2.3 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 a 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:

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

Version is defined in X.509 as follows:

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

2.3.2.3.1 Registering Biometric Processes

The need to identify this process is negligible 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.

2.3.2.3.2 Biometric Processing Parameters

As stated above, biometric processes only needs 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 may 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 reenter 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 reentering a biometric sample several times.

The processing parameters could follow the established by ECMA.219 `AuthMparam` 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.

2.3.2.4 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       AuthMparam OPTIONAL, -- Defined in ECMA.219
    matchingVersion    Version } -- Defined in X.509
```

Version is defined in X.509 as follows:

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

2.3.2.4.1 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 in the system, there must be a way to choose which software to use. Registering the biometric matching method would be a 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.

2.3.2.4.2 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 may 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 has compatibility 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.

2.3.3 ASN.1 Authentication Information (AI) Attribute Definition

There currently exists no current 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:

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

```
AuthenticationInfo ::= SEQUENCE OF {  
    authenticationMethod [0] AuthenticationMethod, -- defined in ECMA.219  
    exchangeAI [1] AuthMparm, -- the data, as defined in ECMA.219  
    biometricInfo BiometricInfo OPTIONAL -- defined in section 2.3.2.2 of this document }
```

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

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

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

--- ECMA.219 definitions

```
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

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

Each implementation of a biometric matching algorithm and biometric processing functions 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 document.

2.3.4 ASN.1 Authentication Attribute Certificate definition

The attribute certificate that holds the authentication information attribute is described in ASN.1 as follows [Cert 4]:

```
AttributeCertificate ::= SIGNED { AttributeCertificateInfo }

AttributeCertificateInfo ::= SEQUENCE {
    versionVersion DEFAULT v1,
    subjectCHOICE {
        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,

    authenticationInfo AuthenticationInfo,

    UniqueissuerID,

    issuerUniqueIDUniqueIdentifier OPTIONAL,
    extensions Extensions OPTIONAL }

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

AttCertValidityPeriod ::= SEQUENCE {
    notBeforeTimeGeneralizedTime,
    notAfterTimeGeneralized Time }
```

3.0 Approximate Certificate Data Size

An approximation of data sizes can be made on the following assumptions.

- The size of the signature and public key info is set at 512 bits (64 octets where 1octet = 1 byte).
- No extensions are used.
- Packed Encoding Rules (PER) is utilized by the CA signature certificates and user certificate.

Table 5: Contents of Attribute Cert - Identification & Authentication Cert

item	item size	# of items	Total Size
version	5 octets	1	5 octets
owner (baseCertificateID)	8 octets	1	8 octets
issuer (AA)	183 octets	1	183 octets
signature	9 octets	1	9 octets
serialNumber	6 octets	1	6 octets
validity	32 octets	1	32 octets
authenticationInfo -biometricInfo	500 octets	1	500 octets
issuerUniqueID (Token Serial #)	16 octets	1	16 octets
algorithmIdentifier	9 octets	1	9 octets
signatureValue	70 octets	1	70 octets
Total			838 octets

If additional fields are added (such as extensions) simply add the length of the new field to the total.

Table 6: Contents of Cert_{Host}

item	item size	# of items	Total Size
version	5 octets	1	5 octets
serialNumber	6 octets	1	6 octets
signature	9 octets	1	9 octets
issuer	183 octets	1	183 octets

