U.S. Government
Biometric Verification Mode
Protection Profile
for
Medium Robustness Environments

Information
Assurance
Directorate

Version 1.1

July 25, 2007

Preface

Protection Profile Title:

U.S. Government Protection Profile Biometric Verification Mode for Medium Robustness Environments

Criteria Version:

This Protection Profile *"US Government Protection Profile Biometric Verification Mode for Medium Robustness Environments"* (PP) was updated using Version 3.1 of the Common Criteria (CC).

Editor's note: The purpose of this update was to bring the PP up to the new CC 3.1 standard without changing the authors' original meaning or purpose of the documented requirements. The original PP was developed using version 2.x of the CC. The CC version 2.3 was the final version 2 update that included all international interpretations. CC version 3.1 used the final CC version 2.3 Security Functional Requirements (SFR)s as the new set of SFRs for version 3.1. Some minor changes were made to the SFRs in version 3.1, including moving a few SFRs to Security Assurance Requirements (SAR)s. There may be other minor differences between some SFRs in the version 2.3 PP and the new version 3.1 SFRs. These minor differences were not modified to ensure the author's original intent was preserved.

The version 3.1 SARs were rewritten by the common criteria international community. The NIAP/CCEVS staff developed an assurance equivalence mapping between the version 2.3 and 3.1 SARs. The assurance equivalent version 3.1 SARs replaced the version 2.3 SARs in the PP.

Any issue that may arise when claiming compliance with this PP can be resolved using the observation report (OR) and observation decision (OD) process.

Further information, including the status and updates of this protection profile can be found on the CCEVS website: http://www.niap-ccevs.org/cc-scheme/pp/. Comments on this document should be directed to ppcomments@missi.ncsc.mil. The email should include the title of the document, the page, the section number, the paragraph number, and the detailed comment and recommendation.

Record of Release

(Back to TOC)

Release #	**Date**	**Area Affected**	**Comment**
Release 1.0	February 23, 2006	Complete Document	Release of the NIAP evaluated PP
Release 1.1	July 25, 2007	Assurance requirements Functional requirements NIAP Interps Rationale	Updated to CC version 3.1

Protection Profile Title:

U.S. Government Biometric Verification Mode Protection Profile for Medium Robustness Environments.

Criteria Version:

This Protection Profile (PP) was developed using Version 3.1 of the Common Criteria (CC) [1] and applying the NIAP interpretations that have been approved by TTAP/CCEVS Management as of July 10, 2002.

Table of Contents

1.0 INTRODUCTION

This Biometric Verification Mode Protection Profile (PP) for Medium Robustness Environments was sponsored by the Biometrics Management Office (BMO) and the National Security Agency (NSA). A verification mode biometrics device is one that *authenticates* a user for a claimed identity. This is distinctly different from an identification mode biometrics device, which attempts to identify an individual by their biometric characteristic. This Protection Profile is intended to be used as follows:

- For product vendors and security product evaluators, this PP defines the requirements that must be addressed by specific products as documented in vendor Security Targets (STs).
- For system integrators, this PP is useful in identifying areas that need to be addressed to provide secure system solutions. By matching the PP with available STs, security gaps may be identified and products or procedures may be configured to bridge these gaps.

1.1 Protection Profile Identification

Title: U.S. Government Biometric Verification Mode Protection Profile (PP) for Medium Robustness Environments

Sponsor: The Biometrics Management Office and the National Security Agency (NSA)

CC Version: Common Criteria (CC) Version 3.1, and applicable interpretations.

Registration: <to be provided upon registration>

Protection Profile Version: Version 1.1, dated July 25, 2007

Keywords: Protection Profile, Medium Robustness Environments, verification mode, liveness, biometrics

1.2 Protection Profile Overview

This Protection Profile (PP) specifies the minimum functional and assurance security requirements for biometric products operating in verification mode to provide authentication allowing physical and logical access control to facilities as well as to information systems in medium robustness environments (see Section 3.0 for a characterization of medium robustness environments). Biometric systems are enabling technologies designed to augment existing security measures by positively authenticating individuals based on measurable physical features or behaviors. Due to the unique nature of a biometrics TOE and the desire of the PP authors to attempt to accommodate the wide range of biometric technologies, extended requirements were necessary, as was a great deal of refinement of the CC requirements.

The requirements section of this PP specifies a need to protect biometric templates, to provide confidentially, and integrity. Since the biometric package (which includes the user identifier and their associated reference template(s)) may be stored in a device outside the control of the TOE, the biometrics TOE encrypts biometric packages for confidentiality reasons, and an enrolling TOE cryptographically signs a biometrics package so that modification of the package can be detected.

A TOE conformant to this PP satisfies the specified functional requirements, as well as the Medium Robustness assurance requirements that are expressed in Section 5.2 TOE Security Assurance Requirements.

STs that claim conformance to this PP shall meet a minimum standard of demonstrable-PP conformance as defined in section D3 of part 1.

This PP defines:

- assumptions about the security aspects of the environment in which the TOE will be used;
- threats that are to be addressed by the TOE;
- security objectives of the TOE and its environment;
- functional and assurance requirements to meet those security objectives; and
- rationale demonstrating how the requirements meet the security objectives, and how the security objectives address the threats.

1.3 Related Protection Profiles

A basic robustness PP for a biometric TOE operating in verification mode has many of the same functional requirements, but does not require the use of cryptography to protect the biometric packages. Contrary to a medium robustness TOE, the basic robustness TOE has a reliance on the IT environment in order to address some of the threats and to enforce its security policies. The basic robustness PP has less stringent assurance requirements as well.

Rather than write a PP that specifies requirements for both verification mode and identification mode, a decision was made to write a PP for each mode of operation. This affords product developers the opportunity to evaluate their product and claim conformance to a PP if their product operates in only one of the modes of operation. This approach allows a product that operates in both modes the opportunity to claim conformance to each of the PPs. The following PPs make up the family of PPs sponsored by the BMO and NSA:[1]

[1] This is the first Protection Profile to be released in the family of Biometrics PPs and the remaining PPs are currently in draft form and not yet available for public release.

- U.S. Government Biometric Verification Mode Protection Profile For Basic Robustness Environments, dated (TBD)

- U.S. Government Biometric Identification Mode Protection Profile For Medium Robustness Environments, dated (TBD)

- U.S. Government Biometric Identification Mode Protection Profile For Basic Robustness Environments, dated (TBD)

1.4 Conventions

The notation, formatting, and conventions used in this PP are largely consistent with those used in version 3.1 of the Common Criteria (CC). Selected presentation choices are discussed here to aid the PP user.

The CC allows several operations to be performed on functional requirements; *refinement*, *selection*, *assignment*, and *iteration* are defined in paragraph C.4 of Part 1 of the CC. Each of these operations is used in this PP.

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by the word refinement in **bold text** and the added/changed words are in **bold text**. In cases where words from a CC requirement were deleted, a separate attachment indicates the words that were removed.

The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections that have been made by the PP authors are denoted by *italicized text*, selections to be filled in by the ST author appear in square brackets with an indication that a selection is to be made, [selection:], and are not italicized.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments that have been made by the PP authors are denoted by showing the value in square brackets, [Assignment_value], assignments to be filled in by the ST author appear in square brackets with an indication that an assignment is to be made [assignment:].

The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing the iteration number in parenthesis following the component identifier, (iteration_number).

As this PP was sponsored, in part by NSA, National Information Assurance Partnership (NIAP) interpretations are used and are presented with the NIAP interpretation number as part of the requirement identifier (e.g., **FAU_GEN.1-NIAP-0410** for Audit data generation).

The CC paradigm also allows protection profile and security target authors to create their own requirements. Such requirements are termed ‘extended requirements’ and are permitted if the CC does not offer suitable requirements to meet the authors’ needs. Extended requirements must be identified and are required to use the CC class/family/component model in articulating the requirements. In this PP, extended requirements will be indicated with the "EXP" following the component name.

Application Notes are provided to help the developer, either to clarify the intent of a requirement, identify implementation choices, or to define "pass-fail" criteria for a requirement. For those components where Application Notes are appropriate, the Application Notes will follow the requirement component.

1.5 Protection Profile Organization

Section 1, Protection Profile Introduction, provides document management and overview information necessary to identify the PP along with references to other related PP's.

Section 2, Target of Evaluation (TOE) Description, defines the TOE and establishes the context of the TOE by referencing generalized security requirements.

Section 3, TOE Security Environment (TSE), describes the expected environment in which the TOE is to be used. This section defines the set of threats that are relevant to the secure operation of the TOE, organizational security policies with which the TOE must comply, and secure usage assumptions applicable to this analysis.

Section 4, Security Objectives, defines the set of security objectives to be satisfied by the TOE and by the TOE operating environment.

Section 5, IT Security Requirements, defines the security functional and assurance requirements derived from the Common Criteria, Part 2 and Part 3, respectively, that must be satisfied by the TOE and the Non-IT environment.

Section 6, Rationale, provides rationale to demonstrate that the security objectives satisfy the threats and policies. This section also explains how the set of requirements are complete relative to the security objectives and presents a set of arguments that address dependency analysis and use of the extended requirement.

Section 7, References, provides background material for further investigation by users of the PP.

Section 8, Terminology, provides a listing of definitions of terms.

Section 9, Acronyms, provides a listing of acronyms used throughout the document.

Section 10, Refinements, identifies the refinements that were made to CC requirements where text is deleted from a requirement.

2.0 TOE DESCRIPTION

This section describes biometric authentication devices as the Target of Evaluation (TOE) for this protection profile.

Biometric TOEs are unlike other information-technology-related TOEs. Untrusted users who interact with the TOE (known as “subjects” in the biometrics community, but not in the Common Criteria community) are not really *users* of the TOE. Their only role is to present a claimed identity and a fresh biometric sample, and the biometric TOE decides whether the biometric sample comes from a live individual and whether the biometric sample matches the biometric previously enrolled by the user with the claimed identity. The TOE does not contain any user data and does not provide a logical interface to untrusted users. The TOE only contains TSF data and the logical interface presented is only for administrative functions.

The physical and logical boundaries of the TOE will differ depending upon a vendor’s implementation and the intended use of the product. There are many permutations of where these components can be hosted.

For controlling physical access (e.g., a building or room), a TOE could be comprised of components that are physically and logically housed in a single unit. An example is a device whose ultimate purpose is to control access to a door, which performs the capture and comparison functions within a single unit and is stand alone. A TOE could also have multiple capture devices that transmit the live template to a server that then performs the comparison function, which then generates the match/no match decision.

For controlling local logical access to an IT product (e.g., a workstation) the TOE’s physical boundary could take different forms as well. As with the example above, the TOE could be contained in a single unit and provide a match/no match decision to the IT product, or the TOE could be physically separated. If the TOE is physically separated it could use the IT product to transmit data (e.g., the live template, capture device’s identity) through the IT product to another component of the TOE that performs the comparison function, which then in turn provides the match/no match decision to the IT product. It is important to note that the TOE includes all the hardware and software that play a role in the TOE being able to satisfy the security requirements specified in this PP. When the TOE is physically separated, cryptography is used to maintain confidentiality and to detect modification of the transmitted data. It is also important to note that none of the TOE’s software is executing on a platform other than the trusted platform provided by the TOE. This means that the comparison software or any capture controller function is not running on an IT product other than the TOE. Figure 1 illustrates an example of a distributed TOE. In this example, the capture device is connected to an IT product (e.g., workstation) via a direct connection (e.g., USB connection) and the IT product is connected to a network. The capture device transmits the live template, and possibly other data (e.g., unique device id), to the comparator through a path that is not trusted with respect to the TOE. This is acceptable, since the capture device signs and encrypts the data being transmitted. The comparator retrieves the reference template from storage. The reference template is included in the biometric package, which is encrypted and cryptographically signed by the TOE (or another authorized entity). The comparator compares the templates and generates a match/no match decision, which is then sent to the IT product in the clear. Sending the decision in the clear is permitted, since once the decision leaves the TOE's scope of control, it is left to the IT environment, including the IT product, to handle the decision appropriately.

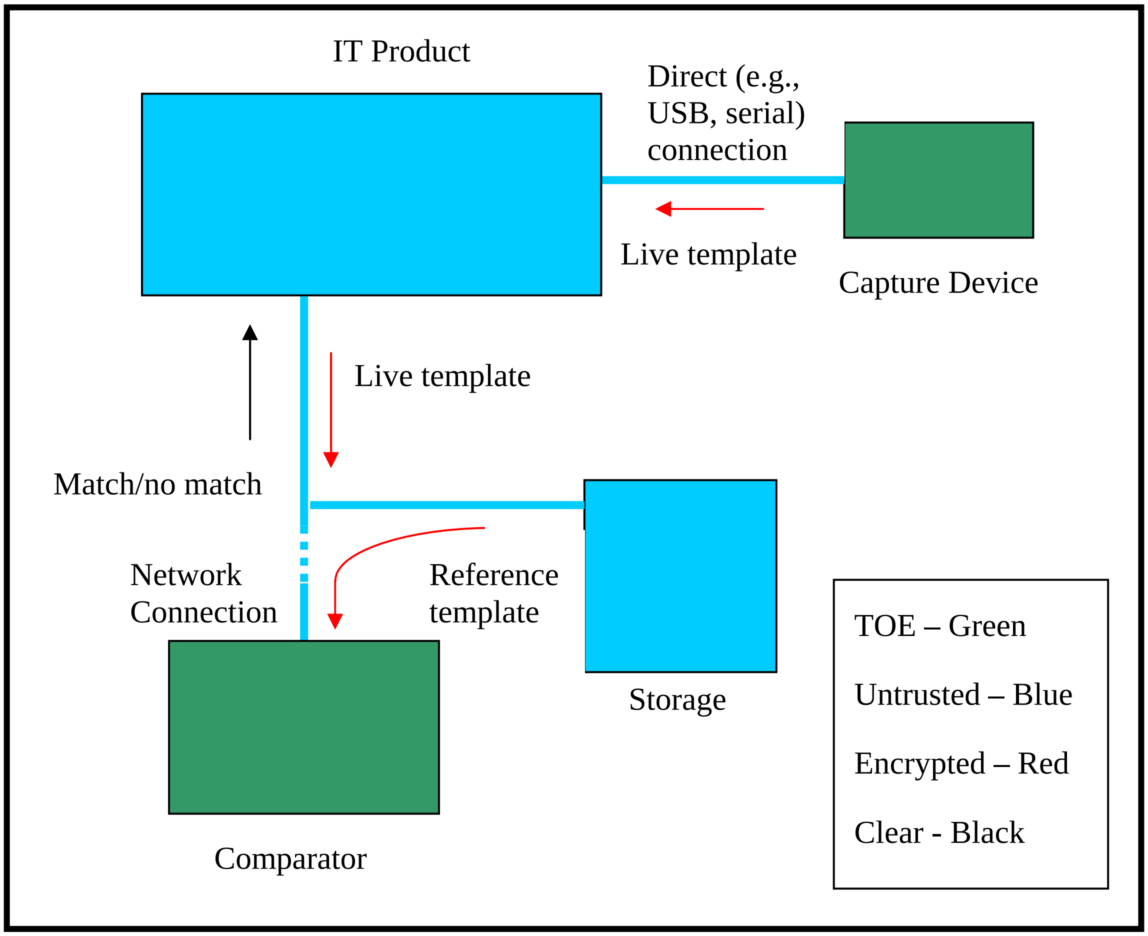


Figure 1. Example of a distributed TOE.

Another important aspect of the TOE, as defined by this PP, is that the storage of the biometric reference template is outside the scope of the TOE. This was done to allow flexibility in the deployment of the TOE and the scenarios under which the TOE or instantiations of the TOE may be used. The requirements in this PP were written to allow the storage of reference templates to take place at a single repository, be distributed across database servers, to allow single reference templates to be stored on a smart card, or other ways in which a developer wishes to handle storage. This is secure, since the biometrics package that contains the reference template is signed and encrypted by the TOE that performs the enrollment. However, the enrolling TOE must be a trusted signing authority for any instantiation of TOEs that are to use the biometric package when performing the authentication process.

This TOE requires that a second, non-biometric authentication mechanism (e.g., password, PIN) be available to end-users for administrative purposes. This was done to provide end-users with the flexibility of requiring more rigorous authentication for an administrator if they choose, or to allow administrators to solely use the non-biometric authentication mechanism. The latter may be useful if the capture device became unusable.

2.1 Biometric TOE Functionality

“Biometric Authentication” refers to the automatic identification or identity verification of living individuals based on physiological or behavioral characteristics. Examples of physiological characteristics include hand or finger images, facial characteristics, speaker verification and eye patterns. Biometric authentication is the “automatic”, “real-time”, “non-forensic” subset of the broader field of human identification.

In this protection profile, biometric devices are seen as components of security systems that provide positive authentication. As with other types of authentication technologies, biometrics provides mechanisms to quickly and securely associate an identity with a person. The distinctive feature about biometric technologies as an authentication factor is that the presenter of a valid biometric that matches an enrolled biometric is, by definition, an authorized user, in contrast with technologies such as tokens or passwords, where valid instances of these items can be presented by unauthorized users.

Figure 2 shows a simple model of a biometric TOE showing major components required for this protection profile. This figure, as well as Figure 3 and Figure 4, shows a one-way information flow that does not take into account the information flow from the TOE to the user (e.g., prompt for entering a claimed user identifier, or other information, such as directing the user which finger to present to the capture device). The following is a description of each block in the diagram:

- *Liveness Check & Capture* – A liveness check that determines if the host of the biometric sample has certain characteristics belonging to living human beings. In capture, a sample of the user’s biometric is acquired using the required sensor (camera, microphone, fingerprint scanner, etc.). It is important to note the liveness check is performed at the same time as the capturing of the biometric characteristic.

- *Extraction* – Process by which the biometric sample captured in the previous block is transformed into an electronic representation. During enrollment this electronic representation is known as the biometric template. During the authentication process, it is known as the live sample.

- *Package Creation* – Performed only during enrollment. The TOE creates a “Biometrics Package” during enrollment. The biometric package is where a user identifier is associated with one or more reference templates. Cryptographically bind the user's identity and additional information with the biometric template to create a biometric package for storage.

- *Package Assurance* – Performed only during enrollment. Uses cryptographic methods to protect the confidentiality and integrity of the biometric package for storage.

- *Package Validation* – Performed only during authentication. Verifies the integrity of the biometric package received from storage and the validity of the signing authority.

- *Comparison* – Performed only during authentication. Matches the live sample and biometric templates. The result from the matching is a score, which is then compared against predefined threshold values.

- *Security Management Functions* – The TOE provides management functions to the TOE administrator that includes setting of the threshold, determining audit events, reviewing audit information, and key management.

This protection profile requires that when the matching score is outside the maximum and minimum threshold range, a *no-match* result is generated.

Cryptographic methods and modules must comply with approved standards and be validated by NIST's FIPS 140-2 validation program.

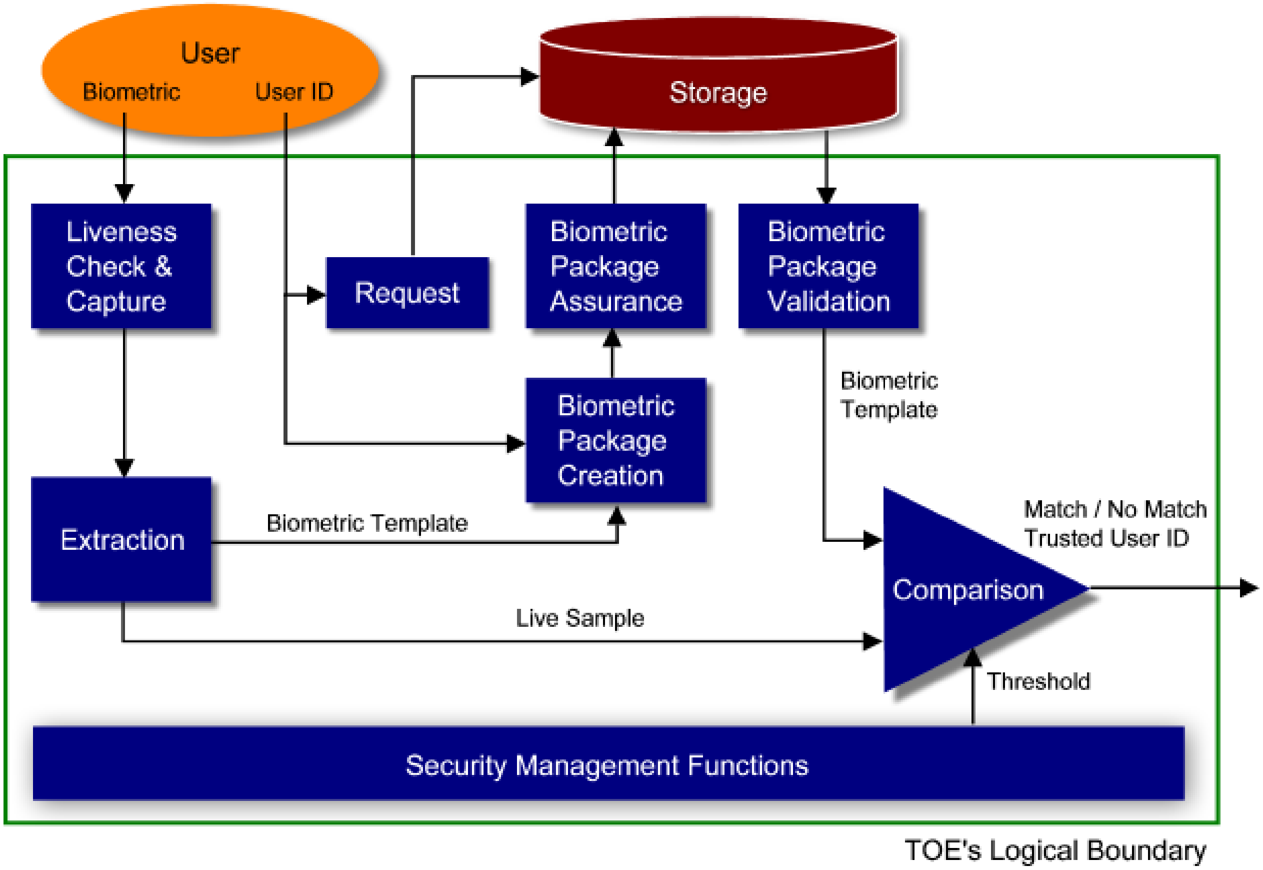


Figure 2. TOE functional block diagram

The basic processes a biometric TOE supports are enrollment and authentication. During enrollment, the biometric TOE captures the biometric sample from an enrollee, transforms it into a biometric template, and associates this template with the enrollee's identity for storage.

During authentication, the biometric TOE can be used for identification or verification of the person's identity. In identification, the biometric TOE attempts to determine the identity of a person by comparing the captured biometric sample against a database of enrolled templates for a match. In verification, the biometric device verifies a person's claimed identity by matching a captured biometric sample against the enrolled template associated with the claimed identity. This document considers a biometric TOE operating only in the verification mode.

The next sections describe the enrollment and verification modes in more detail.

2.1.1 The Enrollment Process

Figure 3 highlights the components of a biometric TOE involved during enrollment. Certainly, the process to enroll a user in the biometric TOE will form a part of a larger registration step.

The site should follow appropriate procedures for validating the identity of the individuals before enrolling them into their system. Only enrollment administrators can enroll users in a biometric TOE. The TOE's administrative guidance provides enrollment administrators guidance about acceptable quality metrics in regards to the quality of the biometric template.

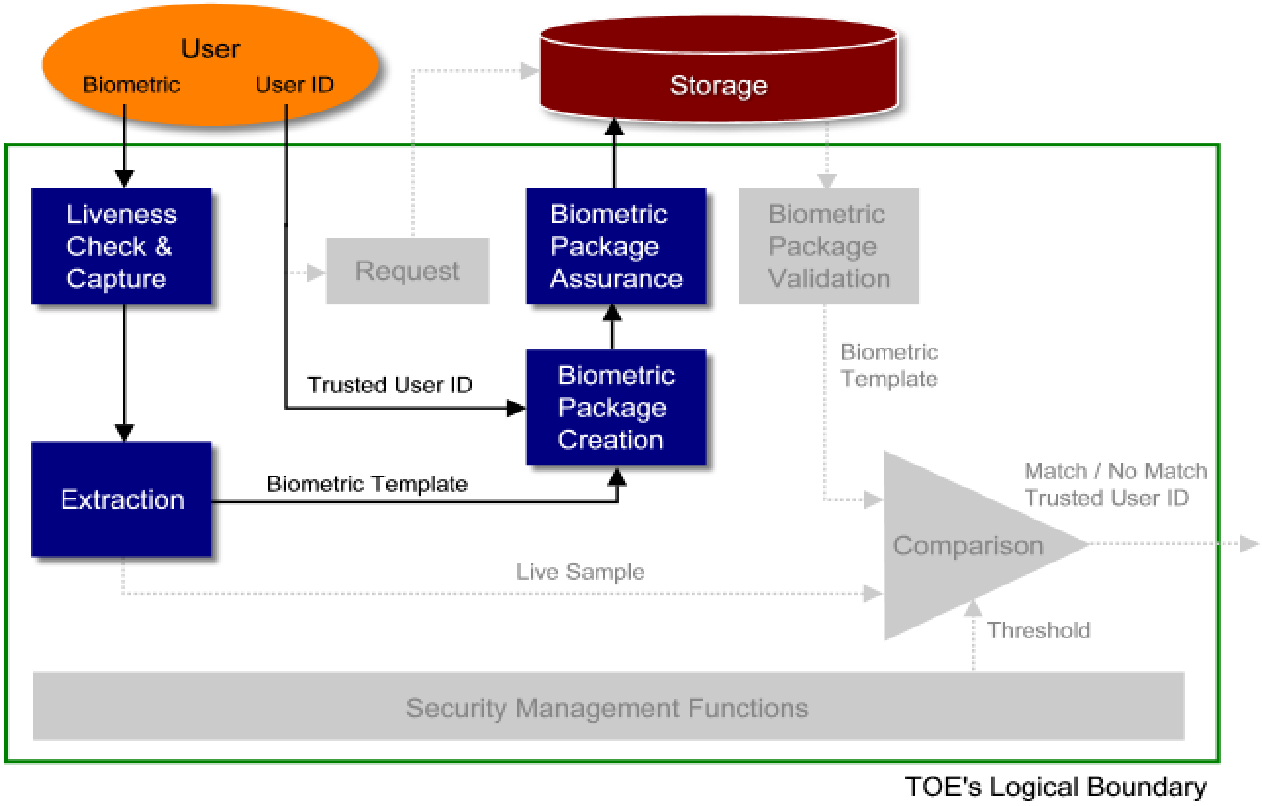


Figure 3. Block diagram of the enrollment process.

During enrollment, a biometric package is created that binds the *trusted user identifier* with the biometric template(s). It may include additional information if the TOE developer wishes, such as access privileges. After enrollment, the biometric package may be stored locally within the TOE, or on a storage device outside the TOE. The storage of biometric packages is outside the scope of this protection profile. Since the storage of the biometric packages is outside of the TOE's scope of control, cryptographic methods are used to ensure confidentiality of the biometric package is maintained, and to detect modification of the package. These methods are employed during the enrollment and verification processes.

2.1.2 The Verification Process

Figure 4 highlights the components of a TOE involved during verification. The TOE retrieves the biometric package of the user's *claimed identity* from storage, decrypts the package, and confirms that the TOE or a trusted signing authority has cryptographically signed the biometrics package.

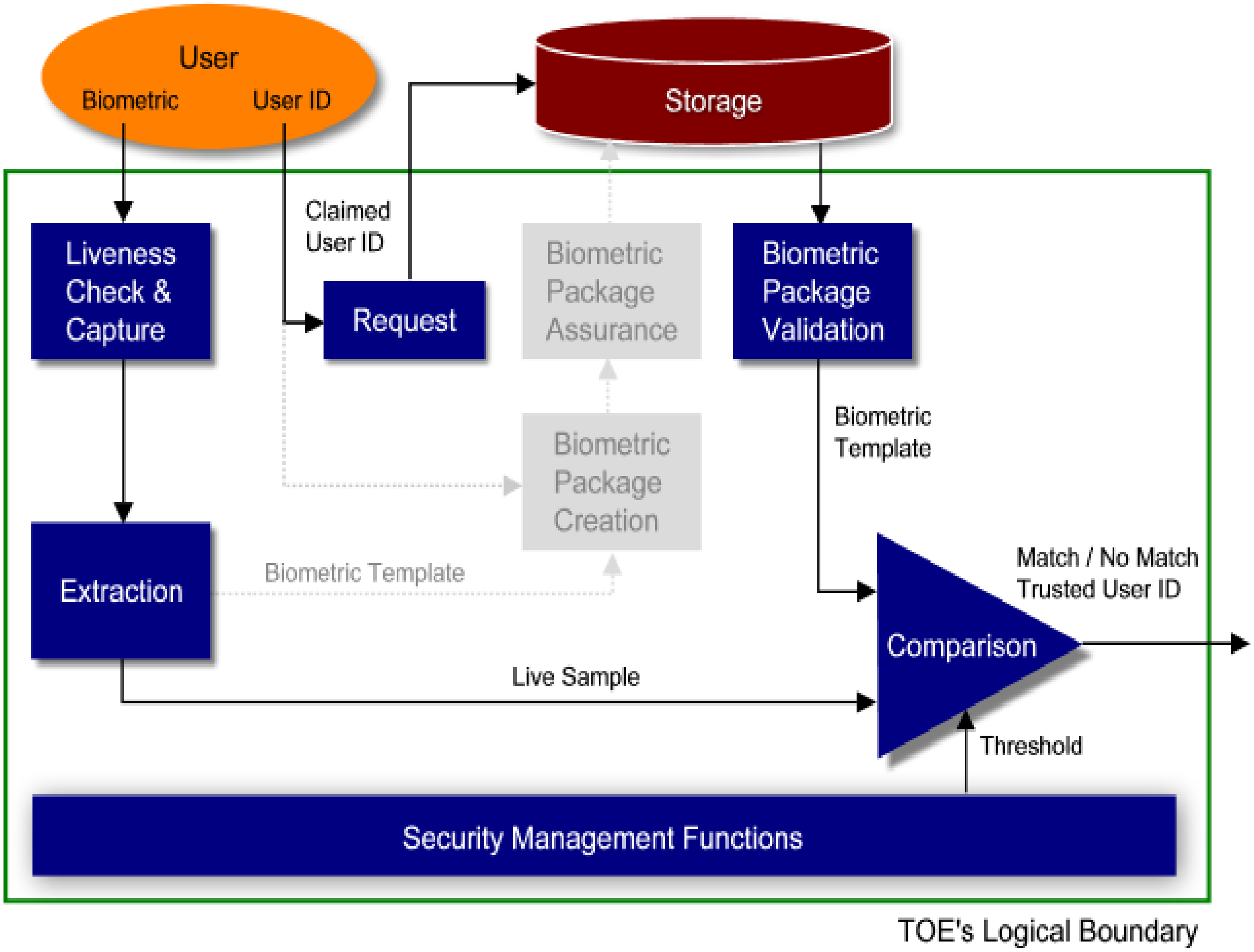


Figure 4. Verification process.

The biometric template(s) in the validated biometric package is then matched against a live sample captured from the user and a match/no-match result is generated. The Security Administrator can set a threshold range that determines the match/no-match result. However, the false acceptance and false rejection rates stated in this protection profile limit the range of acceptable values for the thresholds. The match/no-match result from the verification process is then passed to the IT environment, which will use the decision accordingly.

It is important to note the distinction between the *claimed user identifier* and *trusted user identifier*. The claimed user identifier is what the user presents to the biometrics TOE and is used to determine which biometric package to use in the verification process. The trusted user identifier is the identifier that is bound with the reference template in the biometrics package. This is a trusted user identifier, since the identity has been authenticated, whereas the claimed user identifier has not been authenticated. These two identifiers could be the same identifier (e.g., joe_user), but it is not required.

3.0 TOE SECURITY ENVIRONMENT

In trying to specify the environments in which TOEs with various levels of robustness are appropriate, it is useful to first discuss the two defining factors that characterize that environment: ***value of the resources*** and ***authorization of the entities*** to those resources.

In general terms, the environment for a TOE can be characterized by the authorization (or lack of authorization) the least trustworthy entity has with respect to the highest value of TOE resources (i.e. the TOE itself and all of the data processed by the TOE).

Note that there are an infinite number of combinations of entity authorization and value of resources; this conceptually "makes sense" because there are an infinite number of potential environments, depending on how the resources are valued by the organization, and the variety of authorizations the organization defines for the associated entities. In Section 3.3, these two environmental factors will be related to the robustness required for selection of an appropriate TOE.

3.1 Value of Resources

Value of the resources associated with the TOE includes the data being processed or used by the TOE, as well as the TOE itself (for example, a real-time control processor). "Value" is assigned by the using organization. For example, in the DoD low-value data might be equivalent to data marked "FOUO", while high-value data may be those classified Top Secret. In a commercial enterprise, low-value data might be the internal organizational structure as captured in the corporate on-line phone book, while high-value data might be corporate research results for the next generation product. Note that when considering the value of the data one must also consider the value of data or resources that are accessible through exploitation of the TOE. For example, a biometric TOE does not contain any user data that requires protection, but it may provide access to an entity with high value data. If the biometric device was being depended upon to protect the high value data, then it must be treated as a high-value-data TOE.

3.2 Authorization of Entities

Authorization that entities (e.g., users, administrators, other IT systems) have with respect to the TOE (and thus the resources of that TOE, including the TOE itself) is an abstract concept reflecting a combination of the trustworthiness of an entity and the access and privileges granted to that entity with respect to the resources of the TOE. For instance, entities that have total authorization to all data on the TOE are at one end of this spectrum; these entities may have privileges that allow them to read, write, and modify anything on the TOE, including all TSF data. Entities at the other end of the spectrum are those that are authorized to few or no TOE resources. For example, in the case of a router non-administrative entities may have their packets routed by the TOE, but that is the extent of their authorization to the TOE's resources. In the case of an OS, an entity may not be allowed to log on to the TOE at all (that is, they are not valid users listed in the OS's user database)**.**

It is important to note that authorization ***does not*** refer to the ***access*** that the entities actually have to the TOE or its data. For example, suppose the owner of the system determines that no one other than employees is authorized to certain data on a TOE, yet the owner connects the TOE to the Internet. There are millions of entities that are not ***authorized*** to the data (because they are not employees), but they actually have connectivity to the TOE through the Internet and thus can attempt to access the TOE and its associated resources.

Entities are characterized according to the value of resources to which they are authorized; the extent of their authorization is implicitly a measure of how trustworthy the entity is with respect to compromise of the data (that is, compromise of any of the applicable security policies; e.g., confidentiality, integrity, availability). In other words, in this model the greater the extent of an entity's authorization, the more trustworthy (with respect to applicable policies) that entity is.

3.3 Selection of appropriate Robustness level

Robustness is a characteristic of a TOE defining how well it can protect itself and its resources; a more robust TOE is better able to protect itself. This section relates the defining factors of IT environments, authorization, and value of resources to the selection of appropriate robustness levels.

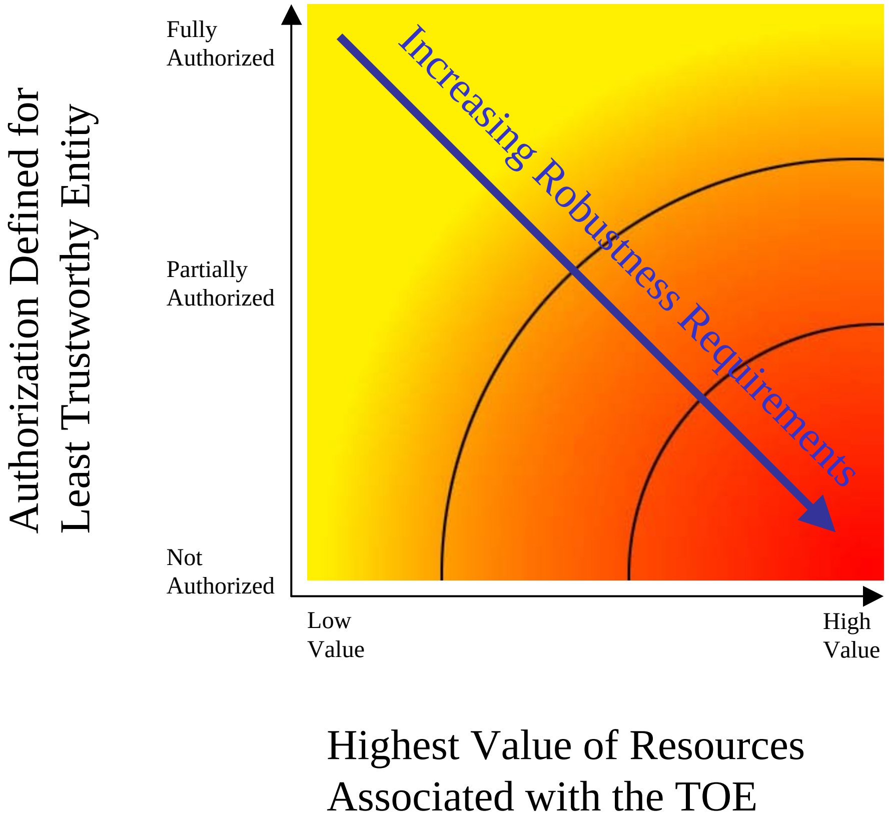
When assessing any environment with respect to Information Assurance the critical point to consider is the likelihood of an attempted security policy compromise, which was characterized in the previous section in terms of entity authorization and resource value. As previously mentioned, robustness is a characteristic of a TOE that reflects the extent to which a TOE can protect itself and its resources. It follows that as the likelihood of an attempted resource compromise increases, the robustness of an appropriate TOE should also increase.

It is critical to note that several combinations of the environmental factors will result in environments in which the likelihood of an attempted security policy compromise is similar. Consider the following two cases:

The first case is a TOE that processes only low-value data. Although the organization has stated that only its employees are authorized to log on to the system and access the data, the system is connected to the Internet to allow authorized employees to access the system from home. In this case, the least trusted entities would be unauthorized entities (e.g. non-employees) exposed to the TOE because of the Internet connectivity. However, since only low-value data are being processed, the likelihood that unauthorized entities would find it worth their while to attempt to compromise the data on the system is low and selection of a basic robustness TOE would be appropriate.

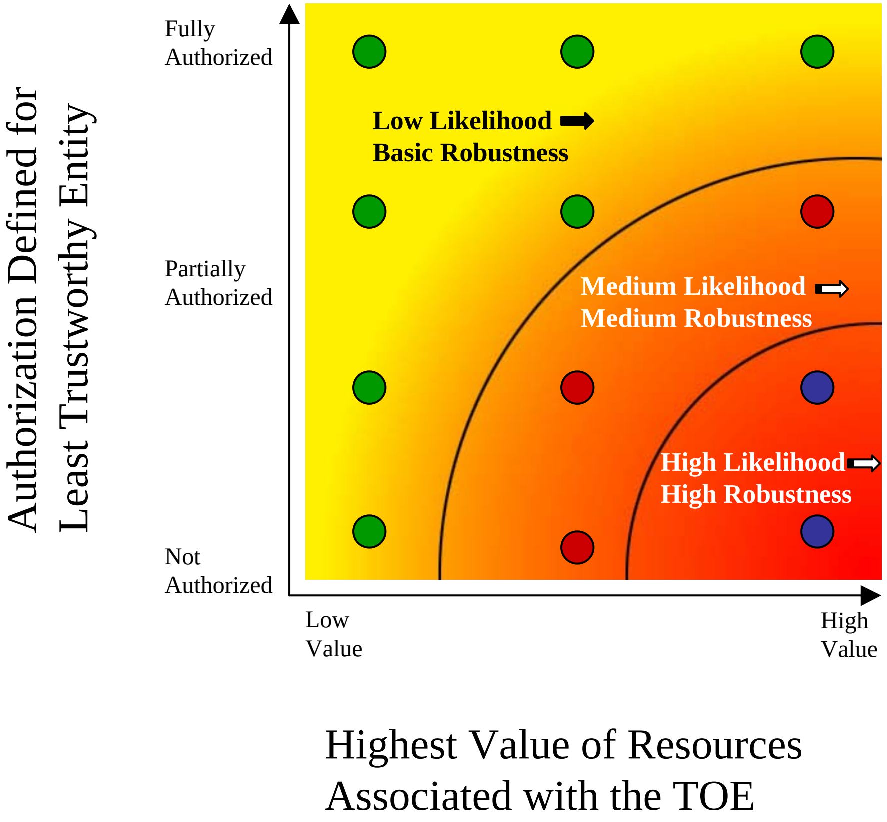
The second case is a TOE that processes high-value (e.g., classified) information. The organization requires that the TOE be stand-alone, and that every user with physical and logical access to the TOE undergo an investigation so that they are authorized to the highest value data on the TOE. Because of the extensive checks done during this investigation, the organization is assured that only highly trusted users are authorized to use the TOE. In this case, even though high value information is being processed, it is unlikely that a compromise of that data will be attempted because of the authorization and trustworthiness of the users and once again selection of a basic robustness TOE would be appropriate.

The preceding examples demonstrated that it is possible for radically different combinations of entity authorization/resource values to result in a similar likelihood of an attempted compromise. As mentioned earlier, the robustness of a system is an indication of the protection being provided to counter compromise attempts. Therefore, a basic robustness system should be sufficient to counter compromise attempts where the likelihood of an attempted compromise is low. The following chart depicts the "universe" of environments characterized by the two factors discussed in the previous section: on one axis is the authorization defined for the least trustworthy entity, and on the other axis is the highest value of resources associated with the TOE.



As depicted in this figure, the robustness of the TOEs required in each environment steadily increases as one goes from the upper left of the chart to the lower right; this corresponds to the need to counter increasingly likely attack attempts by the least trustworthy entities in the environment. Note that the shading of the chart is intended to reflects the notion that different environments engender similar levels of "likelihood of attempted compromise", signified by a similar color. Further, the delineations between such environments are not stark, but rather are finely grained and gradual.

While it would be possible to create many different "levels of robustness" at small intervals along the “Increasing Robustness Requirements” line to counter the increasing likelihood of attempted compromise due to those attacks, it would neither be practical nor particularly useful. Instead, in order to implement the robustness strategy where there are only three robustness levels: Basic, Medium, and High, the graph is divided into three sections, with each section corresponding to set of environments where the likelihood of attempted compromise is roughly similar. This is graphically depicted in the picture above.



In this second representation of environments and the robustness plane, the “dots” represent given instantiations of environments; like-colored dots define environments with a similar likelihood of attempted compromise. Correspondingly, a TOE with a given robustness should provide sufficient protection for environments characterized by like-colored dots. In choosing the appropriateness of a given robustness level TOE PP for an environment, then, the user must first consider the lowest authorization for an entity as well as the highest value of the resources in that environment. This should result in a “point” in the chart above, corresponding to the likelihood that that entity will attempt to compromise the most valuable resource in the environment. The appropriate robustness level for the specified TOE to counter this likelihood can then be chosen.

The difficult part of this activity is differentiating the authorization of various entities, as well as determining the relative values of resources; (e.g., what constitutes “low value” data vs. “medium value” data). Because every organization will be different, a rigorous definition is not possible. In Section 3.6 of this PP, the targeted threat level for a medium robustness biometric device operating in a verification mode is characterized. This information is provided to help organizations insure that the functional requirements specified by this medium robustness PP are appropriate for their intended application of a compliant biometric authentication device.

It is important to note to vendors and end users that any IT entity that is used to protect National Security information, and employs cryptography as a protection mechanism, will require the TOE's key management techniques to be approved by NSA when the TOE is fielded.

The remainder of this section addresses the following:

1. Biometric specific environment issues;

2. Assumptions about the security aspects of a compliant TOE environment;

3. Threats to TOE which are addressed by the TOE; and

4. Organizational security policies that compliant TOEs must enforce.

3.4 Biometric TOE Environment

Biometric technology is somewhat different than other IT technologies in that the inputs to the TOE are not perfectly repeatable in practice. That is, one biometric sample from an individual will not be exactly the same as a corresponding sample from the same individual a few seconds or minutes (let alone years) later. Therefore certain performance requirements for the TOE are stated in terms of probabilities. These probabilities must account not only for variations in the TOE's performance, but also for natural variation in the inputs to the TOE.

The end-user must take into consideration the trade-offs between using a biometric device versus another form of authentication. Biometrics may offer a convenient means of authentication since users are not required to remember a password that is not easily guessable. Biometrics also offers an advantage in that it may be more difficult to perform a brute force attack against a user's account than with a password mechanism. The maximum false acceptance rate (1×10^{-6}) for this TOE is weaker than the probability that a password can be guessed (1×10^{-8} for the non-biometric authentication mechanism in this PP). But it may be much more difficult to prepare and present 10^6 different biometric samples than it is to enter 10^8 passwords.

However, the degree of assurance in the authentication of an individual using biometric technologies varies. In order to accommodate a wide range of technologies this PP mandates a maximum false acceptance rate. End-users should pay close attention to the provided selection in the FIA_SOS.2 requirement, as this requirement affords a product developer the ability to provide a lower false acceptance rate if appropriate for their product. Another varying factor in the quality of the authentication decision is ability of the TOE to perform a check for liveness. Various technologies may be limited in their ability to perform a liveness check and the end-user should consider this when determining the suitability of a biometric product. The FIA_UAU.5 requirement should be considered when comparing products, as the product developer fills in an assignment that states how their product performs a check for liveness of the biometric characteristic being presented to the capture device. The PP authors could not specify what takes place during a liveness test, due to the varying biometric technologies and the state of technology increasingly improving in this area.

3.5 Assumptions

The specific conditions below are assumed to exist in a PP-compliant TOE environment.

A.ENROLLMENT_APPROVAL It is assumed that sites follow appropriate procedures for validating the identity of enrolled individuals.

A.NO_GENERAL_PURPOSE There are no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the TOE.

A.OPERATING_RANGE The TOE is placed in an environment that does not exceed its normal operating range (e.g., temperature, humidity) as defined by the vendor.

3.6 Threats

In addition to helping define the robustness appropriate for a given environment, the threat agent is a key component of the formal threat statements in the PP. Threat agents are typically characterized by a number of factors such as ***expertise***, ***available resources***, and ***motivation***. Because each robustness level is associated with a variety of environments, there are corresponding varieties of specific threat agents (that is, the threat agents will have different combinations of motivation, expertise, and available resources) that are valid for a given level of robustness. The following discussion explores the impact of each of the threat agent factors on the ability of the TOE to protect itself (that is, the robustness required of the TOE).

The ***motivation*** of the threat agent seems to be the primary factor of the three characteristics of threat agents outlined above. Given the same expertise and set of resources, an attacker with low motivation may not be as likely to attempt to compromise the TOE. For example, an entity with no authorization to low value data none-the-less has low motivation to compromise the data; thus a basic robustness TOE should offer sufficient protection. Likewise, the fully authorized user with access to highly valued data similarly has low motivation to attempt to compromise the data, thus again a basic robustness TOE should be sufficient.

Unlike the motivation factor, however, the same can't be said for ***expertise***. A threat agent with low motivation and low expertise is just as unlikely to attempt to compromise a TOE as an attacker with low motivation and high expertise; this is because the attacker with high expertise does not have the motivation to compromise the TOE even though they may have the expertise to do so. The same argument can be made for ***resources*** as well.

Therefore, when assessing the robustness needed for a TOE, the motivation of threat agents should be considered a “high water mark”. *That is, the robustness of the TOE should increase as the motivation of the threat agents increases.*

Having said that, the relationship between expertise and resources is somewhat more complicated. In general, if resources include factors other than just raw processing power (money, for example), then expertise should be considered to be at the same “level” (low, medium, high, for example) as the resources because money can be used to purchase expertise. Expertise in some ways is different, because expertise in and of itself does not automatically procure resources. However, it may be plausible that someone with high expertise can procure the requisite amount of resources by virtue of that expertise (for example, hacking into a bank to obtain money in order to obtain other resources).

It may not make sense to distinguish between these two factors; in general, it appears that the only effect these may have is to lower the robustness requirements. For instance, suppose an organization determines that, because of the value of the resources processed by the TOE and the trustworthiness of the entities that can access the TOE, the motivation of those entities would be “medium”. This normally indicates that a medium robustness TOE would be required because the likelihood that those entities would attempt to compromise the TOE to get at those resources is in the “medium” range. However, now suppose the organization determines that the entities (threat agents) that are the least trustworthy have no resources and are unsophisticated. In this case, even though those threat agents have medium motivation, the likelihood that they would be able to mount a successful attack on the TOE would be low, and so a basic robustness TOE may be sufficient to counter that threat.

It should be clear from this discussion that there is no “cookbook” or mathematical answer to the question of how to specify exactly the level of motivation, the amount of resources, and the degree of expertise for a threat agent so that the robustness level of TOEs facing those threat agents can be rigorously determined. However, an organization can look at combinations of these factors and obtain a good understanding of the likelihood of a successful attack being attempted against the TOE. Each organization wishing to procure a TOE must look at the threat factors applicable to their environment; discuss the issues raised in the previous paragraph; consult with appropriate accreditation authorities for input; and document their decision regarding likely threat agents in their environment. The important general points we can make are:

1. The motivation for the threat agent defines the upper bound with respect to the level of robustness required for the TOE**.**

2. A threat agent’s expertise and/or resources that are “lower” than the threat agent’s motivation (e.g., a threat agent with high motivation but little expertise and few resources) may lessen the robustness requirements for the TOE (see next point, however**).**

3. The availability of attacks associated with high expertise and/or high availability of resources (for example, via the Internet or "hacker chat rooms") introduces a problem when trying to define the expertise of, or resources available to, a threat agent.