Table 6: Contents of Cert_{Host}

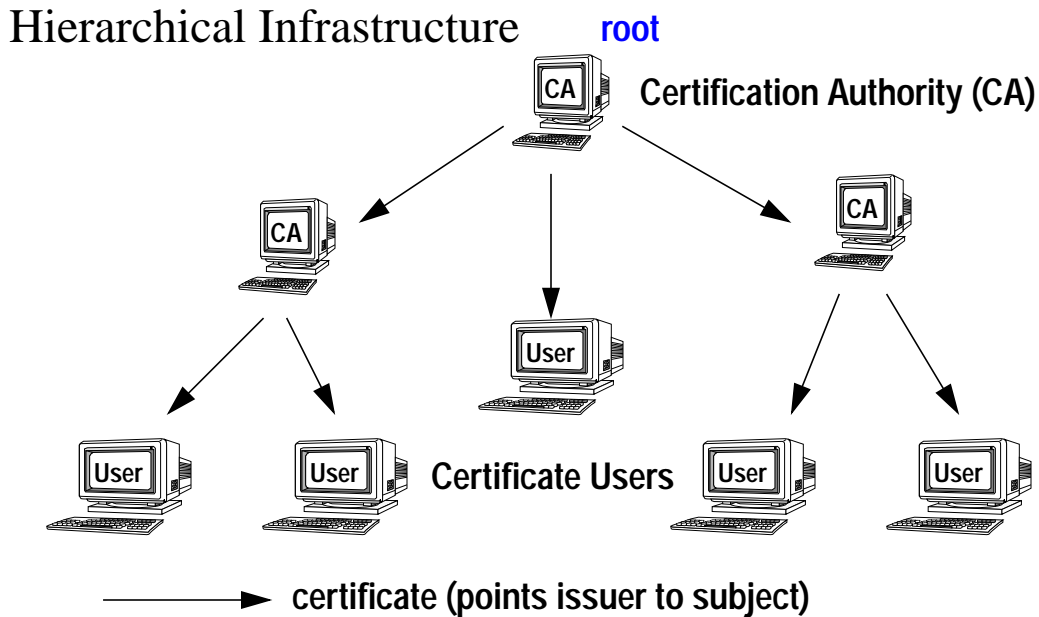
item	item size	# of items	Total Size
validity	32 octets	1	32 octets
subject	183 octets	1	183 octets
subjectPublicKeyInfo	82 octets	1	82 octets
algorithmIdentifier	9 octets	1	9 octets
signatureValue	70 octets	1	70 octets
Total			579 octets

Appendix E: Public Key Infrastructure

1.0 Background

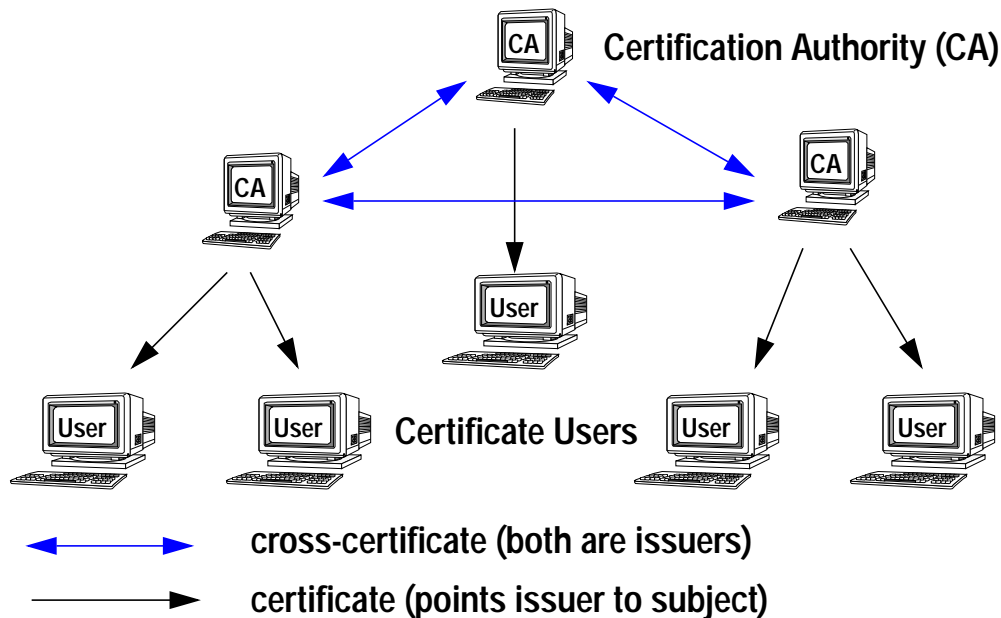
The use of public key cryptography for all but the smallest community of users will require the existence of an infrastructure to support the generation and distribution of keys [PKI1]. The preferred method is currently through public key certificates. Public key certificates are digital documents that, at a minimum, contain the name and public key of a user, and are digitally signed by a certification authority (CA). These public key certificates are utilized to reliably associate a user's name to his public key, thereby providing authentication in the form of digital signatures.

Trust is transferred via certification paths in a public key infrastructure (PKI). A certification path is nothing more than a chain of certificates in which the signature on the previous certificate is verified by the public key of the next certificate until a certificate is reached which is trusted by the verifier. This is usually the CA of the verifier. This path will follow one of two topologies: hierarchical or network. In a hierarchical topology, all CA's are arranged under a 'root' CA that issues certificates to subordinate CA's, who may issue certificates to users or to subordinate CA's, and so on. This topology works best for government or military applications.