It is important to note that while some of the threats listed in this PP are the same as though listed in the Biometric Verification Mode PP for Basic Robustness they are not necessarily countered or mitigated in the same manner or to the same degree. The rationale section of the PP provides the details of how a threat is countered/mitigated.

3.6.1 Threats Addressed by the TOE

The following threats are addressed by the TOE and should be read in conjunction with the threat rationale section. There are other threats that the TOE does not address (e.g., malicious developer inserting a backdoor into the TOE, emissions occurring during enrollment that would allow an eavesdropper to reconstruct either the biometric sample or the generated template) and it is up to a site to determine how these types of threats apply to its environment.

T.ADMIN_ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.ADMIN_ROGUE	An administrator's intentions may become malicious resulting in user or TSF data being compromised.
T.AUDIT_COMPROMISE	A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.
T.BYPASS	An attacker may bypass any component of the biometric product and gain unauthorized authentication.
T.CRYPT_ATTACK	An attacker may defeat security functions through a cryptographic attack against the algorithm, through cryptanalysis on encrypted data, or through a brute-force attack and thereby gaining unauthorized authentication.
T.CRYPTO_COMPROMISE	A malicious user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus

	compromise the cryptographic mechanisms and the data protected by those mechanisms.
T.HIGH_QUALITY_ARTIFACT	An attacker may use a high quality artifact (e.g., artificial hand/fingerprint, life-size photograph, or other synthetic means) to gain unauthorized authentication.
T.MIMIC	An attacker may masquerade as an enrolled user by presenting their biometric characteristic that is similar, or by reproducing the biometric characteristics of the enrolled user (e.g., changing his/her voice, forging a signature, or other mean of mimicry) to gain unauthorized authentication.
T.FLAWED_DESIGN	Unintentional or intentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a malicious user or program.
T.CORRUPTED_IMPLEMENTATION	Unintentional or intentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a malicious user or program.
T.POOR_TEST	Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities.
T.REPLAY_RESIDUAL_IMAGE	An attacker may attempt to “reuse” an authorized user’s biometric residual characteristic (e.g., finger print left on capture device) to gain unauthorized access.
T.RESIDUAL_DATA	Residual biometric authentication data from a previous valid user if not cleared from memory may allow an attacker to gain unauthorized authentication.
T. REFERENCE_TEMPLATE	An attacker modifies or creates a biometric reference template in storage or transmission to/from storage to gain unauthorized authentication.

T.POOR_ENROLLMENT	An attacker may direct an attack against a low quality reference template and gain unauthorized authentication.
T.TAMPER	An attacker may modify or otherwise alter the software or hardware components, the connections between them thereby gaining unauthorized authentication.
T.MALICIOUS_TSF_ COMPROMISE	A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified, or deleted).
T.UNATTENDED_SESSION	An attacker may gain unauthorized access to an administrator's unattended session.
T.UNAUTHORIZED_ACCESS	A user may gain access to administrative functions for which they are not authorized according to the TOE security policy.
T.UNIDENTIFIED_ACTIONS	The administrator may fail to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.
T.UNKNOWN_STATE	When the TOE is initially started or restarted after a failure, design flaws, or improper configurations may cause the security state of the TOE to be unknown.

3.7 Organizational Security Policies

PP-compliant TOEs must address the organizational security policies described below.

P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.
P.ACCOUNTABILITY	The authorized users of the TOE shall be held accountable for their actions within the TOE.
P.CRYPTOGRAPHIC_ FUNCTIONS	The TOE shall provide cryptographic functions (i.e., encryption/decryption and digital signature

	operations) to maintain the confidentiality and allow for detection of modification of TSF data that is transmitted between physically separated portions of the TOE, or stored outside the TOE.
P.CRYPTOGRAPHY_ VALIDATED	Where the TOE requires FIPS-approved security functions, only NIST FIPS validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e.; encryption, decryption, signature, hashing, key distribution, and random number generation services).
P.VULNERABILITY_ANALYSIS_TEST	The TOE must undergo appropriate independent vulnerability analysis and penetration testing to demonstrate that the TOE is resistant to an attacker possessing a medium attack potential.

4.0 SECURITY OBJECTIVES

This chapter describes the security objectives for the TOE and the TOE's operating environment. The security objectives are divided between TOE Security Objectives (i.e., security objectives addressed directly by the TOE) and Security Objectives for the Operating Environment (i.e., security objectives addressed by the IT domain or by non-technical or procedural means).

4.1 TOE Security Objectives

This section defines the security objectives that are to be addressed by the TOE.

O.ROBUST_ADMIN_GUIDANCE	The TOE will provide administrators with the necessary information for secure delivery and management.
O.ADMIN_MULTIPLE_ROLE	The TOE will provide multiple administrative roles to isolate non-overlapping administrative functions.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security-relevant events associated with users.
O.AUDIT_PROTECTION	The TOE will provide the capability to protect audit information.
O.AUDIT_REVIEW	The TOE will provide the capability to selectively view audit information, and alert the administrator of identified potential security violations.
O.AUTHENTICATION	The TOE will provide a biometric authentication mechanism to authenticate users for the IT environment or non-IT environment.
O.CHANGE_MANAGEMENT	The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development.
O.CORRECT_ TSF_OPERATION	The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site.
O.CRYPTOGRAPHIC_ FUNCTIONS	The TOE shall provide cryptographic functions (i.e., encryption/decryption and digital signature

	operations) to maintain the confidentiality and allow for detection of modification of TSF data that is transmitted between physically separated portions of the TOE, or stored outside the TOE.
O.CRYPTOGRAPHY_ VALIDATED	The TOE shall use NIST FIPS 140-2 validated cryptomodules for cryptographic services implementing FIPS-approved security functions and random number generation services used by cryptographic functions.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE.
O.DOCUMENT_KEY_LEAKAGE	The bandwidth of channels that can be used to compromise key material shall be documented.
O.THOROUGH_FUNCTIONAL_ TESTING	The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements.
O.MAINT_MODE	The TOE shall provide a mode from which recovery or initial startup procedures can be performed.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
O.RESIDUAL_INFORMATION	The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated or upon completion of a function that residual biometric data could not be reused.
O.SELF_PROTECTION	The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.
O.SOUND_DESIGN	The design of the TOE will be the result of sound design principles and techniques; the design of the TOE, as well as the design principles and techniques, are adequately and

	accurately documented.
O.SOUND_IMPLEMENTATION	The implementation of the TOE will be an accurate instantiation of its design, and is adequately and accurately documented.
O.TIME_STAMPS	The TOE shall provide reliable time stamps and the capability for the administrator to set the time used for these time stamps.
O.ROBUST_TOE_ACCESS	The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate
O.VULNERABILITY_ANALYSIS_TEST	The TOE will undergo appropriate independent vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies.

4.2 Security Objectives for the Operating Environment

This section defines the security objectives that are to be addressed by non-technical or procedural means. All of the assumptions stated in Section 3.4 are considered to be security objectives for the environment. The mapping and rationale for the security objectives are described in Section 6.

OE.ENROLLMENT_APPROVAL	Sites follow appropriate procedures for validating the identity of enrolled individuals.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the TOE.
OE.OPERATING_RANGE	The TOE is placed in an environment that does not exceed its normal operating range (e.g., temperature, humidity) as defined by the vendor.

5.0 IT SECURITY REQUIREMENTS

This section provides functional and assurance requirements that must be satisfied by a Protection Profile-compliant TOE. These requirements consist of functional components from Part 2 of the CC and assurance components from Part 3 of the CC.

5.1 TOE Functional Security Requirements

This section provides functional and assurance requirements that must be satisfied by a PP-compliant TOE. These requirements consist of components from the CC Part 2 and Part 3, NIAP interpreted requirements, and extended requirements. Table 5.1 summarizes the TOE Functional Requirements to meet the stated objectives. Table 5.2 identifies the extended requirements that were necessary to express the desired functionality.

As a vehicle for providing a further understanding of and context for security requirements, *Application Notes* have been selectively added to this PP. When they appear in the text, these follow either an element, a component, or set of components. In certain cases, the SFRs need interpretation to deal with particular characteristics of biometric systems or to convey the PP author's intent of an SFR, including any left open assignments or selections. Advice on interpretation is provided in the form of application notes where the authors felt it appropriate.

Table 5.1 - Security Functional Requirements

Functional Components (from CC Part 2)	
FAU_ARP.1	Security alarms
FAU_SAR.1	Audit review
FAU_SAR.2	Restricted audit review
FAU_SAR.3	Selectable audit review
FAU_STG.3	Action in case of possible audit data loss
FCS_CKM.1(1)	Cryptographic Key Generation (for symmetric keys)
FCS_CKM.1(2)	Generation (for asymmetric keys)
FCS_CKM.2	Cryptographic Key Distribution
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1(1)	Cryptographic Operation (for data encryption/decryption)
FCS_COP.1(2)	Cryptographic Operation (for cryptographic signature)

Functional Components (from CC Part 2)	
FCS_COP.1(3)	Cryptographic Operation (for cryptographic hashing)
FCS_COP.1(4)	Cryptographic Operation (for cryptographic key agreement)
FDP_RIP.2	Full residual information protection
FIA_ATD.1	User attribute definition
FIA_SOS.1	Verification of secrets
FIA_SOS.2	TSF Generation of secrets
FIA_UAU.2	User authentication before any action
FIA_UAU.5	Multiple authentication mechanisms
FIA_UAU.7	Protected authentication feedback
FIA_UID.2	User identification before any action
FMT_MOF.1(1)	Management of security functions behavior (audit selection)
FMT_MOF.1(2)	Management of security functions behavior (audit review)
FMT_MOF.1(3)	Management of security functions behavior (alarms)
FMT_MOF.1(4)	Management of security functions behavior (TSF non-Cryptographic Self-test)
FMT_MOF.1(5)	Management of security functions behavior (Cryptographic Self-test)
FMT_MOF.1(6)	Management of security functions behavior (Maintenance Mode)
FMT_MOF.1(7)	Management of security functions behavior (Enrollment)
FMT_MOF.1(8)	Management of security functions behavior (non-biometric Authentication Mechanism)
FMT_MOF.1(9)	Management of security functions behavior (Biometric Authentication Mechanism)
FMT_MTD.1(1)	Management of TSF data (cryptographic TSF data)

Functional Components (from CC Part 2)	
FMT_MTD.1(2)	Management of TSF data (time TSF data)
FMT_MTD.1(3)	Management of TSF data (Authentication Mechanism Data)
FMT_REV.1	Revocation
FMT_SMR.2	Restrictions on security roles
FPT_ITT.1(1)	Basic internal TSF data transfer protection (from disclosure)
FPT_ITT.1(2)	Basic internal TSF data transfer protection (from undetected modification)
FPT_PHP.3	Resistance to physical attack
FPT_TST.1(1)	TSF Testing (for cryptography)
FPT_TST.1(2)	TSF Testing (for key generation components)
FPT_STM.1	Reliable time stamps
FTA_SSL.3	TSF-initiated termination
FTA_TAB.1	Default TOE access banners
FTA_TSE.1	TOE session establishment

Table 5.2 - Extended Security Functional Requirements

Extended Functional Components	
FCS_BCM_(EXT).1	Baseline Cryptographic Module
FCS_CKM_ (EXT).2	Cryptographic Key Handling and Storage
FCS_COP_(EXT).1	Random Number Generation
FIA_ENROLL_(EXT).1	Enrollment
FMT_MTD_(EXT).1	Management of TSF data (Capture device unique identifier)
FPT_ITC_(EXT). 1	TSF confidentiality

Extended Functional Components	
FPT_ITI_(EXT).1	TSF detection of modification
FPT_PHP_(EXT).1	Detection of physical attack
FPT_TST_(EXT).1	TSF testing
FAU_GEN.1-NIAP-0410	Audit data generation
FAU_GEN.2-NIAP-0410	User identity association
FAU_SAA.1-NIAP-0407	Potential violation analysis
FAU_SEL.1-NIAP-0407	Selective audit
FAU_STG.1-NIAP-0423	Protected audit trail storage
FAU_STG.NIAP-0414-1-NIAP-0429	Site-Configurable Prevention of Audit Loss
FIA_AFL.1-NIAP-0425(1)	Authentication failure handling (Against a single non-administrative user identifier)
FIA_AFL.1-NIAP-0425(2)	Authentication failure handling (Consecutive failed attempts)
FIA_AFL.1-NIAP-0425(3)	Authentication failure handling (Administrator Users)
FIA_USB.1-NIAP-0415	User-subject binding
FPT_RCV.2-NIAP-406	Recovery from Failure

5.1.1 Security Audit (FAU)

FAU_ARP.1 Security alarms

FAU_ARP.1.1 – **Refinement:** The TSF shall

a) [generate an alarm by [assignment: method determined by the ST Author to generate the alarm],

b) block any further authentication attempts until the Security Administrator defined time period has elapsed, or an action is taken by the Security Administrator,

c) stop ongoing and prevent further enrollment activity until the Security Administrator takes some action,]

upon detection of a potential security violation.

Application Note: The TOE generates an alarm by a method determined by the ST Author. Acceptable methods may include sending an email, paging the Security Administrator, sending a message to an administrative console, sounding an audible alarm (e.g., bell, siren) or providing a visual alarm, such as a flashing light. The intent of this requirement is to alert an administrator that the TOE has encountered a potential security violation. While some implementations may provide an alarm that communicates an alarm condition more effectively to an administrator than other implementations, the PP does not want to exclude devices that may not be able to "immediately alert" an administrator (e.g., stand alone TOEs with no connectivity). The intent in b) is to provide the Security Administrator the choice of preventing the TOE from authenticating users until the Security Administrator takes some action (e.g., enable the TOE to perform authentication, clear the alarm and the TOE implicitly can resume performing authentication), or define a time period in which the TOE can begin performing authentication again. The time period should allow the flexibility of allowing the administrator to "throttle" throughput (e.g., a few minutes) or to assess the alarm and take the appropriate action (e.g., a few hours). The TOE may additionally send an alarm to the host IT environment to signify a potential security violation, but simply signaling the IT environment does not satisfy the intent of this requirement.

FAU_GEN.1-NIAP-0410 Audit data generation

FAU_GEN.1.1-NIAP-0410 – **Refinement:** The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events **listed in Table 5.3**; and

c) **[selection: [assignment: events at a basic level of audit introduced by the inclusion of additional SFRs determined by the ST Author], [assignment: events commensurate with a basic level of audit introduced by the inclusion of extended requirements determined by the ST Author], no additional events].**

Application Note: For the first assignment in the selection, the ST author augments the table (or lists explicitly) the audit events associated with the basic level of audit for any

SFRs that the ST author includes that are not included in this PP.

Likewise, for the second assignment the ST author includes audit events that may arise due to the inclusion of any **extended** requirements not already in the PP. Because “basic” audit is not defined for such requirements, the ST author will need to determine a set of events that are commensurate with the type of information that is captured at the basic level for similar requirements. It is acceptable for the ST author to chose "no additional events", if the ST author has not included additional requirements, or has included additional requirements that do not have a basic level (or commensurate level) of audit associated with them..

Table 5.3 Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_ARP.1	Potential security violation was detected	Identification of the event(s) caused the generation of the alarm
FAU_GEN.1-NIAP-0410	None	
FAU_GEN.2-NIAP-0410	None	
FAU_SAA.1-NIAP-0407	Attempts to enable/disable of any of the analysis mechanisms	The trusted user identity of the administrator performing the function
FAU_SAR.1	Attempts to open the audit trail	The trusted user identity of the administrator performing the function
FAU_SAR.2	Attempts to read information from the audit records	The trusted user identity of the administrator performing the function
FAU_SAR.3	None	
FAU_SEL.1-NIAP-0407	Attempts to modify the audit configuration	The trusted user identity of the administrator performing the function
FAU_STG.1-NIAP-0423	Attempts to backup and delete the audit trail	The trusted user identity of the administrator performing the function

Requirement	Auditable Events	Additional Audit Record Contents
FAU_STG.3	Reaching the defined percentage of storage capacity; Actions taken due to exceeding the threshold	The Audit Administrator defined percentage of storage capacity; The action to be taken if the audit trail becomes full
FAU_STG.NIAP-0414-1-NIAP-0429	None	
FCS_BCM_(EXT).1	None	
FCS_CKM.1(1)	Failure of the activity	
FCS_CKM.1(2)	Failure of the activity	
FCS_CKM _(EXT).2	Failure of the activity	
FCS_CKM.2	None	
FCS_CKM.4	None	
FCS_COP_(EXT).1	Failure of cryptographic operation	Type of cryptographic operation Any applicable cryptographic mode(s) of operation, excluding any sensitive information
FCS_COP.1(1)	Failure of cryptographic operation	Type of cryptographic operation Any applicable cryptographic mode(s) of operation, excluding any sensitive information
FCS_COP.1(2)	Failure of cryptographic operation	Type of cryptographic operation Any applicable cryptographic mode(s) of operation, excluding any sensitive information
FCS_COP.1(3)	Failure of cryptographic operation	Type of cryptographic operation Any applicable cryptographic mode(s) of operation, excluding

Requirement	Auditable Events	Additional Audit Record Contents
		any sensitive information
FCS_COP.1(4)	Failure of cryptographic operation	Type of cryptographic operation Any applicable cryptographic mode(s) of operation, excluding any sensitive information
FDP_RIP.2	None	
FIA_AFL.1-NIAP-0425(1)	Reaching the specified number of failed authentication attempts; The action (e.g. disabling of an account, timeout) taken; The subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of an account)	Claimed identity of the unsuccessfully authenticated user; Trusted user identity of the Security Administrator (if applicable) that took action to re-enable an account; Period of timeout (if applicable)
FIA_AFL.1-NIAP-0425(2)	Reaching the specified number of failed authentication attempts; The action (i.e., disabling of authentication at the offending capture device, timeout) taken; The subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of authentication at the capture device)	Claimed identity of the unsuccessfully authenticated user(s)[2]; Trusted user identity of the Security Administrator (if applicable) that took action to re-enable an account; Period of timeout (if applicable)
FIA_AFL.1-NIAP-0425(3)	Reaching the specified number of failed authentication attempts;	Claimed identity of the unsuccessfully authenticated administrator;

[2] For this requirement, there may be multiple user identifiers associated with this event, and the audit record contains all user identifiers that generated the event.

Requirement	Auditable Events	Additional Audit Record Contents
	The action (e.g. disabling of an account, timeout) taken; The subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of an account)	Trusted user identity of the Security Administrator (if applicable) that took action to re-enable an account; Period of timeout (if applicable)
FIA_ATD.1	None	
FIA_UAU.1	None	
FIA_SOS.1	None.	
FIA_SOS.2	None.	

Requirement	Auditable Events	Additional Audit Record Contents
FIA_UAU.2	None.	
FIA_UAU.5	All use of the authentication mechanism(s)	Claimed identity of the user attempting to authenticate using the biometric authentication mechanism; Trusted user identifier of a successfully authenticated user; Unique identity of the capture device[3]; Result of liveness check; Identity of the IT entity that digitally signed the biometrics package; Comparison score of a non-match decision; Identity of the user presented at the non-biometric authentication mechanism
FIA_UAU.7	None.	
FIA_UID.2	All use of the user identification mechanism, including the user identity provided	
FIA_USB.1-NIAP-0415	Success and failure of binding of user security attributes to a subject	The trusted user identity of the user whose attributes are attempting to be bound
FMT_MOF.1(1)	All attempts to enable, disable,	The trusted user identity of the

[3] The TOE has the ability to uniquely identify the capture device. If the TOE has multiple capture devices this identifier aids the Administrator in determining where the offending action took place. The unique identifier could be a identifier that is transmitted to the TOE, or could be associated with the capture device based on how it is connected to the TOE (e.g., a capture device is uniquely assigned to a physical port on a server component of the TOE).

Requirement	Auditable Events	Additional Audit Record Contents
	determine, or modify the behavior of the audit generation functions in the TSF	administrator performing the function
FMT_MOF.1(2)	All attempts to enable, or modify the behavior of the audit review functions in the TSF	The trusted user identity of the administrator performing the function
FMT_MOF.1(3)	All attempts to modify the behavior of the alarm and analysis functions in the TSF	The trusted user identity of the administrator performing the function
FMT_MOF.1(4)	All attempts to modify the behavior of the self-tests functions in the TSF	The trusted user identity of the administrator performing the function
FMT_MOF.1(5)	All attempts to enable or disable the cryptographic self-tests after key generation in the TSF	The trusted user identity of the administrator performing the function
FMT_MOF.1(6)	None	
FMT_MOF.1(7)	All attempts to determine, or modify the behavior of the enrollment functions in the TSF	The trusted user identity of the administrator performing the function
FMT_MOF.1(8)	All attempts to enable and disable the non-biometric authentication mechanism	The trusted user identity of the administrator performing the function
FMT_MOF.1(9)	All attempts to modify or determine the behavior of the biometric authentication mechanism	The trusted user identity of the administrator performing the function; Any state change (enable/disable) of the liveness check
FMT_MTD.1(1)	All attempts to modify the cryptographic security data	The trusted user identity of the administrator performing the function
FMT_MTD.1(2)	All attempts to set the time and date used to form the time	The trusted user identity of the administrator performing the

Requirement	Auditable Events	Additional Audit Record Contents
	stamps	function
FMT_MTD.1(3)	All attempts to query and set the authentication mechanism data	The trusted user identity of the administrator performing the function
FMT_MTD_(EXT).1	All attempts to set the capture device identifier, if applicable [4]	The trusted user identity of the administrator performing the function
FMT_REV.1	All attempts to revoke security attributes	List of security attributes that were attempted to be revoked The trusted user identity of the administrator performing the function
FMT_SMR.2	All attempts to modify the group of users that are associated with a role	Trusted user identifiers that are associated with the modifications The trusted user identity of the administrator performing the function
FPT_ITT.1(1)	None	
FPT_ITT.1(2)	None	
FPT_PHP.3	None	
FPT_RCV.2-NIAP-406	The fact that a failure or service discontinuity occurred; Resumption of the regular operation;	Type of failure or service discontinuity
FPT_STM.1	Changes to the time and date	Previous time and date; New time and date

[4] If the TOE does not provide the capability to set the capture device identifier (i.e., the TOE hardwires a unique identifier in the capture device) then this audit event is not applicable.

Requirement	Auditable Events	Additional Audit Record Contents
FTA_SSL.3	The termination of a remote session by the session locking mechanism	The trusted user identity of the administrator associated with the session that was terminated
FTA_TAB.1	None	
FTA_TSE.1	All attempts at establishment of an administrator session	The claimed identity of the user attempting to establish the session For unsuccessful attempts, the reason for denial of the establishment attempt
FIA_ENROLL_(EXT).1	All attempts to create a reference template, refreshing reference templates, or adding additional reference templates to a biometric package; All attempts to modify a reference template while resident in the TOE;	Trusted user identity of the administrator attempting to create/modify a reference template; The enrolled user's user identifier.
FPT_ITC_(EXT). 1	Any failure to decrypt a biometric package	Claimed user identifier of the associated biometric package
FPT_ITI_(EXT).1	Detection of modification of the biometric package	User identifier of the associated biometric package
FPT_PHP_(EXT).1	Exposure of internal TOE components.	
FPT_TST.1(1)	TSF testing (for cryptography)	Self-test that failed; The affected TSF components, including the TSF software and TSF data where modification was detected
FPT_TST.1(2)	TSF testing (for key generation)	Self-test that failed; The affected TSF components, including the TSF software and

Requirement	Auditable Events	Additional Audit Record Contents
		TSF data where modification was detected
FPT_TST_(EXT).1	TSF testing	Self-test that failed; The affected TSF components, including the TSF software and TSF data where modification was detected

FAU_GEN.1.2-NIAP-0410 – **Refinement:** The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event **(if applicable)**; and

b) For each audit event **type**, based on the auditable event definitions of the functional components included in the PP/ST, *information specified in column three in Table 5.3*.