In a network topology, CA's cross certify each other. Each CA issues a certificate to the other. This means that each can place trust in the certificates issued by the other without both being 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.

1.1 PKI System Components

1.1.1 Enrollment Station

The enrollment station is a “trusted” entity which is responsible for registering the user into the system. The user must be approved via another mechanism (i.e. clearances) prior to being enrolled on the system [PKI2]. The enrollment station gathers the user’s identification information, the biometric information, and other system required information (i.e. privileges, clearance levels, etc.), and then sends it to the appropriate authority for further processing. The enrollment station retrieves the processed information from the authorities and places the certificates into the token.

1.1.2 Certificate Authority

The Certificate Authority (CA) is a “trusted” entity which is a which creates, distributes, and maintains public key certificates for the system. The enrollment station passes the raw (unformatted) public key to the CA in the form of a signed certificate request. If the CA accepts the certificate request, it formats and signs the certificate in accordance with the X.509 specification.

1.1.3 Attribute Authority

The Attribute Authority(AA) is a “trusted” entity which is responsible for creating attribute certificates as defined by the system [PKI3]. The enrollment station send the attribute information (biometric, clearance, and privilege information) to the AA. Once the information is validated, the AA formats and signs the attribute certificate in accordance with X.509. The attribute certificates are returned to the enrollment station for distribution to the appropriate token.

1.1.4 ID Station

The ID station is a “trusted” entity responsible for performing biometric verification of the user. To perform this function, it must extract the signed biometric template from the token, verify the signature, read the biometric scan from the user, and compare the scanned image with that of the verified biometric template [PKI4]. The ID station uses the “what you have” and “who you are” portions of identification and authentication. The “what you know” is obtained by the workstation. Since the functions performed on the ID station are of a critical nature to the system, connectivity with the rest of the system is not allowed. ID stations may be nested or placed in parallel with each other but are not to be interconnected. Thus, the system cannot use a trusted third party architecture.

1.1.5 Workstation

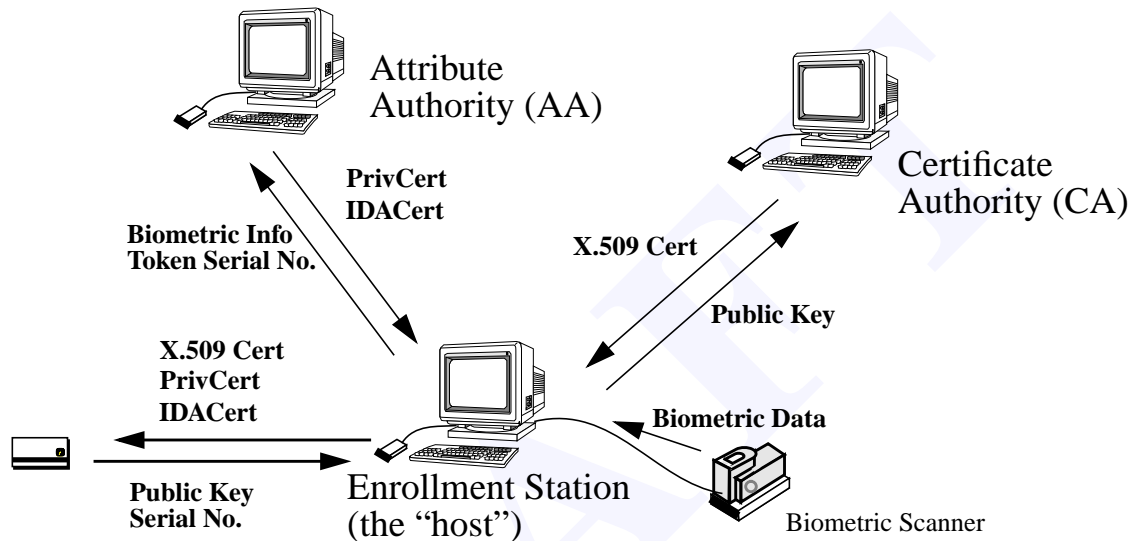
The workstation is an “untrusted” entity which can be any service access point which provides authentication beyond that of the ID station by requiring that the user provide a password to the token. The workstation does contain a “trusted” sub component called a CryptoCard. The CryptoCard is utilized for encrypting all data on the workstation to protect against malicious access. The workstation is responsible for performing checks on authentication data issued to the token by an ID station. Services offered by the workstation may include: system logon, document signing, document source verification, or encryption/decryption.