Application Note: A subject identity is distinct from a user identifier (trusted or claimed). A subject identity is typically an active entity that is acting on behalf of a user (e.g., a process, in which case the process id would be the subject identity). In general, this subject may be a trusted subject or an untrusted subject. In this TOE there are two types of users: the untrusted users, which only have limited access to the TOE (i.e., present their biometric characteristic to the capture device); and trusted users, which are the administrators that administer the TOE. Since the untrusted users have limited interaction with the TOE, this TOE only has trusted subjects. The intent of requiring the identity of a trusted subject resulting from an authentication event is to provide information on which authentication mechanism(s) was used. The thought is that the biometric authentication mechanism(s) and the additional administrator authentication mechanism may have distinct subject identities, which could provide the Audit Administrator valuable information. In limited cases the subject identity or outcome does not apply (i.e., FPT_PHP_(EXT).1)

FAU_GEN.2-NIAP-0410 User Identity Association

FAU_GEN.2.1-NIAP-0410 –For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

Application Note: The claimed user identifier may not be associated with a biometrics package (e.g., an invalid claimed user identifier was presented), however, the supplied claimed user identifier is captured in the audit record. This requirement applies somewhat differently depending on the type of user (i.e., untrusted user, administrator). For untrusted users, the TOE associates auditable events to a claimed user identifier that is supplied when a user attempts to authenticate. This claimed identifier may not be the same as the trusted user identifier (the one bound to the reference template) and this case is different than administrative users, because the TOE may have no knowledge of the human user associated with the supplied user identifier. This is because untrusted users may have been enrolled on a different TOE. However, the TOE is always able to associate the trusted user identifier of administrators with human users, since administrative users are "registered" in the TOE as required by FIA_ATD.1.

FAU_SAA.1-NIAP-0407 Potential violation analysis

FAU_SAA.1.1-NIAP-0407 – The TSF shall be able to apply a set of rules in monitoring events and based upon these rules indicate a potential violation of the TSP.

FAU_SAA.1.2-NIAP-0407 – **Refinement**: The TSF shall enforce the following rules for monitoring events:

a) Accumulation of [

- a Security Administrator specified number of authentication failures against a single non-administrative user identifier,
- a Security Administrator specified number of consecutive failed authentication attempts,
- a Security Administrator specified number of authentication failures against an administrative user identifier;

b) Any failure of the cryptographic self-tests;

c) Any failure of the other TSF self-tests;

d) Any failure to generate a cryptographic key;

e) Detection of physical attack;

f) Any failure to decrypt a biometrics package;

g) Detection of modification of a biometrics package];

h) [selection: [assignment: any other rules], "no additional rules"].

Application Note: The intent of this requirement is that an alarm is generated (FAU_ARP.1) once the threshold for the event in (a) is met. Once the alarm has been generated it is assumed that the "count" for that event is reset to zero. The Security Administrator settable number of authentication failures in (a) is intended to be the same value as specified in the iterations of FIA_AFL.1.1-NIAP-0425(1) – (3).

FAU_SAR.1 Audit review

FAU_SAR.1.1 - The TSF shall provide [the Audit Administrator and Security Administrator] with the capability to read [all audit information] from the audit records.

FAU_SAR.1.2 – **Refinement**: The TSF shall provide the audit records in a manner suitable for the **Audit Administrator and Security Administrator** to interpret the information.

FAU_SAR.2 Restricted audit review

FAU_SAR.2.1 – **Refinement**: The TSF shall prohibit all users read access to the audit records, except **the Audit Administrator and Security Administrator**.

FAU_SAR.3 Selectable audit review

FAU_SAR.3.1 – **Refinement:** The TSF shall provide the ability to perform *searches and sorting* of audit data based on:

a) [user identifier;

b) subject identity;

c) reference template creation;

d) ranges of one or more: dates, times;

e) events that generate an alarm; and

f) [selection: [assignment: other criteria determined by the ST Author], no additional criteria]].

Application Note: The Audit Administrator and Security Administrator are the only users who can perform these functions, since they are the only users with read access to the audit records in the audit trail. Audit data should be capable of being searched and sorted on all criteria specified in a – e, if applicable (i.e., not all criteria will exist in all audit records). Sorting means to arrange the audit records such that they are “grouped” together for administrative review. For example the Audit Administrator may want all the audit records for a specified time period presented together to facilitate their audit review. In item (e), these are the events specified in FAU_SAA.1. If no additional criteria are provided by the TOE to perform searches or sorting of audit data, the ST author selects “no additional criteria”.

FAU_SEL.1-NIAP-0407 Selective Audit

FAU_SEL.1.1-NIAP-0407 - **Refinement**: The TSF shall **allow only the Audit Administrator** to include or exclude auditable events from the set of audited events based on the following attributes:

a) *user identity*;

b) *event type*;

c) [success of auditable events;

d) failure of auditable events; and

e) [selection: [assignment: list of additional criteria that audit selectivity is based upon], no additional criteria]].

Application Note: “event type” is to be defined by the ST author; the intent is to be able to include or exclude classes of audit events. While the Audit Administrator has the capability to “pre-select” audit events, this does not mean that the Audit Administrator has the capability to implicitly disable alarm events (FAU_SAA.1). If the Audit Administrator de-selects an event listed in FAU_SAA.1 that event will still generate an alarm if the Security Administrator has enabled that event(s) to generate an alarm.

FAU_STG.1-NIAP-0423 Protected audit trail storage

FAU_STG.1.1-NIAP-0423 – **Refinement:** The TSF shall **restrict** the **backup and deletion of stored** audit records **in the audit trail to the Audit Administrator**.

FAU_STG.1.2-NIAP-0423 - **Refinement**: The TSF shall *prevent* modifications to the audit records in the audit trail.

FAU_STG.3 Action in case of possible audit data loss

FAU_STG.3.1 - **Refinement**: The TSF shall [generate an alarm by [assignment: method determined by the ST Author to generate the alarm]], if the audit trail exceeds [an Audit Administrator settable percentage of storage capacity].

Application Note: As with FAU_ARP.1, the TSF generates an alarm to indicate that the audit trail has reached the Audit Administrator defined percentage of storage capacity.

FAU_STG.NIAP-0414-1-NIAP-0429 Site-Configurable Prevention of Audit Loss

FAU_STG.NIAP-0414-1.1-NIAP-0429 - **Refinement**: The TSF shall provide the **Audit Administrator** the capability to select one of the following actions: *prevent auditable events, except those taken by the Audit Administrator, overwrite the oldest stored audit records* **or** [selection: [assignment: other actions to be taken in case of audit storage failure], no other actions] to be taken if the audit trail is full.

Application Note: The TOE provides the Audit Administrator the option of preventing audit data loss by preventing auditable events from occurring, except those actions taken by the Audit Administrator. This means that **only** the Audit Administrator can successfully be authenticated. The Audit Administrator actions under these circumstances are not required to be audited, since a user acting in this role will have to perform some action to manage the audit trail and address the problem. The TOE also provides the Audit Administrator the option of overwriting "old" audit records rather than preventing auditable events, which may protect against a denial-of-service attack.

FAU_STG.NIAP-0414-1.2-NIAP-0429 **Refinement**: The TSF shall **as a default** *prevent auditable events, except those taken by the* ***Audit Administrator*** if the audit trail is full.

Application Note: While FAU_STG.NIAP-0414-1.1-NIAP-0429 provides the Audit Administrator with the capability to select the TOE's behavior when the audit trail is full, this requirement ensures that as a default, audit records are not lost when the audit trail becomes full.

5.1.2 Cryptographic Support (FCS)

This section specifies the cryptographic support required in the TOE. Evolving public standards on cryptographic functions and related areas have required an interim approach to writing cryptographic requirements. These cryptographic requirements are expected to be achievable in commercial products in the near term, and gradually mature over time. Today these requirements represent a step in the direction of helping to improve the security in COTS products. Over time, the Protection Profile will be updated as the underlying public standards and the body of related special publications mature.

5.1.2.1 Extended: Baseline Cryptographic Module (FCS_BCM_(EXT))

The cryptographic requirements are structured to accommodate use of the FIPS 140-2 standard and NIST's Cryptomodule Validation Program (CMVP) in meeting the requirements. Note that *FIPS-approved* cryptographic functions are required to be implemented in a *FIPS-validated module running in FIPS-approved mode*. FCS_BCM reflects this requirement, and it specifies the required FIPS validation levels for the security functions. Note also that some of the requirements of this Protection Profile go beyond what is required for FIPS 140-2 validation.

Application Note: A FIPS-approved cryptographic function is a security function (e.g., cryptographic algorithm, cryptographic key management technique, or authentication technique) that is either: 1) specified in a Federal Information Processing Standard (FIPS), or 2) adopted in a FIPS and specified either in an appendix to the FIPS or in a document referenced by the FIPS.

Extended: Baseline Cryptographic Module (FCS_BCM_(EXT).1)

FCS_BCM_(EXT).1.1 **All FIPS-approved cryptographic functions implemented by the TOE shall be implemented in a cryptomodule that is FIPS 140-2 validated, and perform the specified cryptographic functions in a FIPS-approved mode of operation. The FIPS 140-2 validation shall include an algorithm validation certificate for all FIPS-approved cryptographic functions implemented by the TOE.**

Application Note: This Protection Profile shall use the term "FIPS 140-2" for simplicity. FIPS PUB 140-2 is currently undergoing a regular five year review; in the near future, FIPS PUB 140-3 will supersede it. Security Targets written to comply with this Protection Profile may replace it with the successor standard that is in force at the time of evaluation.

Application Note: This requirement does not preclude additional cryptographic algorithms from being implemented in the cryptomodule, and/or used by the TOE for purposes OTHER than those explicitly stated in this Protection Profile.

FCS_BCM_(EXT).1.2 **All cryptographic modules implemented in the TOE** *[selection:*

(1) Entirely in hardware shall have a minimum overall rating of FIPS PUB 140-2, Level 3,

(2) Entirely in software shall have a minimum overall rating of FIPS PUB 140-2, Level 1 and also meet FIPS PUB 140-2, Level 3 for the following: Cryptographic Module Ports and Interfaces; Roles, Services and Authentication; Cryptographic Key Management; and Design Assurance.

(3) As a combination of hardware and software shall have a minimum overall rating of FIPS PUB 140-2, Level 1 and also meet FIPS PUB 140-2, Level 3 for the following: Cryptographic Module Ports and Interfaces; Roles, Services and Authentication; Cryptographic Key Management; and Design Assurance.]

Application Note: "Combination of hardware and software" means that some part of the cryptographic functionality will be implemented as a software component of the TSF. The combination of a cryptographic hardware module and a software device driver whose sole purpose is to communicate with the hardware module is considered a hardware module rather than "combination of hardware and software".

Application Note: Note that the requirements for selections (2) and (3) are the same. The ST author should make it clear how the cryptomodule is implemented.

5.1.2.2 Cryptographic Key Management (FCS_CKM)

NIST Special Publication 800-57, “Recommendation for Key Management” contains additional protection mechanisms that vendors are encouraged to implement. It should also be used as guidance for the cryptographic key management requirements.

Cryptographic Key Generation (for symmetric keys) (FCS_CKM.1(1))

FCS_CKM.1.1(1) Refinement: **The TSF shall generate symmetric cryptographic keys** using a FIPS-Approved Random Number Generator as specified in FCS_COP_(EXT).1, and provide integrity protection to generated symmetric keys in accordance with NIST SP 800-57 “Recommendation for Key Management” Section 6.1.

Application Note: NIST SP 800-57 “Recommendation for Key Management” Section 6.1 states: “Integrity protection can be provided by cryptographic integrity mechanisms (e.g. cryptographic checksums, cryptographic hashes, MACs, and signatures), non-cryptographic integrity mechanisms (e.g. CRCs, parity, etc.) […], or physical protection mechanisms.” Guidance for the selection of appropriate integrity mechanisms is given in Sections 6.2.1.2 and 6.2.2.2 of NIST SP 800-57 “Recommendation for Key Management”.

Application Note: Note that there is a separate requirement for Cryptographic Key Agreement (FCS_COP.1(4)).

Cryptographic Key Generation (for asymmetric keys) (FCS_CKM.1(2))

FCS_CKM.1.1(2) Refinement: **The TSF shall generate** asymmetric **cryptographic keys in accordance** with the mathematical specifications of the FIPS-approved or NIST-recommended standard [*assignment: specify standard(s)]*, using a domain parameter generator and *[selection:*

(1) a FIPS-Approved Random Number Generator as specified in FCS_COP_(EXT).1, and/or

(2) a prime number generator as specified in ANSI X9.80 “Prime Number Generation, Primality Testing, and Primality Certificates” using random integers with deterministic tests, or constructive generation methods]

in a cryptographic key generation scheme that meets the following:

- **The TSF shall provide integrity protection and assurance of domain parameter and public key validity to generated asymmetric keys in accordance with NIST SP 800-57 “Recommendation for Key Management” Section 6.1.**
- **Generated key strength shall be equivalent to, or greater than, a symmetric key strength of 128 bits using conservative estimates.**

Application Note: NIST SP 800-57 “Recommendation for Key Management” Section 6.1 states: “Integrity protection can be provided by cryptographic integrity mechanisms (e.g.

cryptographic checksums, cryptographic hashes, MACs, and signatures), non-cryptographic integrity mechanisms (e.g. CRCs, parity, etc.) […], or physical protection mechanisms." Guidance for the selection of appropriate integrity mechanisms is given in Sections 6.2.1.2 and 6.2.2.2 of NIST SP 800-57 "Recommendation for Key Management".

Application Note: Assurance of domain parameter and public key validity provides confidence that the parameters and keys are arithmetically correct. Guidance for the selection of appropriate validation mechanisms is given in NIST SP 800-57 "Recommendation for Key Management," NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography," and FIPS PUB 186-3, "Digital Signature Standard."

Application Note: See NIST Special Publication 800-57, "Recommendation for Key Management" for information about equivalent key strengths.

Cryptographic Key Distribution (FCS_CKM.2)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method *[selection:*

(3) Manual (Physical) Method, and/or

(4) Automated (Electronic) Method]

that meets the following:

- **NIST Special Publication 800-57, "Recommendation for Key Management" Section 8.1.5**
- **NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"**

Application Note: NIST Special Publication 800-56A "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" is only applicable when public key schemes are used in key transport methods.

Application Note: DoD applications may have additional key distribution requirements related to the DoD PKI and certificate formats.

Extended: Cryptographic Key Handling and Storage (FCS_CKM_(EXT).2)

FCS_CKM_(EXT).2.1 **The TSF shall perform a key error detection check on each transfer of key (internal, intermediate transfers).**

Application Note: A parity check is an example of a key error detection check.

FCS_CKM_(EXT).2.2 **The TSF shall store <u>persistent</u> <u>secret and private</u> keys <u>when not in use</u> in encrypted form or using split knowledge procedures.**

Application Note: Note that this requirement is stronger than the FIPS 140-2 key storage requirements, which state: "Cryptographic keys stored within a cryptographic module shall be stored in plaintext form or encrypted form."

Application Note: A persistent key, such as a file encryption key, is one that must be available in the system over long periods of time. A non-persistent key, such as a key used to encrypt or decrypt a single message or a session, is one that is ephemeral in the system.

Application Note: "When not in use" is interpreted in the strictest sense so that persistent keys only exist in plaintext form during intervals of operational necessity. For example, a file encryption key exists in plaintext form only during actual encryption and/or decryption processing of a file. Once the file is decrypted or encrypted, the file encryption key should immediately be covered for protection.

Application Note: A "split knowledge procedure" is a process by which a cryptographic key is split into multiple key components, individually sharing no knowledge of the original key, which can be subsequently input into, or output from, a cryptographic module by separate entities and combined to recreate the original cryptographic key.

FCS_CKM_(EXT)_2.3 **The TSF shall destroy non-persistent cryptographic keys after a cryptographic administrator-defined period of time of inactivity.**

Application Note: The cryptographic administrator must have the ability to set a threshold of inactivity after which non-persistent keys must be destroyed in accordance with FCS_CKM.4.

FCS_CKM_(EXT).2.4 **The TSF shall prevent archiving of expired (private) signature keys.**

Application Note: This requirement is orthogonal to typical system back-up procedures. Therefore, it does not address the problem of archiving an active (private) signature key during a system back-up and saving the key beyond its intended life span.

Cryptographic Key Destruction (FCS_CKM.4)

Application Note: Note that this requirement is stronger than the FIPS 140-2 key zeroization requirements, which state: "A cryptographic module shall provide methods to zeroize all plaintext secret and private cryptographic keys and CSPs within the module."

FCS_CKM.4.1 Refinement: **The TSF shall destroy cryptographic keys in accordance with a** cryptographic key zeroization method **that meets the following:**

a) Key zeroization requirements of FIPS PUB 140-2, "Security Requirements for Cryptographic Modules"

b) Zeroization of all plaintext cryptographic keys and all other critical cryptographic security parameters shall be immediate and complete.

Application Note: The term "immediate" here is meant to impart some urgency to the destruction: it should happen as soon as practical after the key is no longer required to be in plaintext. It is certainly permissible to complete a critical section of code before destroying the key. However, the destruction shouldn't wait for idle time, and there shouldn't be any non-determined event (such as waiting for user input) which occurs before it is destroyed.

c) The TSF shall zeroize each intermediate storage area for plaintext key/critical cryptographic security parameter (i.e., any storage, such as memory buffers, that is included in the path of such data) upon the transfer of the key/critical cryptographic security parameter to another

location.

Application Note: Item c) pertains to the elimination of internal, temporary copies of keys/parameters during processing, and not to the locations that are used for the storage of the keys, which are specified in item b). The temporary locations could include memory registers, physical memory locations, and even page files and memory dumps.

d) For non-volatile memories other than EEPROM and Flash, the zeroization shall be executed by overwriting three or more times using a different alternating data pattern each time.

Application Note: Although verification of the zeroization of each intermediate location consisting of non-volatile memories is desired here (by checking for the final known alternating data pattern), it is not required at this time. However, vendors are highly encouraged to incorporate this verification whenever possible into their implementations.

e) For volatile memory and non-volatile EEPROM and Flash memories, the zeroization shall be executed by a single direct overwrite consisting of a pseudo random pattern, followed by a read-verify.

5.1.2.3 Cryptographic Operation (FCS_COP)

Cryptographic Operation (for data encryption/decryption) (FCS_COP.1(1))

FCS_COP.1.1(1) Refinement: **The cryptomodule shall perform** encryption and decryption using the FIPS-approved security function AES algorithm operating in *[assignment: one or more FIPS-approved modes]* and cryptographic key size of [*selection: one or more of 128 bits, 192 bits, 256 bits*]**.**

Cryptographic Operation (for cryptographic signature) (FCS_COP.1(2))

FCS_COP.1.1(2) Refinement: **The TSF shall perform** cryptographic signature services using the FIPS-approved security function *[selection:*

(5) Digital Signature Algorithm (DSA) with a key size (modulus) of [assignment: 2048 bits or greater],

(6) RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of [assignment: 2048 bits or greater], or

(7) Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of [selection: one or more of 256 bits, 384 bits, 521 bits], using only the NIST curve(s) [selection: one or more of P-256, P-384, P-521 as defined in FIPS PUB 186-3, “Digital Signature Standard”]]

that meets NIST Special Publication 800-57, “Recommendation for Key Management.”

Application Note: For elliptic curve-based schemes, the key size refers to the $\log_2$ of the order of the base point. As the preferred approach for key exchange, elliptic curves will be required after all the necessary standards and other supporting information are fully established.

Cryptographic Operation (for cryptographic hashing) (FCS_COP.1(3))

FCS_COP.1.1(3) Refinement: **The TSF shall perform** cryptographic hashing services using the FIPS-approved security function Secure Hash Algorithm and message digest size of *[selection: one or more of 256 bits, 384 bits, 512 bits]*.

Application Note: The message digest size should correspond to double the system symmetric encryption key strength.

Cryptographic Operation (for cryptographic key agreement) (FCS_COP.1(4))

Application Note: "Cryptographic key agreement" is a procedure where the resultant secret keying material is a function of information contributed by two participants, so that no party can predetermine the value of the secret keying material independently from the contributions of the other parties.

FCS_COP.1.1(4) Refinement: **The TSF shall perform** cryptographic key agreement services using the FIPS-approved security function as specified in NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" *[selection:*

(1) [assignment: Finite Field-based key agreement algorithm] and cryptographic key sizes (modulus) of [assignment: 2048 bits or greater], or

(2) [assignment: Elliptic Curve-based key agreement algorithm] and cryptographic key size of [assignment: one or more of 256 bits, 384 bits, 521 bits], using only the NIST curve(s) [selection: one or more of P-256, P-384, P-521 as defined in FIPS PUB 186-3, "Digital Signature Standard"]]

Application Note: For elliptic curve-based schemes, the key size refers to the $\log_2$ of the order of the base point. As the preferred approach for key exchange, elliptic curves will be required after all the necessary standards and other supporting information are fully established.

a) that meets NIST Special Publication 800-57, "Recommendation for Key Management."

Application Note: Some authentication mechanism on the keying material is recommended. In addition, repeated generation of the same shared secrets should be avoided.

Application Note: FIPS 140-2 Annex D specifies references for FIPS-approved Key Establishment Techniques, one of which is NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography."

Extended: Random Number Generation (FCS_COP_(EXT).1)

FCS_COP_(EXT).1.1 The TSF shall perform all random number generation (RNG) services in accordance with a FIPS-approved RNG [*assignment: one of the RNGS specified in FIPS 140-2 Annex C*] seeded by *[selection:*

(1) one or more independent hardware-based entropy sources, and/or

(2) one or more independent software-based entropy sources, and/or

(3) a combination of hardware-based and software-based entropy sources.]

Application Note: The ST author should specify how the RNG is seeded.

FCS_COP_(EXT).1.2 **The TSF shall defend against tampering of the random number generation (RNG)/ pseudorandom number generation (PRNG) sources.**

Application Note: The RNG/PRNG should be resistant to manipulation or analysis of its sources, or any attempts to predictably influence its states. Three examples of very different approaches the TSF might pursue to address this include: a) identifying the fact that physical security must be applied to the product, b) applying checksums over the sources, or c) designing and implementing the TSF RNG with a concept similar to a keyed hash (e.g., where periodically, the initial state of the hash is changed unpredictably and each change is protected as when provided on a tamper-protected token, or in a secure area of memory.

5.1.3 User Data Protection (FDP)

FDP_RIP.2 Full residual information protection

FDP_RIP.2.1 – **Refinement**: The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: allocation of the resource to, deallocation of the resource from] all objects **or the TSF's completion of a function.**

Application Note: This SFR ensures residual biometric data (e.g., biometric samples stored temporarily in the capture device) is not available after its use in the functional component. This requirement was refined, since the resources may not be deallocated or reallocated (e.g., memory may be allocated to a function and never released). The intent is that once the TSF is has completed the processing of data, that data is no longer accessible. For example, clearing a biometric sample from the capture device memory after its operation, or from the "Matching and Comparison" component(s) after a match/no match decision is made.

5.1.4 Identification and Authentication (FIA)

FIA_AFL.1-NIAP-0425(1) Authentication failure handling (Against a single non-administrative user identifier)

FIA_AFL.1.1-NIAP-0425(1) – **Refinement:** The TSF shall detect when [a Security Administrator configurable number] **of** unsuccessful **biometric** authentication attempts occur related to [a claimed user identifier, **[selection: [assignment: other authentication mechanisms identified by the ST Author], none]**].

FIA_AFL.1.2-NIAP-0425(1) - **Refinement:** When the defined number of **consecutive** unsuccessful authentication attempts has been met, the TSF shall [ignore any further authentication attempts related to that user until the Security Administrator defined time period for non-administrative users has elapsed, or an action is taken by the Security Administrator].

Application Note: The intent of these requirements is to allow the Security Administrator to set the number of unsuccessful authentication attempts that are associated with a claimed user identifier that is **not** associated with an administrative role. The Security Administrator also has the option of configuring the TOE so further authentication attempts associated with the claimed user identifier are ignored until the Security Administrator takes an action (e.g., re-enables the account) or to ignore further authentication attempts associated with the user identifier until a Security Administrator configured time period for non-administrative users has elapsed (e.g., the TOE will not authenticate a user associated with that non-administrative claimed user identifier for 5 minutes). The ST author should fill in the selection if the TOE provides additional authentication mechanisms (e.g., multiple biometric authentication mechanisms, password mechanism). If the TOE reaches the Security Administrator configured setting, then an alarm is generated as required by FAU_SAA.1.

FIA_AFL.1-NIAP-0425(2) Authentication failure handling (Consecutive failed attempts)

FIA_AFL.1.1-NIAP-0425(2) - The TSF shall detect when [a Security Administrator configurable number] **of** unsuccessful authentication attempts occur related to [consecutive failed biometric authentication attempts].

FIA_AFL.1.2-NIAP-0425(2) – **Refinement:** When the defined number of **consecutive** unsuccessful authentication attempts has been met, the TSF shall [ignore any further authentication attempts from the offending capture device until the Security Administrator defined time period for consecutive failed authentication attempts has elapsed, or an action is taken by the Security Administrator].

Application Note: The intent of this requirement is to provide the Security Administrator the capability to set the number of consecutive failed authentication attempts, regardless of the claimed user identifier. This configurable number is different than that specified in FIA_AFL.1. For example, the Security Administrator may decide to set the failed number of authentication attempts against a non-administrative claimed user identifier to be three, and may set the failed number of consecutive failed authentication attempts to six. The Security Administrator defined time period is also distinct from the non-administrative user defined period defined in FIA_AFL.1(1). For example, the Security Administrator may set the time period for non-administrative users to be 5 minutes, but might configure the consecutive failed authentication attempts time period to be one hour. As with the pervious iteration, if the TOE reaches the Security Administrator configured setting, then an alarm is generated as required by FAU_SAA.1.

FIA_AFL.1-NIAP-0425(3) Authentication failure handling (Administrator Users)

FIA_AFL.1.1-NIAP-0425(3) - The TSF shall detect when [a Security Administrator configurable number] of unsuccessful authentication attempts occur related to [the administrator's account].

FIA_AFL.1.2-NIAP-0425(3) – **Refinement**: When the defined number of **consecutive** unsuccessful authentication attempts has been met, the TSF shall [ignore any further authentication attempts related to that user until the Security Administrator defined time period for administrative users has elapsed, or an action is taken by the Security Administrator].

Application Note: This iteration of FIA_AFL.1 applies to claimed user identifiers associated with an administrative role. The TOE has the ability to associate claimed user identifiers with administrative accounts, otherwise it would not know that that claimed user identifier may have to use the non-biometric authentication mechanism. The Security Administrator configurable number is distinct from the configurable number specified in the previous two iterations, as is the Security Administrator time period. This configurable setting applies to the any authentication mechanism used to authenticate administrative users of the TOE (e.g., biometric authentication mechanism(s), non-biometric authentication mechanism (e.g., password). As with the previous iterations of FIA_AFL.1, if the TOE reaches the Security Administrator configured setting, then an alarm is generated as required by FAU_SAA.1. Since the administrators may be required to use more than the biometric authentication mechanism, this requirement applies to any authentication mechanism used by the administrators.

FIA_ATD.1 User attribute definition

FIA_ATD.1.1 – **Refinement**: The TSF shall maintain the following list of security attributes belonging to **administrative** users:

- [trusted user identifier,
- role(s), and

- **[selection: [assignment: any other security attributes defined by the ST Author], none.]]**

and restrict the ability to assign and modify these security attributes to the Security Administrator.

Application Note: The TOE only associates security attributes with administrative users. An administrator may have more than one role associated with their trusted user identifier, however they can only act in one role at a time. Untrusted users do not have any interaction with the TOE that requires the association of security attributes. Due to the TOE having the ability to authenticate untrusted users that have not been enrolled on TOE, it may not be possible for the TOE to associate security attributes with untrusted users. The IT environment is responsible for associating security attributes with the user identifier authenticated via the TOE.

FIA_ENROLL_(EXT).1 Enrollment

FIA_ENROLL_(EXT).1.1 The TSF shall enforce the following rules:

a) Creation of the biometrics package, which contains:

- Trusted user identifier,
- reference template(s),
- [selection: [assignment: list of additional information determined by the ST Author], no additional information],

is performed during enrollment only;

b) A reference template cannot be modified;[5]

c) Enrollment (e.g., initial, refreshing reference templates, adding additional reference templates[6]) is performed by the Enrollment Administrator;

d) The failure-to-enroll rate is less than or equal to [assignment: rate assigned by ST Author that does not exceed a maximum value of 5%];

e) The Enrollment administrator is provided a quality metric of the newly created reference template;

[5] The reference template cannot be modified once it has been created. For biometric technologies that continuously gather biometric characteristics to improve the quality of the reference template, a new template is created, rather than modifying an existing template.

[6] A biometric package may contain more than one reference template (e.g., a multifactor biometric device, to accommodate multiple vendors or technologies in a user's biometric package).

f) Upon successful enrollment, the biometrics package is digitally signed by the TOE, and then encrypted before transfer to storage;

g) [selection: [assignment: other rules determined by the ST Author], none].

Application Note: The biometrics package may have more than one reference template associated with a trusted user identifier. This may be the case if the TOE that uses multiple biometric characteristics when authenticating a user (e.g., both thumb prints).

The assignment in item (a) may be filled in with other information such as which finger the user has enrolled with, a distress template (e.g., if the user attempts to authenticate with a biometric characteristic known to indicate a distress situation – using the right thumb instead of the left) or other information the TOE may use. If the ST author adds additional attributes, they should consider adding or augmenting existing requirements that use those attributes (e.g., adding a rule in FIA_UAU.5 that handles a distress indicator).

Item (b) ensures the reference template cannot be modified once it has been created. The TOE ensures the reference template cannot be modified while it is in the TOE's scope of control, and the TOE determines if the reference template has been modified while in storage or transit through the use of a cryptographic signature. In the case of "refreshing" a reference template, the old reference template is replaced with a new one, and the TOE must digitally resign the biometric package containing the new template.

Item (d) requires that the Enrollment Administrator be provided a quality metric of the newly created reference template. In a biometric system, the level of security achieved is known to be dependent on the quality of the biometric reference templates. If a poor enrollment is allowed, then that user may be open to easy attack by an imposter. This PP does not explicitly contain a minimally acceptable quality metric. This is left to the ST author and is discussed in the administrator guidance. The administrative guidance informs the Enrollment Administrator what are acceptable quality metrics. This allows the Enrollment Administrator to make an informed decision regarding the quality of the reference template and whether they should attempt to re-enroll the user.

For item (e), the ST author could add a rule that allows the TOE to be configured such that it will perform a comparison of any new reference template against the existing templates if they desire. This would allow the Enrollment Administrator the opportunity to find out which pairs of individuals cannot easily be distinguished by the product by performing inter-template comparisons. Such information should be kept confidential because an attacker may discover this information and try to make use of it.

FIA_SOS.1 Verification of secrets

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [the following: For each attempt to use a non-biometric authentication mechanism, the probability that a random attempt to authenticate will succeed is less than one in 1 x 10^8].

Application Note: The ST specifies the method of authentication in FIA_UAU.5.1. When the non-biometric authentication is provided by a password mechanism, the ST shows that the restrictions upon passwords (length, alphabet, and other characteristics) result in a password space conforming to the specified metric. Administrators are able to select their authentication data (e.g., chose a password), but the TOE ensures that the chosen authentication data meets the identified metric.

FIA_SOS.2 TSF Generation of secrets

FIA_SOS.2.1 - The TSF shall provide a mechanism to generate secrets that meet [the following:

a) For each attempt to use the authentication mechanism, the False Acceptance Rate shall be in a Security Administrator settable range with a minimum value of: [assignment: rate assigned by ST Author] to a maximum value of: 1 in 100,000, and

b) False Rejection Rate shall be in a Security Administrator settable range with a minimum value of: [assignment: rate assigned by ST Author] to a maximum value of: 5 in 100.]

Application Note: In this TOE, the TSF generates the secret (i.e., the reference template) using an algorithm that is based on the biometric technology and uses a user's biometric characteristic. Since different biometric technologies provide varying degrees of False Acceptance Rates (FAR), this PP requires that at the maximum, the TOE will not have a FAR greater than 1 in 100,000. The ST author fills in the open assignment with a rate for a FAR their TOE can enforce. If the TOE cannot enforce a FAR less than 1 in 100,000 it is acceptable for the ST author to use the rate 1 in 100,000 in the assignment. Similarly, the False Rejection Rate (FRR) is specified as the maximum rate of false rejections the TOE will generate, and the ST author fills in the assignment with a rate that is less than or equal to the specified maximum rate of 5 in 100.

FIA_SOS.2.2 - The TSF shall be able to enforce the use of TSF generated secrets for [biometric authentication].

Application Note: The PP authors believe one aspect in ensuring that the TOE can enforce the rates specified in this requirement is the degree of quality of the reference templates. If the TOE allows a poor quality reference template to be accepted in the enrollment process, the belief is that these rates may be adversely affected.

FIA_UAU.2 User authentication before any action

FIA_UAU.2.1 – The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application Notes: This requirement really applies only to administrators, since they are the only users of the TOE that perform TSF mediated actions other than authentication. Non-administrative users perform no actions on the TOE other than requesting authentication, which is addressed by FIA_UAU_5.1.

FIA_UAU.5 Multiple authentication mechanisms

> FIA_UAU.5.1 **Refinement**: The TSF shall provide [a biometric authentication mechanism, [assignment: non-biometric authentication mechanism that meets the strength of secrets metric defined in FIA_SOS.1], **[selection: [assignment: any other authentication mechanisms defined by the ST Author], none.]**] to **perform** user authentication.

Application Note: The TOE provides at a minimum, one biometric authentication mechanism and another non-biometric authentication mechanism (e.g., password mechanism, personal identification number). The non-biometric authentication mechanism is to be used, at the option of the Security Administrator, to authenticate administrators of the TOE. This non-biometric authentication mechanism satisfies the FIA_SOS.1 requirement.

The ST author may fill in the selection with an assignment of additional authentication mechanisms or may choose none in the selection. If the ST author fills in the assignment, then they should ensure that the additional mechanisms satisfy the appropriate FIA_SOS requirements, or iterate the FIA_SOS requirements to specify the strength of secrets those mechanisms provide. The ST author should also ensure that the rules in FIA_UAU.5.2 are enforced by the additional mechanisms, or create new rules that correspond to the behavior of the additional mechanisms.

If the TOE provides multiple biometric mechanisms, or multifactor authentication (biometric and non-biometric (e.g., token, password) mechanisms) for non-administrative users then the ST author should either iterate this requirement to accommodate additional authentication mechanisms, or specify the additional mechanisms and the rules that apply to those mechanisms. The TOE provides at least one biometric mechanism that satisfies the rules stated in this requirement. Any additional biometric mechanism(s) satisfy the rules specified by the ST author, which could be those specified in this requirement.

> FIA_UAU.5.2 **Refinement**: The TSF shall authenticate any user’s claimed identity according to the [following:
>
> - ➢ For **non-administrative users**, the TSF shall authenticate a user and provide the IT environment with the trusted user identifier and a match/non-match decision according to the following rules:
>
>   b) At the option of the Security Administrator a liveness check for [selection: involuntary response(s), voluntary response(s), vital sign(s), realness] which consists of [assignment: a description of what the TOE does in performing the liveness check] is performed and passed when

the user supplied biometric characteristic is captured. If the liveness check fails, the TOE does not perform a comparison of the templates;

c) verify the integrity of the biometrics package(s), confirming that the TOE has cryptographically signed the biometrics package, or the biometrics package has been cryptographically signed by a trusted authority;

d) in order to provide a match decision the comparison score is within the range specified by the maximum threshold and minimum threshold, otherwise a non-match decision is generated;

e) at the option of the Security Administrator, the TOE will not successfully authenticate the same claimed user identifier consecutively in a time duration specified by the Security Administrator;

f) [selection: [assignment: other rules determined by the ST Author], none].

Application Note: The ST author fills in the first selection based on what the TOE provides to the environment. If the TOE is used as an entry device on a door, the match/no match decision may be an electrical signal that opens the door if the TOE determines a match. If the TOE is providing authentication services to an IT environment, the expectation is the TOE will provide the IT environment with the user identifier that was supplied by the user, and the match/no match decision.

The selection in (a) is determined by the type of liveness check the TOE performs. This may consist of multiple instances of the identified types or some combination. The assignment in (a) provides a description of the technique the TOE uses in performing the liveness test. This is not intended to have the developer’s proprietary algorithm described, rather it is intended to inform the end-user of what the TOE does with respect to liveness checking.

For item (b), the intent is that the TOE has the capability to maintain a list of trusted entities (e.g., Certificate Authorities) so the TOE can validate the integrity of a biometrics package that was not created by the TOE (e.g., the user was enrolled on another TOE) by using a trusted entity’s public key to verify that entity had cryptographically signed the biometrics package.

For item (c) the TOE has a threshold range. The purpose of having a range is to provide the capability for a site to determine that if a match score is too high (e.g., a perfect match with the reference template) the TOE will provide a no match decision.

For item (d), the Security Administrator has the ability to configure the TOE to prevent the same user from successfully authenticating consecutively in a Security Administrator defined period of time. For example, the Security Administrator could configure the TOE so that once User X has successfully authenticated, User X cannot be the **next** user to be authenticated until 10 minutes have passed. This functionality is intended to ensure a user cannot attempt to “use” a residual left from a biometric characteristic from another user.

- ➢ For **administrative users**, the Security Administrator can choose that these users require authentication only by the biometric authentication mechanism(s), only by the non-biometric authentication mechanism, or both types of authentication mechanisms.

  - When the TOE is configured to require administrators to use the biometric authentication mechanism, the TSF shall authenticate the administrative user and determine a match/non-match decision, according to the following rules:

    a) At the option of the Security Administrator a liveness check for [selection: involuntary response(s), voluntary response(s), vital sign(s), realness] which consists of [assignment: a description of what the TOE does in performing the liveness check] is performed and passed when the user supplied biometric characteristic is captured. If the liveness check fails, the TOE does not perform a comparison of the templates;

    b) verify the integrity of the biometrics package(s), confirming that the TOE has cryptographically signed the biometrics package, or the biometrics package has been cryptographically signed by a trusted authority;

    c) in order to provide a match decision the comparison score is within the range specified by the maximum threshold and minimum threshold, otherwise a non-match decision is generated;

    d) at the option of the Security Administrator, the TOE will not successfully authenticate the same claimed user identifier consecutively in a time duration specified by the Security Administrator;

    e) [selection: [assignment: other rules determined by the ST Author], none].

  - When the TOE is configured to require administrators to use the non-biometric authentication mechanism, the TSF shall authenticate the administrative user according to the following rules:

    a) The authentication mechanism must provide a delay between failed authentication attempts, such that there can be no more than a Security Administrator configurable number of attempts per minute;

b) Any feedback given during an attempt to use the authentication mechanism will not increase the probability of guessing above the metrics specified in FIA_SOS.1;

- When the TOE is configured to require administrators to use a biometric and non-biometric mechanism, the TSF shall authenticate the administrative user according to the following rules:

  a) The rules for each mechanism specified for the administrator above hold true;

  b) The administrator must be successfully authenticated by both mechanisms;

  c) The authentication mechanisms provide no feedback unless both mechanisms are successful, other than to inform the user that the authentication process failed.

].

Application Note: The intent of item c) is to ensure the TOE does not indicate to the user which authentication mechanism failed (e.g., your password succeeded, but the biometric authentication failed).

FIA_UAU.7 Protected authentication feedback

FIA_UAU.7.1 – **Refinement**: The TSF shall provide only [instructional information] to **aid** the user **in supplying their biometric characteristic to the TOE**.

Application Note: This requirement means that the biometric system must not inform the user of any “score” against the threshold range that might help the attacker to fool the device in subsequent authentication attempts. Instructional information includes positioning information, volume, which finger to authenticate with, etc.

FIA_UID.2 User identification before any action

FIA_UID.2.1 – The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

FIA_USB.1-NIAP-0415 User-subject binding

FIA_USB.1.1-NIAP-0415: **Refinement:** The TSF shall associate the **following** user security attributes with subjects acting on behalf of that user: **[trusted user identifier, role, [selection: assignment: list of other security attributes determined by the ST Author to be bound, none]]**.

Application Note: As with FIA_UAU.2, the only users that attributes are associated with subjects are those of administrative users, since those are the only users that will have subjects with any “user” attributes.

5.1.5 Security Management Requirements (FMT)

FMT_MOF.1(1) Management of security functions behavior (audit selection)

FMT_MOF.1.1(1) - The TSF shall restrict the ability to *enable, disable, determine and modify the behavior of* the functions:

- [Security Audit (FAU_SEL)]

to [the Audit Administrator].

Application Note: For the Audit function, enable and disable refer to the ability to enable or disable the audit mechanism as a whole. “Determine the behavior” means the ability to determine specifically what on the system is being audited, while “modify the behavior” means the ability to set or unset specific aspects of the audit mechanism, such as what user behavior is audited, etc.

FMT_MOF.1(2) Management of security functions behavior (audit review)

FMT_MOF.1.1(2) - The TSF shall restrict the ability to *enable, and modify the behavior of* the functions:

- [Security Audit (FAU_SAR)]

to [the Audit Administrator and Security Administrator].

Application Note: For the Audit review function, enable refers to the ability to use the audit review tool (e.g., start the program). “Modify the behavior” means the ability to set or unset the criteria used to select/search audit records for review.

FMT_MOF.1(3) Management of security functions behavior (alarms)

FMT_MOF.1.1(3) - The TSF shall restrict the ability to *enable, disable, determine and modify the behavior of* the functions:

- [Security Audit Analysis (FAU_SAA); and
- Security Alarms (FAU_ARP)],

to [the Security Administrator].

Application Note: This requirement ensures only the Security Administrator can enable or disable (turn on or turn off) the alarm notification function. For FAU_ARP.1, behavior modification includes adjusting the defined time period that elapses before the TOE will resume performing authentication. For FAU_SAA, the intent of “modify the behavior” applies to the ability of the Security Administrator to selectively disable or enable specific events that may indicate a potential security violation.

FMT_MOF.1(4) - Management of security functions behavior (TSF non-Cryptographic Self-test)

FMT_MOF.1.1(4) - The TSF shall restrict the ability to *modify the behavior of* the functions:

- [TSF Self-Test (FPT_TST_(EXT).1)]

to [the Security Administrator].

Application Note: “Modify the behavior” refers to specifying the interval at which the test periodically run, or perhaps selecting a subset of the tests to run.

FMT_MOF.1(5) - Management of security functions behavior (Cryptographic Self-test)

FMT_MOF.1.1(5) - The TSF shall restrict the ability to *enable, disable* the functions:

- [TSF Self-Test (FPT_TST1(1) and FPT_TST1(2))]

to [the Security Administrator].

Application Note: The enabling or disabling of the cryptographic self-tests immediately after key generation.

FMT_MOF.1(6) Management of security functions behavior (Maintenance Mode)

FMT_MOF.1.1(6) - The TSF shall restrict the ability to *enable* the functions [to restore the TOE to a secure state from maintenance mode (FPT_RCV.2)] to [the Security Administrator].

Application Note: The intent of this requirement is if the TOE enters a state that it cannot automatically recover from and ensure that it will be able to enforce its security policies, that only a user acting in the role of the Security Administrator can restore the TOE to an operational state. One way this could be accomplished by requiring some form of a required password before startup from the maintenance mode state.

FMT_MOF.1(7) Management of security functions behavior (Enrollment)

FMT_MOF.1.1(7) – **Refinement:** The TSF shall restrict the ability to **perform**, *determine* and *modify the behavior of* the function [enrollment (FIA_ENROLL_(EXT).1)] to [the Enrollment Administrator].

Application Notes: The Enrollment Administrator is the only user that is allowed to perform the enrollment function. “Determine the behavior” refers to the ability of the Enrollment Administrator to view any settings that the TOE may offer that affect the quality of the created reference template, as well as receiving the quality metric of the reference template when it is created. “Modify the behavior” refers to the Enrollment Administrator having the capability to set parameters that may affect the quality of the reference template when it is created, if the TOE offers such capability.

FMT_MOF.1(8) Management of security functions behavior (non-biometric Authentication Mechanism)

> FMT_MOF.1.1(8) - The TSF shall restrict the ability to *enable and disable* the functions: [non-biometric authentication mechanism for required use on individual administrative roles] to [the Security Administrator].

Application Note: The Security Administrator has the ability to require the use of (enable or disable) the non-biometric authentication mechanism for individual administrative accounts.

FMT_MOF.1(9) Management of security functions behavior (Biometric Authentication Mechanism)

FMT_MOF.1.1(9) – **Refinement**: The TSF shall restrict the ability to *determine and modify the behavior of* the functions: [

- biometric authentication mechanism];

and to enable/disable the function:

- **biometric authentication mechanism for required use on individual administrative roles**

to [the Security Administrator].

Application Note: The Security Administrator has the ability to modify the behavior of biometric authentication mechanism by turning the liveness check on or off. Determine in this requirement applies to the Security Administrator being able to query the liveness check setting. The CC includes both the management (modifying the behavior) of a security function, and management of TSF data. It is sometimes confusing where to place certain aspects pertaining to the management of a TSF function, since managing TSF data can have an affect on the behavior of a TSF function. FMT_MTD.1(3) identifies TSF data that will have an impact on the behavior of this function and places restrictions on what administrative role can mange that TSF data.

The intent of this iteration and the previous iteration is to allow the Security Administrator the flexibility in determining which administrative accounts, at the granularity of an individual, require which authentication mechanisms are necessary for access. The intent is to allow an administrator to require specific individual administrative roles to authenticate to the TOE in a Security Administrator defined manner. The intent is to address a concern that the administrative role may not be able to administer the TOE if the biometrics authentication mechanism is rendered unavailable. This requirement, in conjunction with the previous iteration, could allow a Security Administrator to set up a **special** Security Administrator account that would require authentication only be the non-biometric account, while requiring a biometric authentication mechanism for all other administrator authentication attempts.

FMT_MTD.1(1) Management of TSF data (cryptographic TSF data)

> FMT_MTD.1.1(1) - The TSF shall restrict the ability to *modify* the [cryptographic security data] to [the Security Administrator].

Application Note: The intent of this requirement is to restrict the ability to configure the TOE's cryptographic policy to the Security Administrator. Configuring the cryptographic policy is related to things such as: setting modes of operation, key lifetimes, selecting a specific algorithm, manually entering keys, and key length.

FMT_MTD.1(2) Management of TSF data (time TSF data)

> FMT_MTD.1.1(2) – The TSF shall restrict the ability to *modify* the [time and date used to form the time stamps in FPT_STM.1] to [the Security Administrator].

FMT_MTD.1(3) Management of TSF data (Authentication Mechanism Data)

> FMT_MTD.1.1(3) – **Refinement**: The TSF shall restrict the ability to *query, modify* the:

- [value of the minimum threshold; (FIA_UAU.5.2)
- value of the maximum threshold; (FIA_UAU.5.2)
- defined time period for blocking of further authentication attempts:
- time period for non-administrative users (FIA_AFL.1(1));
- time period for consecutive failed authentication attempts (FIA_AFL.1(2));
- time period for administrative users (FIA_AFL.1(3));
- defined time period has elapsed upon an alarm condition (FAU_ARP.1);
- Security Administrator configurable number of attempts per minute (FAU_UAU.5.2);

- time duration restricting the authentication of the same claimed user identifier consecutively;

- list of IT entities that the TOE will accept as cryptographic signing authorities;

- **[selection: [assignment: other data determined by the ST Author], none].**]

to [the Security Administrator].

FMT_MTD_(EXT).1 Management of TSF data (Capture device unique identifier)

FMT_MTD_(EXT).1.1 – The TSF [selection: provides unique capture device identifier(s), restricts the ability to query and modify the capture device identifier to the Security Administrator].

FMT_REV.1 Revocation

FMT_REV.1.1 - **Refinement:** The TSF shall restrict the ability to revoke security attributes associated with the **administrative** *users*, **[selection: [assignment: other additional resources specified by the ST Author], none]** within the TSC to [the Security Administrator].

FMT_REV.1.2 - The TSF shall enforce the rules:

- [revocation of a user’s role is immediate; and

- [selection: [assignment: other rules as determined by the ST Author], none]].

Application Note: The security attributes associated with users are defined in FIA_ATD.1. If the ST author has added additional attributes in FIA_ATD.1 they should use the selection above to identify the rules for revoking those attributes.

FMT_SMR.2 Restrictions on security roles

FMT_SMR.2.1 **– Refinement**: The TSF shall maintain the roles:

- [Security Administrator;

- Audit Administrator;

- Enrollment Administrator, and

- **[selection: [assignment: any other roles determined by the ST Author], none]**].

FMT_SMR.2.2 - The TSF shall be able to associate users with roles.

FMT_SMR.2.3 - The TSF shall ensure that the conditions:

- [All roles shall be able to administer the TOE locally;
- all roles shall be able to administer the TOE remotely;
- all roles are distinct; that is, there shall be no overlap of operations performed by each role, with the following exceptions:
- the Audit Administrator and Security Administrator can review the audit trail; and
- any administrator can invoke the self-tests].

are satisfied.

Application Note: The administering of the TOE is limited to the capabilities associated with each administrative role. When the term administrator is used in this PP it refers to a person acting in any of the roles specified in FMT_SMR.2.1. The FIPS 140 validated cryptographic module for this TOE (level 3 for Roles) requires that unique trusted user identifiers be assigned to administer the cryptographic module. Only users associated with the Security Administrator role are allowed to administer the cryptographic module.

5.1.6 Protection of TSF (FPT)

FPT_ITT.1(1) Basic internal TSF data transfer protection (from disclosure)

FPT_ITT.1.1(1) **Refinement**: The TSF shall **use encryption to** protect TSF data from *disclosure* when it is transmitted between separate parts of the TOE.

FPT_ITT.1(2) Basic internal TSF data transfer protection (from undetected modification)

FPT_ITT.1.1(2) **Refinement**: The TSF shall **use a cryptographic digital signature to detect modification of** TSF data when it is transmitted between separate parts of the TOE.

FPT_ITC_(EXT). 1 TSF confidentiality

FPT_ITC_(EXT).1.1 - The TSF shall use encryption to protect the confidentiality of the biometrics package.