2.0 A PKI Example

2.1 System Enrollment

System enrollment is the process by which one is certified for using the system. The enrollment station is responsible for collecting the enrollment information and delegating the tasks that need to be performed. The system enrollment process is illustrated in the following figure.

The System Enrollment Process



The user must be approved via another mechanism (i.e. a background check) prior to being enrolled on the system. This is typically a manual operation and is not covered by this description.

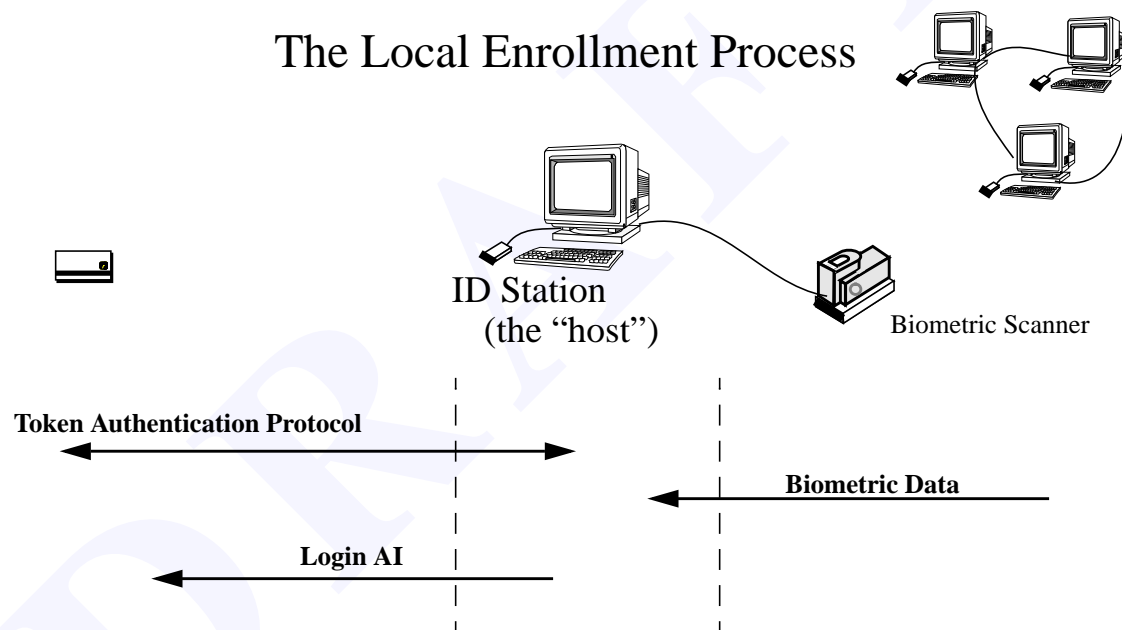
The steps to achieve system enrollment are:

1. The token generates a public/private key pair.
2. The enrollment station retrieves the public key from the token and sends it to the CA for X.509 formatting. The formatted key is returned to the enrollment station and placed in the token for storage.
3. User information including first name, middle initial, last name, organization, organizational unit, role (i.e. Security Officer, admin, auditor, user, etc.), and clearances are entered into the enrollment station.
4. The host reads in the serial number of the token.
5. A biometric livescan is taken of the user to be enrolled. The livescan is processed into a biometric template.
6. The biometric template, the token's serial number, the public key certificate's serial number, and the user's authentication information are sent to the AA. The AA creates the X.509 formatted attribute certificates (the privilege certificate(PrivCert) and the ID authentication certificate (IDACert)). The attribute certificates are returned to the enrollment station and placed in the token.

7. The user enters in a PIN or password to the token.
8. A symmetric key used as a Storage Key (SKey) by the CryptoCard resident in the workstations is generated by the host and symmetric key encrypted using a DomainSKeyEncryption-Key (DSKEK) and distributed to the workstations. The DSKEK is used for subsequent workstation logons.
9. The system may optionally provide an additional authentication attribute (such as a password) to the token which can be used during the logon process.