Application Note: This **extended** requirement is necessary because the CC does not contain a requirement that specifies the desired functionality. The intent is that for this requirement and for FPT_ITT.1(1), the TOE encrypts the biometrics package using the cryptography specified in FCS_COP_(EXT).2. The TOE can then send the biometrics package to a storage device that is not trusted over an unencrypted channel and ensure that the confidentiality of the data in the biometrics package is maintained. This requirement is different from the rest of the FPT_ITC family in that it does not require the storage device to be cryptography aware, it just stores the data it is provided.

The TOE may need to accept biometric packages that have been encrypted by another IT entity (i.e., user was enrolled on a different TOE). The key management requirements in FCS_CKM._(EXT).2 specify the acceptable means for accomplishing key establishment to allow the TOE to decrypt the biometrics package.

FPT_ITI_(EXT).1 TSF detection of modification

FPT_ITI_(EXT).1.1 The TSF shall use a cryptographic digital signature to detect modification of the biometrics package.

FPT_ITI_(EXT).1.2 The TSF shall maintain a list of IT entities that it will accept as valid cryptographic signing authorities.

FPT_ITI_(EXT).1.3 The TSF shall reject a biometrics package if modification is detected.

Application Note: This **extended** requirement is necessary because the CC does not contain a requirement that specifies the desired functionality. The intent is that for this requirement and for FPT_ITT.1(2), the TOE cryptographically signs the biometrics package using the cryptographic digital signature algorithm specified in FCS_COP_(EXT).3. The TOE maintains a list of trusted authorities and will accept biometrics packages that have been signed by those authorities. The key management requirements are presented in FCS_CKM._(EXT).2 requirements and they specify the acceptable means for accomplishing key management.. There are no requirements for specifying the maintenance of the list of trusted authorities. This could be done locally by the Security Administrator, or remotely in a networked TOE. The TOE can send/receive a biometrics package to/from a storage device and not rely on the protection of the transmission medium, or the storage device to protect the integrity of the biometrics package. The TOE is able to verify the integrity of the biometrics package due to the fact that the biometrics package is cryptographically signed.

FPT_PHP_(EXT).1 Detection of physical attack

FPT_PHP_(EXT).1.1 The TSF shall detect physical tampering involving the following scenarios that might compromise the TSF: exposure of the internal components of biometrics TOE, [selection: [assignment: other scenarios determined by the ST author], none].

Application Note: This **extended** requirement is necessary because the existing CC requirements do not allow for identifying the specific scenarios the TOE must detect.

This requirement includes all components of the TOE (e.g., capture device, enrollment device). If accomplished through the use of tamper-proof seals, the TOE developer provides the seals, or instructs the administrator how to obtain the appropriate seals. The administrator's guide instructs the administrator how and where to place the seals in order to allow detection of tampering. The intent of detect is that an audit record and alarm are generated.

FPT_PHP.3 Resistance to physical attack

FPT_PHP.3.1 **Refinement:** The TSF shall **react** [to the exposure of internal components] **of** the [biometrics TOE] by **zeroizing any cryptographic security parameters and** responding automatically such that the TSP is not violated.

FPT_RCV.2-NIAP-0406 Recovery from Failure

FPT_RCV.2.1-NIAP-0406 For [power failures], the TSF shall ensure the return of the TOE to a secure state using automated procedures.

FPT_RCV.2.2-NIAP-0406 When automated recovery from a failure or service discontinuity is not possible, the TSF shall enter a maintenance mode where the ability to return the TOE to a secure state is provided.

Application Note: The administrative guidance provides the Security Administrator with guidance/procedures that instruct them how to bring the TOE back into a secure state. If the TOE is unable to return to a secure state using automated procedures after a power failure the TOE enters a maintenance mode.

FPT_STM.1 Reliable time stamps

FPT_STM.1.1 - The TSF shall be able to provide reliable time stamps for its own use.

Explicit: TSF Testing (FPT_TST_(EXT).1)

FPT_TST_(EXT).1.1 The TSF shall run a suite of self tests <u>during the initial start-up and also either periodically during normal operation,</u> or at the request of an authorized administrator to demonstrate the correct operation of the TSF.

FPT_TST_(EXT).1.2 The TSF shall provide authorized administrators with the capability to verify the integrity of stored TSF executable code through the use of the TSF-provided cryptographic services.

Application Note: Refer to FCS_COP.1.1(2) and FCS_COP.1.1(3) for TSF-provided cryptographic services .

TSF Testing (for cryptography) (FPT_TST.1(1))

FPT_TST.1.1(1) Refinement: **The TSF shall run a suite of self tests** in accordance with FIPS PUB 140-2 and Appendix C of this profile **during initial start-up** (on power on)**, at the request of the** cryptographic administrator (on demand)**,** under various conditions defined in section 4.9.1 of FIPS 140-2, **and periodically** (at least once a day) **to demonstrate the correct operation of the** following cryptographic functions:[i]

a) **key error detection;**

b) **cryptographic algorithms;**

c) **RNG/PRNG**

Application Note: These tests apply regardless of whether the cryptographic functionality is implemented in hardware, software, or firmware.

FPT_TST.1.2(1) Refinement: **The TSF shall provide authorized** cryptographic administrators **with the capability to verify the integrity of** TSF data related to the cryptography by using TSF-provided cryptographic functions**.**[ii]

Application Note: Refer to FCS_COP.1.1(2) and FCS_COP.1.1(3) for TSF-provided cryptographic services

.FPT_TST.1.3(1) Refinement: **The TSF shall provide authorized** cryptographic administrators **with the capability to verify the integrity of stored TSF executable code** related to the cryptography by using TSF-provided cryptographic functions**.**[iii]

Application Note: Refer to FCS_COP.1.1(2) and FCS_COP.1.1(3) for TSF-provided cryptographic services .

TSF Testing (for key generation components) (FPT_TST.1(2))

FPT_TST.1.1(2) Refinement**: The TSF shall** perform **self tests** immediately after generation of a key **to demonstrate the correct operation** of each key generation component**.** If any of these tests fails, that generated key shall not be used, the cryptographic module shall react as required by FIPS PUB 140-2 for failing a self-test, and this event will be audited**.**[iv]

Application Note: Key generation components are those critical elements that compose the entire key generation process (e.g., any algorithms, any RNG/PRNGs, any key generation seeding processes, etc.).

Application Note: These self-tests on the key generation components can be executed here as a subset of the full suite of self-tests run on the cryptography in FPT_TST.1(1) as long as all elements of the key generation process are tested.

FPT_TST.1.2(2) Refinement: **The TSF shall provide authorized** cryptographic administrators **with the capability to verify the integrity of TSF data** related to the key generation by using TSF-provided cryptographic functions**.**[v]

Application Note: Refer to FCS_COP.1.1(2) and FCS_COP.1.1(3) for TSF-provided cryptographic services

.FPT_TST.1.3(2) Refinement: **The TSF shall provide authorized** cryptographic administrators **with the capability to verify the integrity of stored TSF executable code** related to the key generation by using TSF-provided cryptographic functions**.vi**

Application Note: Refer to FCS_COP.1.1(2) and FCS_COP.1.1(3) for TSF-provided cryptographic services

5.1.7 TOE Access (FTA)

FTA_SSL.3 TSF-initiated termination

FTA_SSL.3.1 - Refinement: The TSF shall terminate **an administrative** session after a [Security Administrator-configurable time interval of session inactivity].

FTA_TAB.1 Default TOE access banners

FTA_TAB.1.1 - Refinement: Before establishing an **administrative** session, the TSF shall display an advisory **notice and consent** warning message regarding unauthorized use of the TOE.

Application Note: The access banner applies whenever the TOE will provide a prompt for identification and authentication of an administrator. The intent of this requirement is to advise administrators of warnings regarding the unauthorized use of the TOE. For untrusted users, the environment (IT or non-IT) would be responsible for displaying the appropriate banner.

FTA_TSE.1 TOE session establishment

FTA_TSE.1.1 - Refinement: The TSF shall be able to deny establishment **of an administrative session** based on [the combination of: trusted user identifier, role, location, time, and day].

5.2 TOE Security Assurance Requirements

The TOE assurance requirements for this PP are the Medium Robustness Assurance Package and do not map to a CC EAL. The assurance requirements are summarized in the Table 5.2 below.

Table 5.2 – Assurance Requirements

Assurance Class	ASSURANCE COMPONENTS	ASSURANCE COMPONENTS DESCRIPTION
DEVELOPMENT	ADV_ARC.1	Security Architectural Description
	ADV_FSP.5	**Complete semi-formal functional specification with additional error information**

Assurance Class	ASSURANCE COMPONENTS	ASSURANCE COMPONENTS DESCRIPTION
	ADV_IMP.1	Implementation of the TSF
	ADV_INT.3	**Minimally complex internals**
	ADV_TDS.4	**Semiformal modular design**
GUIDANCE DOCUMENTS	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative User guidance
LIFE CYCLE SUPPORT	ALC_CMC.4	Product support, acceptance procedures and automation
	ALC_CMS.4	Problem tracking CM coverage
	ALC_DEL.1	Delivery procedures
	ALC_DVS.1	Identification of security measures
	ALC_FLR.2	**Flaw Reporting Procedures**
	ALC_LCD.1	Developer defined life-cycle model
	ALC_TAT.1	Well-defined development tools
TESTS	ATE_COV.2	Analysis of coverage
	ATE_DPT.3	**Testing: modular design**
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
VULNERABILITY ASSESSMENT	**AVA_CCA_(EXT).1**	**Systematic cryptographic module covert channel analysis (required when Cryptography is invoked)**
	AVA_VAN.4	**Methodical vulnerability analysis**

5.2.1 Class ADV: Development

5.2.1.1 ADV_ARC.1 Security architecture description

Dependencies: ADV_FSP.1 Basic functional specification

ADV_TDS.1 Basic design

Developer action elements:

ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3D The developer shall provide a security architecture description of the TSF.

Content and presentation elements:

ADV_ARC.1.1C The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3C The security architecture description shall describe how the TSF initialization process is secure.

ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

Evaluator action elements:

ADV_ARC.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

5.2.1.2 ADV_FSP.5 Complete semi-formal functional specification with additional error information

Dependencies: ADV_TDS.1 Basic design,
ADV_IMP.1 Implementation representation of the TSF Developer action elements:

Developer action elements:

ADV_FSP.5.1D The developer shall provide a functional specification.

ADV_FSP.5.2D The developer shall provide a tracing from the functional specification to the SFRs. Content and presentation elements:

ADV_FSP.5.1C The functional specification shall completely represent the TSF.

ADV_FSP.5.2C The functional specification shall describe the TSFI using a semi-formal style.

ADV_FSP.5.3C The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.5.4C The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.5.5C The functional specification shall describe all actions associated with each TSFI.

ADV_FSP.5.6C The functional specification shall describe all direct error messages that may result from an invocation of each TSFI.

ADV_FSP.5.7C The functional specification shall describe all error messages that do not result from an invocation of a TSFI.

ADV_FSP.5.8C The functional specification shall provide a rationale for each error message contained in the TSF implementation yet does not result from an invocation of a TSFI.

ADV_FSP.5.9C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification. Evaluator action elements:

Evaluator action elements:

ADV_FSP.5.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.5.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.2.1.3 ADV_IMP.1Implementation representation of the TSF

Dependencies: ADV_TDS.3 Basic modular design

ALC_TAT.1 Well-defined development tools

Developer action elements:

ADV_IMP.1.1D The developer shall make available the implementation representation for the entire TSF.

ADV_IMP.1.2D The developer shall provide a mapping between the TOE design description and the sample of the implementation representation.

Content and presentation elements:

ADV_IMP.1.1C The implementation representation shall define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.1.2C The implementation representation shall be in the form used by the development personnel.

ADV_IMP.1.3C The mapping between the TOE design description and the sample of the implementation representation shall demonstrate their correspondence.

Evaluator action elements:

ADV_IMP.1.1E The evaluator *shall confirm* that, for the selected sample of the implementation representation, the information provided meets all requirements for content and presentation of evidence.

5.2.1.4 ADV_INT.3 Minimally complex internals

Dependencies: ADV_IMP.1 Implementation representation of the TSF

ADV_TDS.3 Basic modular design

ALC_TAT.1 Well-defined development tools

Developer action elements:

ADV_INT.3.1D The developer shall design and implement the entire TSF such that it has well-structured internals.

ADV_INT.3.2D The developer shall provide an internals description and justification.

Content and presentation elements:

ADV_INT.3.1C The justification shall describe the characteristics used to judge the meaning of “well-structured” and “complex”.

ADV_INT.3.2C The TSF internals description shall demonstrate that the entire TSF is well structured.

Evaluator action elements:

ADV_INT.3.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ADV_INT.3.2E The evaluator *shall perform* an internals analysis on the entire TSF.

5.2.1.5 ADV_TDS.4 Semiformal modular design

Dependencies: ADV_FSP.5 Complete semi-formal functional specification with additional error information Developer action elements:

Developer action elements:

ADV_TDS.4.1D The developer shall provide the design of the TOE.

ADV_TDS.4.2D The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design. Content and presentation elements:

Content and presentation elements:

ADV_TDS.4.1C The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.4.2C The design shall describe the TSF in terms of modules, designating each module as SFR-enforcing, SFR-supporting, or SFR-non-interfering.

ADV_TDS.4.3C The design shall identify all subsystems of the TSF.

ADV_TDS.4.4C The design shall provide a semiformal description of each subsystem of the TSF, supported by informal, explanatory text where appropriate.

ADV_TDS.4.5C The design shall provide a description of the interactions among all subsystems of the TSF.

ADV_TDS.4.6C The design shall provide a mapping from the subsystems of the TSF to the modules of the TSF.

ADV_TDS.4.7C The design shall describe each SFR-enforcing and SFR-supporting module in terms of its purpose and interaction with other modules.

ADV_TDS.4.8C The design shall describe each SFR-enforcing and SFR-supporting module in terms of its SFR-related interfaces, return values from those interfaces, interaction with and called interfaces to other modules.

ADV_TDS.4.9C The design shall describe each SFR-non-interfering module in terms of its purpose and interaction with other modules.

ADV_TDS.4.10C The mapping shall demonstrate that all behaviour described in the TOE design is mapped to the TSFIs that invoke it. Evaluator action elements:

Evaluator action elements:

ADV_TDS.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_TDS.4.2E The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

5.2.2 Class AGD: Guidance documents

5.2.2.1 AGD_OPE.1 Operational user guidance

Dependencies: ADV_FSP.1 Basic functional specification

Developer action elements:

AGD_OPE.1.1D The developer shall provide operational user guidance.

Content and presentation elements:

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

Evaluator action elements:

AGD_OPE.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

5.2.2.2 AGD_PRE.1 Preparative procedures

Dependencies: No dependencies.

Developer action elements:

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

Content and presentation elements:

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

Evaluator action elements:

AGD_PRE.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2E The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.2.3 Class ALC: Life-cycle support

5.2.3.1 ALC_CMC.4 Production support, acceptance procedures and automation

Dependencies: ALC_CMS.1 TOE CM coverage

ALC_DVS.1 Identification of security measures

ALC_LCD.1 Developer defined life-cycle model

Developer action elements:

ALC_CMC.4.1D The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.4.2D The developer shall provide the CM documentation.

ALC_CMC.4.3D The developer shall use a CM system.

Content and presentation elements:

ALC_CMC.4.1C The TOE shall be labeled with its unique reference.

ALC_CMC.4.2C The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.4.3C The CM system shall uniquely identify all configuration items.

ALC_CMC.4.4C The CM system shall provide automated measures such that only authorized changes are made to the configuration items.

ALC_CMC.4.5C The CM system shall support the production of the TOE by automated means.

ALC_CMC.4.6C The CM documentation shall include a CM plan.

ALC_CMC.4.7C The CM plan shall describe how the CM system is used for the development of the TOE.

ALC_CMC.4.8C The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

ALC_CMC.4.9C The evidence shall demonstrate that all configuration items are being maintained under the CM system.

ALC_CMC.4.10C The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.

Evaluator action elements:

ALC_CMC.4.1E The evaluator ***shall confirm*** that the information provided meets all requirements for content and presentation of evidence.

5.2.3.2 ALC_CMS.4 Problem tracking CM coverage

Dependencies: No dependencies.

Developer action elements:

ALC_CMS.4.1D The developer shall provide a configuration list for the TOE.

Content and presentation elements:

ALC_CMS.4.1C The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation representation; and security flaw reports and resolution status.

ALC_CMS.4.2C The configuration list shall uniquely identify the configuration items.

ALC_CMS.4.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

Evaluator action elements:

ALC_CMS.4.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

5.2.3.3 ALC_DEL.1 Delivery procedures

Dependencies: No dependencies.

Developer action elements:

ALC_DEL.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the consumer.

ALC_DEL.1.2D The developer shall use the delivery procedures.

Content and presentation elements:

ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

Evaluator action elements:

ALC_DEL.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

5.2.3.4 ALC_DVS.1Identification of security measures

Dependencies: No dependencies.

Developer action elements:

ALC_DVS.1.1D The developer shall produce development security documentation.

Content and presentation elements:

ALC_DVS.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

Evaluator action elements:

ALC_DVS.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.1.2E The evaluator *shall confirm* that the security measures are being applied.

5.2.3.5 ALC_FLR.2Flaw reporting procedures

Dependencies: No dependencies.

Developer action elements:

ALC_FLR.2.1D The developer shall document flaw remediation procedures addressed to TOE developers.

ALC_FLR.2.2D The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.2.3D The developer shall provide flaw remediation guidance addressed to TOE users.

5.2.3.6 ALC_LCD.1 Developer defined life-cycle model

Dependencies: No dependencies.

Developer action elements:

ALC_LCD.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2D The developer shall provide life-cycle definition documentation.

Content and presentation elements:

ALC_LCD.1.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

Evaluator action elements:

ALC_LCD.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

5.2.3.7 ALC_TAT.1 Well-defined development tools

Dependencies: ADV_IMP.1 Implementation representation of the TSF

Developer action elements:

ALC_TAT.1.1D The developer shall identify each development tool being used for the TOE.

ALC_TAT.1.2D The developer shall document the selected implementation-dependent options of each development tool.

Content and presentation elements:

ALC_TAT.1.1C Each development tool used for implementation shall be well-defined.

ALC_TAT.1.2C The documentation of each development tool shall unambiguously define the meaning of all statements as well as all conventions and directives used in the implementation.

ALC_TAT.1.3C The documentation of each development tool shall unambiguously define the meaning of all implementation-dependent options.

Evaluator action elements:

ALC_TAT.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

5.2.4 Class ATE: Tests

5.2.4.1 ATE_COV.2 Analysis of coverage

Dependencies: ADV_FSP.2 Security-enforcing functional specification

ATE_FUN.1 Functional testing

Developer action elements:

ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

Content and presentation elements:

ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.

Evaluator action elements:

ATE_COV.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

5.2.4.2 ATE_DPT.3 Testing: modular design

Dependencies: ADV_ARC.1 Security architecture description
ADV_TDS.4 Semiformal modular design
ATE_FUN.1 Functional testing

Developer action elements:

ATE_DPT.3.1D The developer shall provide the analysis of the depth of testing.

Content and presentation elements:

ATE_DPT.3.1C The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems and modules in the TOE design.

ATE_DPT.3.2C The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.

ATE_DPT.3.3C The analysis of the depth of testing shall demonstrate that all TSF modules in the TOE design have been tested.

Evaluator action elements:

ATE_DPT.3.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

5.2.4.3 ATE_FUN.1Functional testing

Dependencies: ATE_COV.1 Evidence of coverage

Developer action elements:

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation elements:

ATE_FUN.1.1C The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.

Evaluator action elements:

ATE_FUN.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

5.2.4.4 ATE_IND.2 Independent testing - sample

Dependencies: ADV_FSP.2 Security-enforcing functional specification

AGD_OPE.1 Operational user guidance

AGD_PRE.1 Preparative procedures

ATE_COV.1 Evidence of coverage

ATE_FUN.1 Functional testing

Developer action elements:

ATE_IND.2.1D The developer shall provide the TOE for testing.

Content and presentation elements:

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Evaluator action elements:

ATE_IND.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator *shall execute* a sample of tests in the test documentation to verify the developer test results.

ATE_IND.2.3E The evaluator *shall test* a subset of the TSF to confirm that the TSF operates as specified.

5.2.5 Class AVA: Vulnerability assessment

5.2.5.1 AVA_CCA_(EXT).1 Systematic Cryptographic Module covert channel analysis

Dependencies: ADV_FSP.4 Complete Functional Specification

ADV_IMP.1 Implementation of the TSF

AGD_OPE.1 Operational user guidance

AGD_PRE.1 Preparative User guidance

Application notes: The covert channel analysis is performed only upon the cryptographic module; a search is made for the leakage of critical cryptographic security parameters from the cryptographic module, rather than a violation of an information control policy. Inappropriate handling / leakage of any critical cryptographic security parameters (covered or not) that by design and implementation lie outside the cryptographic module is not addressed by this CCA. Thus, leakage of such parameters in such designs and implementations must be investigated by other means.

Developer action elements:

AVA_CCA_(EXT).1.1D For the cryptographic module, the developer shall conduct a search for covert channels for the leakage of critical cryptographic security parameters whose disclosure would compromise the security provided by the module.

Application Note: The remainder of the TOE need not be subjected to a covert channel analysis. (Ideally, a covert channel analysis on the entire TSF would determine if TSF interfaces can be used covertly for the leakage of critical cryptographic security parameters. While such extensive covert channel analysis is more complete, it is also difficult and expensive. At this time it is considered beyond the scope of effort and cost considered reasonable for COTS medium robustness products. Consequently, covert channel analysis has been limited here to the cryptographic module, but that analysis limitation does come with some added risk of unknown leakage from other parts of the TOE.

AVA_CCA_(EXT).1.2D The developer shall provide covert channel analysis documentation.

Content and presentation of evidence elements:

AVA_CCA_(EXT).1.1C The analysis documentation shall identify covert channels in the cryptographic module and estimate their capacity.

AVA_CCA_(EXT).1.2C The analysis documentation shall describe the procedures used for determining the existence of covert channels in the cryptographic module, and the information needed to carry out the covert channel analysis.

AVA_CCA_(EXT).1.3C The analysis documentation shall describe all assumptions made during the covert channel analysis.

AVA_CCA_(EXT).1.4C The analysis documentation shall describe the method used for estimating channel capacity, based on worst-case scenarios.

AVA_CCA_(EXT).1.5C The analysis documentation shall describe the worst case exploitation scenario for each identified covert channel.

AVA_CCA_(EXT).1.6C The analysis documentation shall provide evidence that the method used to identify covert channels is systematic.

Evaluator action elements:

AVA_CCA_(EXT).1.1E The NSA evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_CCA_(EXT).1.2E The NSA evaluator shall confirm that the results of the covert channel analysis show that the cryptographic module meets its functional requirements.

AVA_CCA_(EXT).1.3E The NSA evaluator shall selectively validate the covert channel analysis through independent analysis and testing.

Application Note: The cryptographic security parameters are to be defined in the Security Target

5.2.5.2 AVA_VAN.4 Methodical vulnerability analysis

Dependencies: ADV_ARC.1 Security architecture description

ADV_FSP.2 Security-enforcing functional specification

ADV_TDS.3 Basic modular design

ADV_IMP.1 Implementation representation of the TSF

AGD_OPE.1 Operational user guidance

AGD_PRE.1 Preparative procedures

Developer action elements:

AVA_VAN.4.1D The developer shall provide the TOE for testing.

Content and presentation elements:

AVA_VAN.4.1C The TOE shall be suitable for testing.

Evaluator action elements:

AVA_VAN.4.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.4.2E The evaluator *shall perform* a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.4.3E The evaluator *shall perform* an independent, methodical vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design, security architecture description and implementation representation to identify potential vulnerabilities in the TOE.

AVA_VAN.4.4E The evaluator *shall conduct* penetration testing based on the identified potential vulnerabilities to determine that the TOE is resistant to attacks performed by an attacker possessing Moderate attack potential.

6.0 RATIONALE

This section describes the rationale for the Security Objectives and Security Functional Requirements as defined in Section 5. Additionally, this section describes the rationale for not satisfying all of the dependencies.

6.1 Rationale for TOE Security Objectives

Table 6.1 provides the mapping from Security Objectives to Threats and Policies, as well as a rationale that discusses how a threat or policy is addressed.