2.2 Local Enrollment

Local enrollment occurs when the a system enrolled user needs access to a local domain. This usually involve the creating of a system account using tools provided by the local operating system (OS). The operating system must be upgraded to provide the local enrollment access control techniques described in the following sections. The local enrollment process is illustrated in the following diagram:



The steps to achieve local enrollment are as follows:

1. The token and the local host mutually authenticate each other using the Token Authentication Protocol (reference [5]). As part of the protocol, the token sends the PrivCert and IDACert to the host over the established encrypted channel.
2. The host validates the **signatureValue** of the PrivCert and IDACert.
3. The **clearance** information of the PrivCert is compared to the classification of the local system. The **clearance** on the certificate must be equal to or higher than those of the local system.
4. The serial number of the token is compared to the **issuerUniqueID** attribute found in the PrivCert and IDACert.

5. The **serialNumber** of the public key certificate is compared to the **owner (baseCertificateID)** attribute found in the PrivCert and IDACert.
6. A livescan biometric sample is taken of the user and processed into a biometric template. The template is compared to a compatible biometric template found in the IDACert.
7. If each step in this sequence occurs correctly, then the user is given an account to the system, using the OS local registration process (such as the Windows NT user manager).
8. A symmetric key used as a Storage Key (SKey) by the CryptoCard resident in the workstations is generated by the host and symmetric key encrypted using a DomainSKeyEncryption-Key (DSKEK) and distributed to the workstations. The DSKEK is used for subsequent Workstation logons.
9. The system may optionally provide an additional authentication attribute (such as a password) to the token which can be used during the logon process.

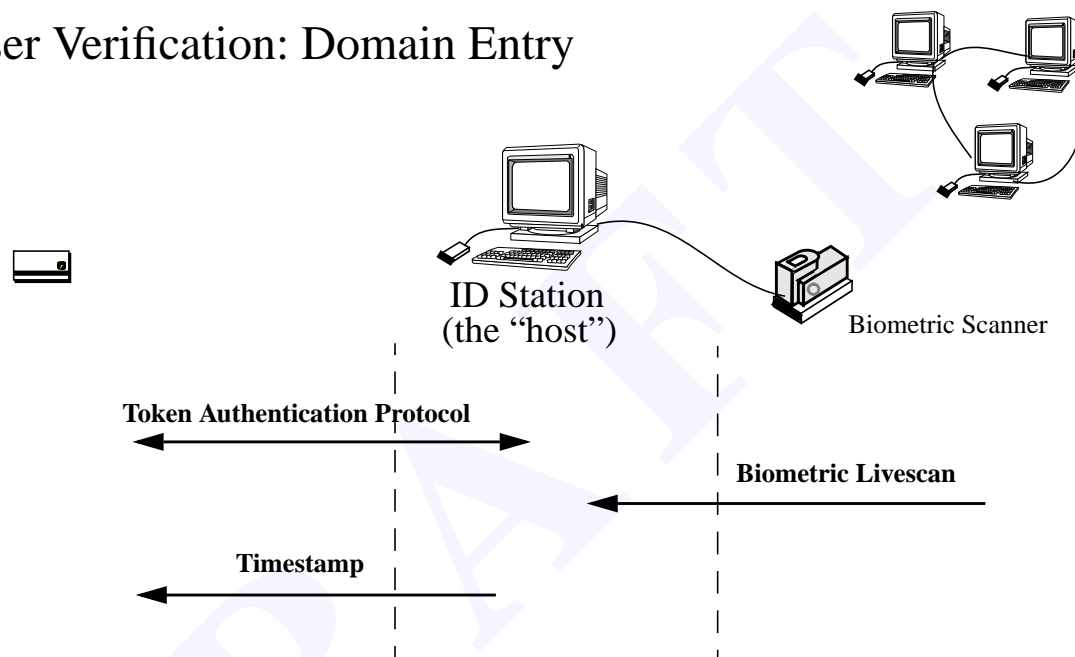
2.2.1 Workstation Access

Since a “trusted” workstation is a distant reality, the system distributes the trust to several components in order to limit the vulnerabilities of the untrusted workstation. The IDstation is a trusted entity which performs the biometric and token verification. If the verification is successful, a timestamped certificate (referred to as the timestamp) is issued to the token allowing access to the system for a specific amount of time. When the user logs into a workstation this information is used to verify that the biometric verification has taken place.

2.2.1.1 Domain Verification

The ID station is responsible for performing user verification. The ID station is intended to be placed at an enclave entrance (such as a door) where all will have easy access. It can (optionally) be used to control a locking mechanism providing some physical security. The ID station verifies the validity of the token by performing the Authentication Protocol (reference [5]). The authenticity of the biometric data contained in the ID Authentication Certificate is accomplished by further attribute certificate validation. The user is verified by comparison of certified template to a livescan. The following diagram illustrates the domain verification process:

User Verification: Domain Entry



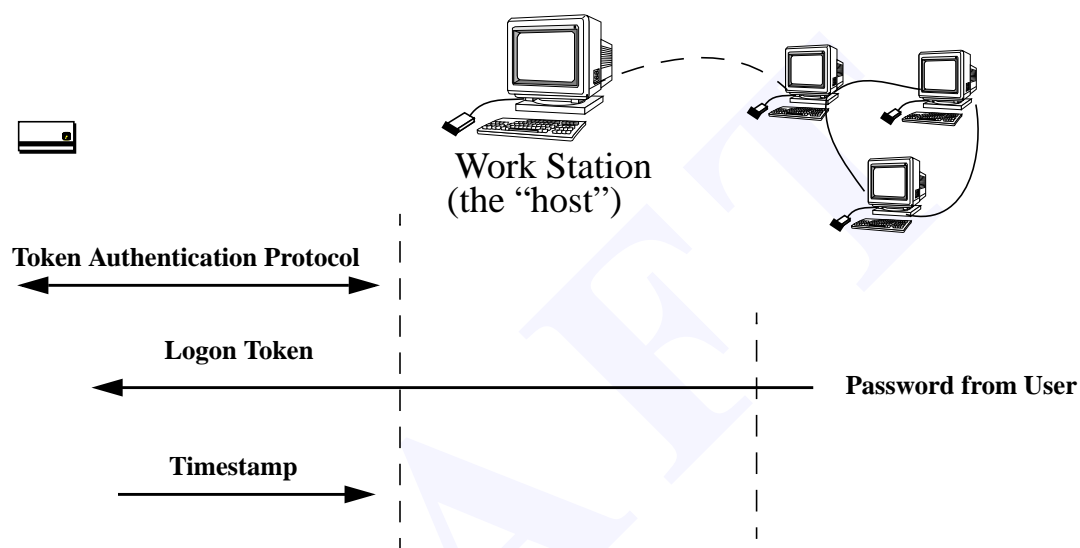
The steps to achieve domain verification are as follows:

1. The token and the local host mutually authenticates each other using the Token Authentication Protocol (reference [5]). As part of the protocol, the token sends the PrivCert and IDACert to the host over the established encrypted channel.
2. The host validates the **signatureValue** of the PrivCert and IDACert.
3. The **clearance** information of the PrivCert is compared to the classification of the local system. The **clearance** on the certificate must be equal to or higher than those of the local system.
4. The serial number of the token is compared to the **issuerUniqueID** attribute found in the PrivCert and IDACert.
5. The **serialNumber** of the public key certificate is compared to the **owner (baseCertificateID)** attribute found in the PrivCert and IDACert.
6. A livescan biometric sample is taken of the user and processed into a biometric template. The template is compared to a compatible biometric template found in the IDACert.
7. If each step in this sequence occurs correctly, then a timestamped certificate is generated and signed by the ID station and transferred onto the token.

2.2.1.2 Workstation Verification

The workstation OS's logon function must be trusted to perform the functions specified in this section to log the user into the system. The following diagram illustrates the workstation verification process:

User Verification: Workstation Login



The steps to achieve workstation verification are as follows:

1. The token and the local host mutually authenticate each other using the Token Authentication Protocol (reference [5]).
2. The token sends the PrivCert to the host over an encrypted channel.
3. The host validates the **signatureValue** of the PrivCert.
4. The **clearance** information of the PrivCert is compared to the classification of the local system. The **clearance** on the certificate must be equal to or higher than those of the local system.
5. The serial number of the token is compared to the **issuerUniqueID** attribute found in the PrivCert.
6. The **serialNumber** of the public key certificate is compared to the **owner (baseCertificateID)** attribute found in the PrivCert.
7. The token provides the timestamped certificate to the host for validation.
8. The user logs into (provides a password or PIN to) the token.
9. If each step in this sequence occurs correctly, then the user is logged onto the network.