Table 6.1 Security Objectives to Threats and Policies Mappings

Threat/Policy	Objectives Addressing the Threat	Rationale
T.ADMIN_ERROR An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.	O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management. O.ADMIN_MULTIPLE_ROLE The TOE will provide multiple administrative roles to isolate non-overlapping administrative functions. O.MANAGE The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	O.ROBUST_ADMIN_GUIDANCE helps to mitigate this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner and to provide the administrator with instructions to ensure the TOE was not corrupted during the delivery process. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured in a way that is unsecure. O.ADMIN_MULTIPLE_ROLE plays a role in mitigating this threat by limiting the functions an administrator can perform in a given role. So for example, the Audit Administrator could not make a configuration mistake that would impact the TOE's ability to authenticate users by lowering the threshold. O.MANAGE also contributes to mitigating this threat by providing administrators the capability to view configuration settings. For example, if the Audit Administrator made a mistake when configuring the settings for events to be audited, providing them the capability to view the settings affords them the ability to review the settings and discover any mistakes that might have been made.
T.ADMIN_ROGUE An administrator's intentions may become malicious resulting in user or TSF data being compromised.	O.ADMIN_MULTIPLE_ROLE The TOE will provide multiple administrative roles to isolate non-overlapping administrative functions.	O.ADMIN_MULTIPLE_ROLE mitigates this threat to a limited degree by limiting the functions available to an administrator. This is somewhat different than the part this objective plays in countering T.ADMIN_ERROR, in that this presumes that separate individuals will be assigned separate roles. If the Audit Administrator's intentions become malicious they would not be able to impact the configuration settings affecting the TOE's authentication mechanism. On the other hand, if the Security Administrator becomes malicious they could affect the TOE's authentication mechanism, but the Audit Administrator may be able to detect those actions.
T.AUDIT_COMPROMISE	O.AUDIT_PROTECTION	O.AUDIT_PROTECT contributes to mitigating this threat by controlling access to the audit trail. No one

Threat/Policy	Objectives Addressing the Threat	Rationale
A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.	The TOE will provide the capability to protect audit information. O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated or upon completion of a function that residual biometric data could not be reused. O.SELF_PROTECTION The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.	is allowed to modify audit records, the Audit Administrator is the only one allowed to delete the audit trail. The TOE has the capability to prevent auditable actions from occurring if the audit trail is full. O.RESIDUAL_INFORMATION prevents a user not authorized to read the audit trail from access to audit information that might otherwise be persistent in a TOE resource (e.g., memory). By ensuring the TOE prevents residual information in a resource, audit information will not become available to any user or process except those explicitly authorized for that data. O.SELF_PROTECTION contributes to countering this threat by ensuring that the TSF can protect itself from users. If the TSF could not maintain and control its domain of execution, it could not be trusted to control access to the resources under its control, which includes the audit trail.
T.BYPASS An attacker may bypass any component of the biometric product and gain unauthorized authentication.	O.SELF_PROTECTION The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.	O.SELF_PROTECTION mitigates this threat by requiring that the TOE respond to physical tampering in a manner that would not allow a user to authenticate or appear to be authenticated due to the bypassing of any component of the TOE. This objective also requires that the biometric data that is transmitted between physically separate components of the TOE be encrypted and cryptographically signed, which would prevent an attacker from "inserting" data in any communication path between TOE components.
T.CRYPT_ATTACK An attacker may defeat security functions through a cryptographic attack against the algorithm, through cryptanalysis on encrypted data, or through a brute-force attack and thereby gaining unauthorized authentication.	O.CRYPTOGRAPHY_VALIDATED The TOE shall use NIST FIPS 140-2 validated cryptomodules for cryptographic services implementing FIPS-approved security functions and random number generation services used by cryptographic functions. O.CRYPTOGRAPHIC_ FUNCTIONS The TOE shall provide cryptographic functions (i.e., encryption/decryption and digital signature operations) to maintain the confidentiality and allow for detection of modification of TSF data that is transmitted between physically separated portions of the TOE, or stored outside the TOE.	O.CRYPTOGRAPHY_VALIDATED contributes to mitigating this threat by requiring FIPS-approved functions to be used, thus lessening the chance that a poorly-thought-out algorithm could be compromised by an adversary. Additionally, the requirements levied on the cryptomodule by the FIPS process, and the verification of those requirements by the FIPS labs, helps add assurance that the cryptographic module can protect itself. O.CRYPTOGRAPHIC_ FUNCTIONS specifies the cryptographic algorithms and key management requirements that have been deemed appropriate to protect the confidentiality and integrity of data that is commensurate with the level of assurance provided by the TOE. This objective also requires that the cryptographic module providing these services has been validated to meet the FIPS 140-2 requirements as specified in this PP, which provides a moderate degree of assurance that the cryptographic algorithms, and key generation components are correctly implemented.
T.CRYPTO_COMPROMISE A malicious user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed,	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated or upon completion of a function that	O.RESIDUAL_INFORMATION mitigates the possibility of malicious users or processes from gaining inappropriate access to cryptographic data, including keys. This objective ensures that the cryptographic data does not reside in a resource that has been used by the cryptographic module and then

Threat/Policy	Objectives Addressing the Threat	Rationale
modified, or deleted), thus compromise the cryptographic mechanisms and the data protected by those mechanisms.	residual biometric data could not be reused. O.SELF_PROTECTION The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure. O.DOCUMENT_KEY_LEAKAGE The bandwidth of channels that can be used to compromise key material shall be documented.	reallocated to another process. O.SELF_PROTECTION contributes to countering this threat by ensuring that the TSF can protect itself from users. If the TSF could not maintain and control its domain of execution, it could not be trusted to control access to the resources under its control, which includes the cryptographic data and executable code. O.DOCUMENT_KEY_LEAKAGE addresses this threat by requiring the developer to perform an analysis that documents the amount of key information that can be leaked via a covert channel. This provides information that identifies how much material could be inappropriately obtained within a specified time period.
T.HIGH_QUALITY_ARTIFACT An attacker may use a high quality artifact (e.g., artificial hand/fingerprint, life-size photograph, or other synthetic means) to gain unauthorized authentication.	O.AUTHENTICATION The TOE will provide a biometric authentication mechanism to authenticate users for the IT environment or non-IT environment.	In this context, forgery generally refers to the use of an artifact such that the biometric system is spoofed into accepting the artifact as coming from a live human being. It is not possible to make definitive statements on the potential for forging of biometric characteristics. Most biometric characteristics are not secret and may therefore be vulnerable to being copied. There will be varying degrees of difficulty involved. For example, it may be hard to copy a retinal pattern. This form of copying requires the use of a forgery to exploit the copy. Most biometric characteristics could, in principle, be forged given sufficient resources and justification. O.AUTHENTICATION addresses this threat by requiring a liveness test of the presented biometric characteristic. Due to the wide range of biometric technologies and the various forms of liveness tests associated with the different technologies, this PP requires that the ST Author state the type of liveness test done and what the test consists of. The intent is not that developers divulge their proprietary methods, but to provide end-users with enough information so they can intelligently compare what products perform in this regard and determine if that is suitable for their needs. The FAR number specified in this PP also contributes to mitigating this threat.
T.MIMIC An attacker may masquerade as an enrolled user by presenting their biometric characteristic that is similar, or by reproducing the biometric characteristics of the enrolled user (e.g., changing his/her voice, forging a signature, or other mean of mimicry) to gain unauthorized authentication.	O.AUTHENTICATION The TOE will provide a biometric authentication mechanism to authenticate users for the IT environment or non-IT environment. O.ROBUST_TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.	In some cases, an attacker may know that their biometric characteristics are very similar to those of an enrollee and attack that identity. This includes physical twins but is not confined to this case. The greater the number of enrollees, the more likely it is that the impostor resembles one of them. Some biometric products cannot distinguish between twins. Where the biometric product may confuse two individuals, an imposter may know which enrollees they best match and, for example, which finger to use. The risk is not confined to identical twins. In some cases, identical twins do not have identical biometric features (e.g. irises, fingerprints). In other cases, identical twins have identical biometric features (e.g. faces, DNA). As a result of FAR limitations, there may be pairs of unrelated individuals within relatively small samples, who can be reliably identified as each

Threat/Policy	Objectives Addressing the Threat	Rationale
		other. All behavioral biometrics are susceptible to mimic attacks. In a supervised environment, it is considerably more difficult to successfully mimic an enrollee without being detected. O.AUTHENTICATION addresses this threat by requiring a FAR of no more than 1 in 100,000. This threat cannot be totally mitigated and is an inherent weakness in some, if not all, of biometric technologies. O.ROBUST_TOE_ACCESS addresses this threat as it pertains to administrative accounts, since this objective requires the TOE to provide a non-biometric authentication mechanism to authenticate administrators if enabled by the Security Administrator.
T.FLAWED_DESIGN Unintentional or intentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a malicious user or program.	O.CHANGE_MANAGEMENT The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development. O.SOUND_DESIGN The design of the TOE will be the result of sound design principles and techniques; the design of the TOE, as well as the design principles and techniques, are adequately and accurately documented. O.VULNERABILITY_ANALYSIS_ TEST The TOE will undergo appropriate independent vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies.	O.SOUND_DESIGN counters this threat, to a degree, by requiring that the TOE be developed using sound engineering principles. By accurately and completely documenting the design of the security mechanisms in the TOE, including a security model, the design of the TOE can be better understood, which increases the chances that design errors will be discovered. O.CHANGE_MANAGEMENT plays a role in countering this threat by requiring the developer to provide control of the changes made to the TOE's design. This includes controlling physical access to the TOE's development area, and having an automated configuration management system that ensures changes made to the TOE go through an approval process and only those persons that are authorized can make changes to the TOE's design and its documentation. O.VULNERABILITY_ANALYSIS_TEST ensures that the design of the TOE is independently analyzed for design flaws. Having an independent party perform the assessment ensures an objective approach is taken and may find errors in the design that would be left undiscovered by developers that have a preconceived incorrect understanding of the TOE's design.
T.CORRUPTED_IMPLEMENTATION Unintentional or intentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a malicious user or program.	O.CHANGE_MANAGEMENT The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development. O.SOUND_IMPLEMENTATION The implementation of the TOE will be an accurate instantiation of its design, and is adequately and accurately documented. O.THOROUGH_FUNCTIONAL_ TESTING	O.CHANGE_MANAGEMENT plays a role in mitigating this threat in the same way that the flawed design threat is mitigated. By controlling who has access to the TOE's implementation representation and ensuring that changes to the implementation are analyzed and made in a controlled manner, the threat of intentional or unintentional errors being introduced into the implementation are reduced. In addition to documenting the design so that implementers have a thorough understanding of the design, O.SOUND_IMPLEMENTATION requires that the developer's tools and techniques for implementing the design are documented. Having accurate and complete documentation, and having the

Threat/Policy	Objectives Addressing the Threat	Rationale
	The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements. O.VULNERABILITY_ANALYSIS_ TEST The TOE will undergo appropriate independent vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies.	appropriate tools and procedures in the development process helps reduce the likelihood of unintentional errors being introduced into the implementation. Although the previous three objectives help minimize the introduction of errors into the implementation, O.THOROUGH_FUNCTIONAL_TESTING increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification, high level, and low-level design) will be discovered through testing. O.VULNERABILITY_ANALYSIS_TEST helps reduce errors in the implementation that may not be discovered during functional testing. Ambiguous design documentation, and the fact that exhaustive testing of the external interfaces is not required may leave bugs in the implementation undiscovered in functional testing. Having an independent party perform a vulnerability analysis and conduct testing outside the scope of functional testing increases the likelihood of finding errors.
T.POOR_ENROLLMENT An attacker may direct an attack against a low quality reference template and gain unauthorized authentication.	O.AUTHENTICATION The TOE will provide a biometric authentication mechanism to authenticate users for the IT environment or non-IT environment.	A low quality reference template can be caused by poor enrollment procedures, the quality of a user's biometric characteristic, or the biometric technology employed, that could lead to inferior biometric reference templates. O.AUTHENTICATION addresses this threat by requiring the TOE to provide the Enrollment Administrator a quality metric upon the enrollment of an individual. An acceptable quality metric will be dependent on the biometric technology and specific algorithms used by developers in their template generation and comparison function. Thus, a minimum quality metric is not specified in this PP. The administrative guidance documentation for the TOE will discuss quality metrics and what is acceptable for a specific TOE.
T.POOR_TEST Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities.	O.CORRECT_ TSF_OPERATION The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site. O.THOROUGH_FUNCTIONAL_ TESTING The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements. O.VULNERABILITY_ANALYSIS_ TEST The TOE will undergo appropriate independent vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies.	Design analysis determines that TOE's documented design satisfies the security functional requirements. In order to ensure the TOE's design is correctly realized in its implementation, the appropriate level of functional testing of the TOE's security mechanisms must be performed during the evaluation of the TOE. O.THOROUGH_FUNCTIONAL_ TESTING ensures that adequate functional testing is performed to ensure the TSF satisfies the security functional requirements and demonstrates that the TOE's security mechanisms operate as documented. While functional testing serves an important purpose, it does not ensure the TSFI cannot be used in unintended ways to circumvent the TOE's security policies. O.VULNERABILITY_ANALYSIS_TEST addresses this concern by requiring a vulnerability analysis be performed in conjunction with testing that goes beyond functional testing. This objective provides a measure of confidence that the TOE does not contain security flaws that may not be identified through functional testing. While these testing activities are a necessary activity for successful completion of an evaluation, this testing activity does not address the concern that the

Threat/Policy	Objectives Addressing the Threat	Rationale
		TOE continues to operate correctly and enforce its security policies once it has been fielded. Some level of testing must be available to end users to ensure the TOE's security mechanisms continue to operate correctly once the TOE is fielded O.CORRECT_ TSF_OPERATION ensures that once the TOE is installed at a customer's location, the capability exists that the integrity of the TSF (hardware and software) can be demonstrated, and thus providing end users the confidence that the TOE's security policies continue to be enforced.
T.REPLAY_RESIDUAL_IMAGE An attacker may attempt to "reuse" an authorized user's biometric residual characteristic (e.g., finger print left on capture device) to gain unauthorized access.	O.AUTHENTICATION The TOE will provide a biometric authentication mechanism to authenticate users for the IT environment or non-IT environment.	O.AUTHENTICATION addresses this threat by requiring the TOE to provide the Security Administrator the option of disallowing the same user identifier to be authenticated in consecutive attempts. This threat is a concern to TOEs where a user comes into physical contact with the TOE's capture device (e.g., fingerprint). The rule in FIA_UAU.5.2 would prevent an attacker from using any residual biometric characteristic (e.g., a residual fingerprint left on the capture device) from being "re-used" subsequent to the legitimate user being authenticated.
T.RESIDUAL_DATA Residual biometric authentication data from a previous valid user if not cleared from memory may allow an attacker to gain unauthorized authentication.	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated or upon completion of a function that residual biometric data could not be reused.	O.RESIDUAL_INFORMATION counters this threat by ensuring that TSF data is not persistent when resources are released by one user/process and allocated to another user/process. The objective also ensures that the potential for residual data to be mistakenly reused is mitigated even though a process/subject has not deallocated assigned resources.
T. REFERENCE_TEMPLATE An attacker modifies or creates a biometric reference template in storage or transmission to/from storage to gain unauthorized authentication.	O.AUTHENTICATION The TOE will provide a biometric authentication mechanism to authenticate users for the IT environment or non-IT environment.	O.AUTHENTICATION counters this threat by providing the TOE the capability to verify that the biometric package has not been modified while it is in storage or during transmission to/from storage. This objective also ensures that a biometrics package has been created by the TOE, or another trusted entity through the enrollment process.
T.TAMPER An attacker may modify or otherwise alter the software or hardware components, the connections between them thereby gaining unauthorized authentication.	O.SELF_PROTECTION The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.	O.SELF_PROTECTION addresses this threat by ensuring that TOE provides a mechanism that detects the exposure of the internal TOE components, and by entering a state in which users could not gain unauthorized authentication. The TSF self-tests required by this objective ensure that the TOE's hardware is operating correctly and the software and TSF data have not been corrupted by means other than exposing the internal components (e.g., electromagnetic interference).
T.MALICIOUS_TSF_ COMPROMISE A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified, or deleted).	O.MANAGE The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use. O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a	O.MANAGE is necessary because an access control policy is not specified to control access to TSF data. This objective is used to dictate who is able to view and modify TSF data, as well as the behavior of TSF functions. O.RESIDUAL_INFORMATION is necessary to mitigate this threat, because even if the security mechanisms do not allow an administrator to explicitly view TSF data, if TSF data were to inappropriately reside in a resource that was made

Threat/Policy	Objectives Addressing the Threat	Rationale
	protected resource is not released when the resource is reallocated or upon completion of a function that residual biometric data could not be reused. O.SELF_PROTECTION The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.	available to a administrator, that administrator would be able to inappropriately view the TSF data. O.SELF_PROTECTION requires that the TSF be able to protect itself from tampering and that the security mechanisms in the TSF cannot be bypassed. Without this objective, there could be no assurance that administrators could not view or modify TSF data or TSF executables that they are not authorized to access.
T.UNATTENDED_SESSION An attacker may gain unauthorized access to an administrator's unattended session.	O.ROBUST_TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.	O.ROBUST_TOE_ACCESS addresses this threat by requiring functionality in the TOE that places controls on administrative sessions. Administrative sessions are terminated after a Security Administrator defined time period of inactivity. Termination of an inactive administrative session reduces the risk of someone accessing the administrative console/workstation where the session was established, thus gaining unauthorized access to the session.
T.UNAUTHORIZED_ACCESS A user may gain access to administrative functions for which they are not authorized according to the TOE security policy.	O.ROBUST_TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.	O.ROBUST_TOE_ACCESS addresses this threat by mandating the TOE provide a non-biometric authentication mechanism available for administrative user authentication. Using this mechanism eliminates some of the issues of relying solely on a biometric authentication mechanism. This objective also requires the TOE provide a mechanism that can be configured to allow administrative users to establish sessions only under certain circumstances (e.g., day, time, location). This feature helps mitigate when an attacker could establish an administrative session if they were able to obtain the administrators authentication data.
T.UNIDENTIFIED_ACTIONS The administrator may fail to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.	O.AUDIT_REVIEW The TOE will provide the capability to selectively view audit information, and alert the administrator of identified potential security violations.	O.AUDIT_REVIEW helps to mitigate this threat by providing the Security Administrator with a required minimum set of configurable audit events that could indicate a potential security violation. By configuring these auditable events, the TOE monitors the occurrences of these events (e.g. set number of authentication failures) and generates an alarm once an event has occurred or a set threshold has been met. The method of alarm generation is left to the ST Author to describe since this PP attempts to include both networked and stand-alone TOEs the PP author's did not prescribe a method that might not be attainable by all types of TOEs.
T.UNKNOWN_STATE When the TOE is initially started or restarted after a failure, the security state of the TOE may be unknown.	O.MAINT_MODE The TOE shall provide a mode from which recovery or initial startup procedures can be performed. O.CORRECT_ TSF_OPERATION The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site. O.SOUND_DESIGN	O.SOUND_DESIGN works to mitigate this threat by requiring that the TOE developers provide accurate and complete design documentation of the security mechanisms in the TOE, including a security model. By providing this documentation, the possible security states of the TOE at startup or restart after failure should be documented and understood, thereby reducing the possibility that the TOE's security state could be unknown to users of the TOE. O.MAINT_MODE helps to mitigate this threat by ensuring that the TOE does not continue to operate in an insecure state when a hardware or software failure occurs. After a power failure, the TOE attempts to

Threat/Policy	Objectives Addressing the Threat	Rationale
	The design of the TOE will be the result of sound design principles and techniques; the design of the TOE, as well as the design principles and techniques, are adequately and accurately documented. O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.	automatically restore itself to a secure state. For other types of failures this is not necessary. If the TOE cannot automatically recover from a power failure or experiences a different type of failure, the TOE enters a state that disallows further biometric authentication attempts and requires the Security Administrator to follow documented procedures to return the TOE to a secure state. O.CORRECT_TSF_OPERATION addresses this threat by ensuring that the TSF runs a suite of tests to successfully demonstrate the correct operation of the TSF's underlying abstract machine (hardware and software), the TSF, and the TSF's cryptographic components at initial startup of the TOE. In addition to ensuring that the TOE's security state can be verified, the administrators can verify the integrity of the TSF's data and stored code as well as the TSF's cryptographic data and stored code. O.ROBUST_ADMIN_GUIDANCE provides administrative guidance for the secure start-up of the TOE as well as guidance to configure and administer the TOE securely. This guidance provides administrators with the information necessary to ensure that the TOE is started and initialized in a secure manner. The guidance also provides information about the corrective measures necessary when a failure occurs (i.e., how to bring the TOE back into a secure state).
P.ACCESS_BANNER The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.	O.DISPLAY_BANNER The TOE will display an advisory warning regarding use of the TOE.	O.DISPLAY_BANNER satisfies this policy by ensuring that the TOE displays banner that provides administrators with a warning about the unauthorized use of the TOE. The displaying of the banner is not required for non-administrative users, since all TOEs may not have a display device capable of displaying a banner.
P.ACCOUNTABILITY The authorized users of the TOE shall be held accountable for their actions within the TOE.	O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security-relevant events associated with users. O.TIME_STAMPS The TOE shall provide reliable time stamps and the capability for the administrator to set the time used for these time stamps. O.ROBUST_TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate O.AUTHENTICATION The TOE will provide a biometric authentication mechanism to authenticate users for the IT environment or non-IT environment.	This policy has a somewhat different meaning in this TOE than other TOEs that allow untrusted users to process user data. Untrusted users are not provided access to the TOE other than providing their biometric characteristic to the capture device. O.AUDIT_GENERATION addresses this policy by providing the Audit Administrator with the capability of configuring the audit mechanism to record the actions of a specific user, or review the audit trail based on the identity of the user. Additionally, the administrator's user identifier is recorded when any security relevant change is made to the TOE (e.g. modifying TSF data, start-stop of the audit mechanism). O.TIME_STAMPS plays a role in supporting this policy by requiring the TOE to provide a reliable time stamp (settable by only the Security Administrator). The audit mechanism is required to include the current date and time in each audit record. O.ROBUST_TOE_ACCESS supports this policy by requiring the TOE to identify and authenticate

Threat/Policy	Objectives Addressing the Threat	Rationale
		administrators prior to allowing any TOE access or any TOE mediated access on behalf of those users. This objective is necessary to counter this threat since O.AUTHENTICATION addresses the need for a biometrics authentication mechanism, and the Security administrator has the option of not requiring the use of the biometric authentication for administrative users. O.AUTHENTICATION is included since the biometric mechanism may be the only authentication mechanism required for administrators.
P.CRYPTOGRAPHIC_FUNCTIONS The TOE shall provide cryptographic functions for its own use, including encryption/decryption and digital signature operations.	O.CRYPTOGRAPHIC_FUNCTIONS The TOE shall provide cryptographic functions (i.e., encryption/decryption and digital signature operations) to maintain the confidentiality and allow for detection of modification of TSF data that is transmitted between physically separated portions of the TOE, or stored outside the TOE.	O.CRYPTOGRAPHIC_FUNCTIONS implements this policy, requiring a combination of FIPS-validation and non-FIPS-validated cryptographic mechanisms that are used to provide encryption/decryption services, as well as digital signature functions. Functions include symmetric encryption and decryption, digital signatures, as well as key generation and establishment functions.
P.CRYPTOGRAPHY_VALIDATED Where the TOE requires FIPS-approved security functions, only NIST FIPS validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e.; encryption, decryption, signature, hashing, key distribution, and random number generation services).	O.CRYPTOGRAPHY_VALIDATED The TOE shall use NIST FIPS 140-2 validated cryptomodules for cryptographic services implementing FIPS-approved security functions and random number generation services used by cryptographic functions. O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated or upon completion of a function that residual biometric data could not be reused.	O.CRYPTOGRAPHY_VALIDATED satisfies this policy by requiring the TOE to implement NIST FIPS validated cryptographic services. These services will provide confidentiality and integrity protection of TSF data while in transit to remote parts of the TOE. O.RESIDUAL_INFORMATION satisfies this policy by ensuring that cryptographic data are cleared from resources that are shared between users. Keys must be zeroized according to FIPS 140-2.
P.VULNERABILITY_ ANALYSIS_TEST The TOE must undergo appropriate independent vulnerability analysis and penetration testing to demonstrate that the TOE is resistant to an attacker possessing a medium attack potential.	O.VULNERABILITY_ANALYSIS_ TEST The TOE will undergo appropriate independent vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies.	O.VULNERABILITY_ANALYSIS_TEST satisfies this policy by ensuring that an independent analysis is performed on the TOE and penetration testing based on that analysis is performed. Having an independent party perform the analysis helps ensure objectivity and eliminates preconceived notions of the TOE's design and implementation that may otherwise affect the thoroughness of the analysis. The level of analysis and testing requires that an attacker with a moderate attack potential cannot compromise the TOE's ability to enforce its security policies.

6.2 Rationale for the Security objectives for the Environment

All of the security objectives for the environment are restatements of an assumption found in Section 3. Therefore, those security objectives for the non-IT environment trace to the assumptions trivially and are suitable for covering the assumptions.

6.3 Rationale for TOE Security Requirements

Table 6.2 maps the security functional requirements and the assurance requirements to the appropriate TOE objectives. A rationale is presented that provides the reader with a narrative of how the mapped requirement(s) are intended to satisfy the objective.

Table 6.2 - Rationale for TOE Security Requirements

Objective	Requirements Addressing the Objective	Rationale
O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.	ALC_DEL.1 AGD_PRE.1 AGD_OPE.1	ALC_DEL.1 ensures that the administrator is provided documentation that instructs them how to ensure the delivery of the TOE, in whole or in parts, has not been tampered with or corrupted during delivery. This requirement ensures the administrator has the ability to begin their TOE installation with a *clean* (e.g., malicious code has not been inserted once it has left the developer's control) version of the TOE, which is necessary for secure management of the TOE. The AGD_PRE.1 requirement ensures the administrator has the information necessary to install the TOE in the evaluated configuration. Often times a vendor's product contains software that is not part of the TOE and has not been evaluated. The Installation, Generation and Startup (IGS) documentation ensures that once the administrator has followed the installation and configuration guidance the result is a TOE in a secure configuration. The AGD_OPE.1 requirement mandates the developer provide the administrator with guidance on how to operate the TOE in a secure manner. This includes describing the interfaces the administrator uses in managing the TOE, security parameters that are configurable by the administrator, how to configure the TOE's ruleset and the implications of any dependencies of individual rules. The documentation also provides a description of how to setup and review the auditing features of the TOE. AGD_OPE.1 is also intended for non-administrative users, but could be used to provide guidance on security that is common to both administrators and non-administrators (e.g., password management guidelines).
O.ADMIN_MULTIPLE_ROLE The TOE will provide multiple administrative roles to isolate non-overlapping administrative functions.	FMT_SMR.2	FMT_SMR.2 requires that three roles exist for administrative actions: the Security Administrator, who is responsible for configuring the TOE's security policies; the Enrollment Administrator, who is restricted to enrolling users; and the Audit Administrator, who is restricted to reading and managing (e.g., backing up and deleting) the audit trail. The TSF is able to associate a human user with one or more roles and these roles isolate administrative functions in that the functions of these roles do not overlap, with the exception of invoking self-tests and the ability of

Objective	Requirements Addressing the Objective	Rationale
		the Security Administrator to review audit, which was deemed necessary for them to perform trouble shooting or determining policy settings (e.g., getting a large number of failed attempts of authentication for a lot of users may require modification to the threshold setting). The functionality of the roles, as defined by this PP, is predicated on the notion that once the TOE has been setup and is running in a stable configuration the Security Administrator would not be required to frequently administer the TOE. The Audit Administrator will probably be logging into the TOE most often to review the audit trail. Restricting the Audit Administrator's capabilities thus reduces the potential harm that could occur due to an error, or the execution of malicious code.
O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security-relevant events associated with users.	FAU_GEN.1-NIAP-0410 FAU_GEN.2-NIAP-0410 FIA_USB.1-NIAP-0415 FAU_SEL.1-NIAP-0407 FMT_MTD_(EXT).1	FAU_GEN.1-NIAP-0410 defines the set of events that the TOE must be capable of recording. This requirement ensures that the Audit Administrator has the ability to audit any security relevant event that takes place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. There is a minimum of information that must be present in every audit record and this requirement defines that, as well as the additional information that must be recorded for each auditable event. This requirement also places a requirement on the level of detail that is recorded on any additional security functional requirements an ST author adds to this PP. FAU_GEN.2-NIAP-410 ensures that the audit records associate a user identity with the auditable event. In the case of authenticated users, the association is accomplished with the user identifier. In the case of a failed authentication, the presented user identifier is associated with the event even though this identifier cannot be confirmed since these users are not authenticated. This is required since it may provide the Security Administrator with useful information (e.g., a specific user is targeted by an attacker). FAU_SEL.1-NIAP-0407 allows the Audit Administrator to configure which auditable events will be recorded in the audit trail. This provides the administrator with the flexibility in recording only those events that are deemed necessary by site policy, thus reducing the amount of resources consumed by the audit mechanism. FIA_USB.1 plays a role is satisfying this objective by requiring a binding of security attributes associated with users that are authenticated with the subjects that represent them in the TOE. This only applies to authenticated users, since the identity of unauthenticated users cannot be confirmed. Therefore, the audit trail may not always have the proper identity of the user that causes an audit record to be generated (e.g., an attacker/user providing another user's user identifier). FMT_MTD_(EXT).1 requires that the TOE either provides capture devices with assured unique identifiers or provides the capability of setting identifiers to the Security Administrator (it is left to the Security Administrator to ensure they are unique). This requirement ensures that the audit trail indicates the capture device where the event occurred.
O.AUDIT_PROTECTION The TOE will provide the capability to protect audit information.	FAU_SAR.2 FAU_STG.1-NIAP-0423	FAU_SAR.2 restricts the ability to read the audit trail to the Security Administrator and Audit Administrator, thus preventing the disclosure of the audit data to any other user. However, the TOE is not expected to prevent the disclosure of audit data if it has been archived or saved in another form (e.g., moved or copied to an

Objective	Requirements Addressing the Objective	Rationale
	FAU_STG.3 FAU_STG.NAIP-0414-1-NIAP-0429	ordinary file). The FAU_STG family dictates how the audit trail is protected. FAU_STG.1-NIAP-0423 contributes to this objective by restricting the management of the audit trail to the Audit Administrator, this includes the backing up of the audit trail. This restriction helps ensure that audit records will not be lost and the Audit Administrator and Security Administrator will be able to associate events with users. FAU_STG.3 requires that an alarm is generated when the audit trail exceeds a capacity threshold established by the Audit Administrator. This ensures that the Audit Administrator has the opportunity to manage the audit trail before it becomes full and avoiding the possible loss of audit data. FAU_STG.NAIP-0414-1-NIAP-0429 allows the Audit Administrator to configure the TOE so that if the audit trail does become full, either the TOE will prevent any events from occurring (other than actions taken by the Audit Administrator) that would generate an audit record (e.g., depending on the FAU_SEL.1-NIAP-0407 configuration, users may no longer be allowed to authenticate) or the audit mechanism will overwrite the oldest audit records with new records (thus thwarting a denial of service attack). This requirement ensures that as a default, audit records will not be overwritten, and the Audit Administrator must select the overwrite option if that is what they desire.
O.AUDIT_REVIEW The TOE will provide the capability to selectively view audit information, and alert the administrator of identified potential security violations.	FAU_SAA.1-NIAP-0407 FAU_ARP.1 FMT_MOF.1(3) FAU_SAR.3 FAU_SAR.1	FAU_SAA.1-NIAP-0407 defines the events that indicate a potential security violation and will generate an alarm. The triggers for the number of authentication failures are configurable by the Security Administrator. The failure of TSF self-tests, physical tampering, and detection of a modification of a biometrics package will generate an alarm. These events are independent of those selected for audit. For example if the Audit Administrator did not select the event of biometrics package modification in FAU_SEL, the Security Administrator could still configure the TOE to ensure that that event would generate an alarm. FAU_ARP.1 requires that the TOE generate an alarm when a potential security violation has been detected. Due to the wide range of TOE implementations, there is no specific requirement on how the alarm is to be generated. The ST author fills in the assignment of how their implementation will alert the administrator. FAU_SAR.1 provides the Audit Administrator and Security Administrator with the capability to read the audit data contained in the audit trail. This requirement also mandates the audit information be presented in a manner that is suitable for the administrators to interpret the audit trail, which is subject to interpretation. It is expected that the audit information be presented in such a way that the administrators can examine an audit record and have the appropriate information (that required by FAU_GEN.2-NIAP-410) presented together to facilitate the analysis of the audit review. FAU_SAR.3 complements FAU_SAR.1 by providing the administrators the flexibility to specify criteria that can be used to search or sort the audit records residing in the audit trail. FAU_SAR.3 requires the administrators be able to establish the audit review criteria based on a user identifier, time and day, so that the

Objective	Requirements Addressing the Objective	Rationale
		actions of a user can be readily identified and analyzed. Allowing the administrators to perform searches or sort the audit records based on dates, times, subject identities provides the capability to extract the user activity to what is pertinent at that time in order facilitate the administrator's review. It is important to note that the intent of sorting in this requirement is to allow the administrators the capability to organize or group the records associated with a given criteria. FMT_MOF.1(3) restricts the ability to control the behavior of the audit and alarm mechanism to the Security Administrator. The Security Administrator is the only user that controls the behavior of the events that generate alarms and whether the alarm mechanism is enabled or disabled.
O.AUTHENTICATION The TOE will provide a biometric authentication mechanism to authenticate users for the IT environment or non-IT environment.	FIA_UAU.5 FIA_UID.2 FIA_ENROLL_(EXT).1 FPT_ITC_(EXT).1 FPT_ITI_(EXT).1	FIA_UAU.5 requires the TOE to provide at least one biometrics authentication mechanism. This mechanism is the only mechanism that can authenticate non-administrative users and may be used at the discretion of the Security Administrator to authenticate administrative users. The rules regarding the use of the biometric authentication mechanism are specified in this requirement, including the circumstances under which the TOE provides a match/no match decision to the environment. Unlike an identification mode TOE, FIA_UID.2 requires that every user provide a user identifier before they are authenticated. This is essential for a verification mode biometrics device, and is one distinguishing factor from an identification mode biometrics device. Since a biometrics package is associated with a user identifier, it is essential to have a user supply their identifier before an authentication attempt can be made. FIA_ENROLL_(EXT).1 is critical in establishing the requirements for the enrollment of a user. This requirement specifies what a biometrics package minimally consists of, and establishes the restrictions on the creation/modification of a biometric package (which includes the reference template). This requirement also mandates that the Enrollment Administrator be presented with a quality metric upon the potential enrollment of a user. The administrative guide discusses the enrollment procedure and how the quality metric affects the ability of the TOE to satisfy its FAR/FRR numbers. FPT_ITC_(EXT).1 is necessary to ensure the confidentiality of the biometrics package is maintained when the package leaves the TOE's scope of control. Since the storage of the biometrics package is not required to be under the TOE's scope of control the storage device can be untrusted, yet the confidentiality of the package can be assured. While FPT_ITC_(EXT).1 ensures the confidentiality of the biometrics package, FPT_ITI_(EXT).1 is even more critical since it ensures the integrity of the biometrics package is maintained. This is necessary for the same reason that FPT_ITC_(EXT).1 is necessary – the storage of the biometrics package is not trusted with respect to the TOE. If the integrity of the biometrics package cannot be assured, the authentication decision generated by the TOE cannot be trusted. One of the rules in FIA_UAU.5 requires that the integrity of the package be validated before a comparison of the reference template and live template can be made.

Objective	Requirements Addressing the Objective	Rationale
O.CHANGE_MANAGEMENT The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development.	ALC_CMC.4 ALC_CMS.4 ALC_DVS.1 ALC_FLR.2 ALC_LCD.1	ALC_CMC.4 contributes to this objective by requiring the developer have a configuration management plan that describes how changes to the TOE and its evaluation deliverables are managed. The developer is also required to employ a configuration management system that operates in accordance with the CM plan and provides the capability to control who on the development staff can make changes to the TOE and its developed evidence. This requirement also ensures that authorized changes to the TOE have been analyzed and the developer's acceptance plan describes how this analysis is performed and how decisions to incorporate the changes to the TOE are made. ALC_CMC.4 also requires that the CM system use an automated means to control changes made to the TOE. If automated tools are used by the developer to analyze or track changes made to the TOE, those automated tools must be described. ALC_CMS.4 is necessary to define what items must be under the control of the CM system. This requirement ensures that the TOE implementation representation, design documentation, test documentation (including the executable test suite), user and administrator guidance, CM documentation and security flaws are tracked by the CM system. ALC_DVS.1 requires the developer describe the security measures they employ to ensure the integrity and confidentiality of the TOE are maintained. The physical, procedural, and personnel security measures the developer uses provides an added level of control over who and how changes are made to the TOE and its associated evidence. ALC_FLR.2 plays a role in satisfying the "analyzed" portion of this objective by requiring the developer to have procedures that address flaws that have been discovered in the product, either through developer actions (e.g., developer testing) or those discovered by others. The flaw remediation process used by the developer corrects any discovered flaws and performs an analysis to ensure new flaws are not created while fixing the discovered flaws. ALC_LCD.1 requires the developer to document the life-cycle model used in the development and maintenance of the TOE. This life-cycle model describes the procedural aspects regarding the development of the TOE, such as design methods, code or documentation reviews, how changes to the TOE are reviewed and accepted or rejected.

O.CORRECT_ TSF_OPERATION
The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site.
	FPT_TST_(EXT).1
	FPT_TST1(1)
	FPT_TST1(2)
		O_CORRECT_TSF_OPERATION requires two security functional requirements in the FPT class, FPT_TST. These functional requirements provide the end user (i.e., administrator) with the capability to ensure the TOE's security mechanisms continue to operate correctly in the field. FPT_TST_(EXT).1 ensures end-user tests exist to demonstrate the correct operation of the security mechanisms required by the TOE that are provided by the hardware. Hardware failures could render a TOE's software ineffective in enforcing its security policies and this requirement provides the end user the ability to discover any failures in the hardware security mechanisms. This requirement also validates the integrity of the TSF software and TSF data. If TSF software is corrupted it is possible that the TSF would no longer be able to enforce the security policies. This also holds true for TSF data, if TSF data is corrupt the TOE may not correctly enforce its security policies.
		The FPT_TST1(1) and FPT_TST1(2) functional requirements address the critical nature and specific handling of the cryptographic keys. Since the cryptographic keys have specific FIPS PUB

Objective	Requirements Addressing the Objective	Rationale
		requirements associated with them it is important to ensure that any fielded testing on the integrity of these data maintains the same level of scrutiny as specified in the FCS functional requirements. This requirement allows the Security Administrator the option of having the cryptographic self-tests executed after the generation of every key. This may not be practical for some installations, therefore it is left to the Security Administrator's discretion.

O.CRYPTOGRAPHY_VALIDATED
The TOE shall use NIST FIPS 140-2 validated cryptomodules for cryptographic services implementing FIPS-approved security functions and random number generation services used by cryptographic functions.
	FCS_BCM_(EXT).1
	FCS_CKM.1(1)
	FCS_CKM.1(2)
		This objective deals with the issue of using FIPS 140-2-approved cryptomodules in the TOE. A cryptomodule, as used in the components, is a module that is FIPS 140-2 validated (in accordance with FCS_BCM_(EXT).1); the cryptographic functionality implemented in that module are FIPS-approved security functions that have been validated; and the cryptographic functionality is available in a FIPS-approved mode of the cryptomodule. This objective is distinguished from O.CRYPTOGRAPHIC_FUNCTIONS in that this deals only with a requirement to use FIPS 140-2-validated cryptomodules where the TOE requires such functionality; it does not dictate the specific functionality that is to be used.
		FCS_BCM_(EXT).1 is an extended requirement that specifies not only that cryptographic functions that are FIPS-approved must be validated by FIPS, but also what NIST FIPS rating level the cryptographic module must satisfy. The level specifies the degree of testing of the module. The higher the level, the more extensive the module is tested.
		FCS_CKM.1(1) and FCS_CKM.1(2) mandates that the cryptomodule must generate key, and that this key generation must be part of the FIPS-validated cryptomodule.

O.CRYPTOGRAPHIC_FUNCTIONS
The TOE shall provide cryptographic functions (i.e., encryption/decryption and digital signature operations) to maintain the confidentiality and allow for detection of modification of TSF data that is transmitted between physically separated portions of the TOE, or stored outside the TOE.
	FCS_CKM.1(1)
	FCS_CKM.1(2)
	FCS_CKM.2
	FCS_CKM.4
	FCS_CKM_(EXT).2
	FCS_COP.1(1)
	FCS_COP.1(2)
	FCS_COP.1(3)
	FCS_COP.1(4)
	FCS_COP_(EXT).1
		The FCS requirements used in this PP satisfy this objective by levying requirements that ensure the cryptographic standards include the NIST FIPS publications (where possible) and NIST approved ANSI standards. The intent is to have the satisfaction of the cryptographic standards be validated through a NIST FIPS 140 validation.
		In contrast to O.CRYPTOGRAPHY_VALIDATED, this objective is to provide cryptographic functionality that is used by the TOE. The core functionality to be supported is encryption/decryption using a symmetric algorithm, and digital signature generation and verification using asymmetric algorithms. Since these operations involve cryptographic keys, how the keys are generated and/or otherwise obtained have to also be specified.
		FCS_CKM.1(1) is a requirement that a cryptomodule generate symmetric keys. Such keys are used by the TDEA encryption/decryption functionality specified in FCS_COP.1(1).
		FCS_CKM.1(2) is a requirement that a cryptomodule generate asymmetric keys. Such keys are used for cryptographic signatures as specified in FCS_COP.1(2).
		FCS_CKM.1(1), FCS_CKM.1(2) requires that the TSF validate all keys generated to assure that meet relavant standards.
		FCS_CKM_(EXT).2 requires that keys are handled appropriately and associated with the correct entities, and that transfer of keys is done

Objective	Requirements Addressing the Objective	Rationale
		with error detection. Storage of persistent secret and private keys must be done in a secure fashion. FCS_COP.1(3) requires that the TSF provide hashing services using a NIST-approved implementation of the Secure Hash Algorithm. Another way of obtaining key material for symmetric algorithms is through cryptographic key establishment, as specified in FCS_COP.1(4). Key establishment has two aspects: key agreement and key distribution. Key agreement occurs when two entities exchange public data yet arrive at a mutually shared key without ever passing that key between the two entities (for example, the Diffie-Hellman algorithm). Key distribution (FCS_CKM.2) occurs when the key is transmitted from one entity to the TOE. If the entity is electronic and a protocol is used to distribute the key, it is referred to in this PP as "Key Transport". If the key is loaded into the TOE it can be loaded electronically (e.g., from a floppy drive, smart card, or electronic keyfill device) or manually (e.g., typed in). One or more of these methods must be selected. FCS_CKM.4 provides the functionality for ensuring key and key material is zeroized. This applies not only to key that resides in the TOE, but also to intermediate areas (physical memory, page files, memory dumps, etc.) where key may appear. As previously mentioned FCS_COP.1(1) specifies that TDEA be used to perform encryption and decryption operations. FCS_COP.1(2) gives three options for providing the digital signature capability; these requirements reference the approvirate standards for each digital signature option. FCS_COP_(EXT).1 requires that any random number generation and hashing functions, respectively, are part of a FIPS-validated cryptographic module. This requirement does not mandate that the functionality is generally available, but only that it be implemented in a FIPS-validated module should other cryptographic functions need these services.
O.DISPLAY_BANNER The TOE will display an advisory warning regarding use of the TOE.	FTA_TAB.1	FTA_TAB.1 has been refined to apply only to administrative sessions, since an untrusted user does not establish a session with the TOE. In many cases the TOE may not have a display device and therefore no means of displaying a banner to untrusted users. It is expected that an administrator will have to have some type of display device to administrator the TOE (e.g., connect a console) and therefore a notice and consent banner is required.
O.DOCUMENT_KEY_ LEAKAGE The bandwidth of channels that can be used to compromise key material shall be documented.	AVA_CCA_(EXT).2	AVA_CCA_(EXT).2 requires that a covert channel analysis be performed on the entire TOE to determine the bandwidth of possible cryptographic key leakage. While there are no requirements to limit the bandwidth, the results of this analysis will provide useful guidance on what the specified lifetime of the cryptographic keys should be in order to reduce the damage due to a key compromise.
O.THOROUGH_FUNCTIONAL_ TESTING The TOE will undergo appropriate security functional testing that demonstrates the TSF	ATE_COV.2 ATE_FUN.1	In order to satisfy O.THOROUGH_FUNCTIONAL_ TESTING, the ATE class of requirements is necessary. The component ATE_FUN.1 requires the developer to provide the necessary test documentation to allow for an independent analysis of the developer's security functional test coverage. In addition, the

Objective	Requirements Addressing the Objective	Rationale
satisfies the security functional requirements.	ATE_DPT.2 ATE_IND.2	developer must provide the test suite executables and source code, which are used for independently verifying the test suite results and in support of the test coverage analysis activities. ATE_COV.2 requires the developer to provide a test coverage analysis that demonstrates the TSFI are completely addressed by the developer's test suite. While exhaustive testing of the TSFI is not required, this component ensures that the security functionality of each TSFI is addressed. This component also requires an independent confirmation of the completeness of the test suite, which aids in ensuring that correct security relevant functionality of a TSFI is demonstrated through the testing effort. ATE_DPT.2 requires the developer to provide a test coverage analysis that demonstrates depth of coverage of the test suite. This component complements ATE_COV.2 by ensuring that the developer takes into account the high-level and low-level design when developing their test suite. Since exhaustive testing of the TSFI is not required, ATE_DPT.2 ensures that subtleties in TSF behavior that are not readily apparent in the functional specification are addressed in the test suite. ATE_IND.2 requires an independent confirmation of the developer's test results, by mandating a subset of the test suite be run by an independent party. This component also requires an independent party to attempt to craft functional tests that address functional behavior that is not demonstrated in the developer's test suite. Upon successful adherence to these requirements, the TOE's conformance to the specified security functional requirements will have been demonstrated.
O.MAINT_MODE The TOE shall provide a mode from which recovery or initial startup procedures can be performed.	FPT_RCV.2-NIAP-406	This objective is met by using the FPT_RCV.2-NIAP-406 requirement, which ensures that the TOE does not continue to operate in an insecure state when a hardware or software failure occurs. Upon the failure of the TSF self-tests (including the hardware tests required by FPT_TST_(EXT).1) the TOE will enter a mode where it can no longer be assured of enforcing its security policies. Therefore, the TOE enters a state that disallows traffic flow and requires an administrator to follow documented procedures that instruct them on to return the TOE to a secure state. These procedures may include running diagnostics of the hardware, or utilities that may correct any integrity problems found with the TSF data or code. Solely specifying that the administrator reload and install the TOE software from scratch, while might be required in some cases, does not meet the intent of this requirement. An important aspect of this requirement is that upon a power failure, the TOE must attempt to automatically recover from the discontinuity. This aspect is included to eliminate the need of an administrator to have to "restart" every TOE under their purview due to a power failure at an installation.

O.MANAGE
The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
	FMT_MOF.1(1)
	FMT_MOF.1(2)
	FMT_MOF.1(3)
	FMT_MOF.1(4)
	FMT_MOF.1(5)
	FMT_MOF.1(6)
		The FMT requirements are used to satisfy this management objective, as well as other objectives that specify the control of functionality. The requirement's rationale for this objective focuses on the administrator's capability to perform management functions in order to control the behavior of security functions.
		FMT_MOF.1(1) specifies the ability of the Audit administrator to control the security function associated with audit generation. The ability to control this function has been assigned to the appropriate administrative roles. This requirement also allows the Audit Administrator to affect the events that are audited, turn audit off/on, and requires the capability exists that the Audit Administrator can determine/view the configuration settings.

Objective	Requirements Addressing the Objective	Rationale
	FMT_MOF.1(7) FMT_MOF.1(8) FMT_MOF.1(9) FMT_MTD.1(1) FMT_MTD.1(2) FMT_MTD.1(3) FMT_MTD_(EXT).1 FMT_REV.1 FAU_SAR.1 FAU_SAR.2 FAU_SAR.3 FAU_SEL.1-NIAP-0407 FAU_STG.1-NIAP-0423 FAU_STG.3 FAU_STG.NIAP-0414-1-NIAP-0429	FMT_MOF.1(2) provides the Audit Administrator and Security Administrator the ability to enable a function to help them facilitate the review of the audit trail. This requirement also ensures these administrative roles have the ability to select the event types (defined by the ST Author), and criteria that is required by this PP to enhance their ability to review the audit trail. This requirement limits the ability to perform these functions to only those users acting in the role of the Audit Administrator or Security Administrator. FMT_MOF.1(3) dictates the functionality required to manage the alarm functions of the TOE. The ability to control this function is limited to the Security Administrator and provides this role the capability of enabling or disabling the alarm function. This requirement also provides the Security Administrator with the capability to modify the behavior of the function that indicates a potential security violation. So as to ensure the mechanisms are configured as intended, the Security Administrator has the ability to view the conditions under which an alarm will be generated, and if alarm generation is enabled. FMT_MOF.1(4) The Security Administrator is the only role that is able to modify the behavior of the tests (e.g., select when they run, select a subset of the tests). This ensures that the self-tests will run no less than a frequency determined as necessary by the Security Administrator. This requirement also ensures the Security Administrator has the capability to determine that the behavior of the self-tests is configured as they intended. FMT_MOF.1(5) provides the capability for the security administrator to enable or disable the self testing of the cryptographic module after a key is generated. While the testing of the cryptographic components responsible for generating keys is important to ensure keys are generated correctly, it may be too resource consuming in some instances, and this management function provides the capability to turn the self tests off. FMT_MOF.1(6) was necessary to restrict the ability to restore the TOE to an operational mode after the TOE entered into a maintenance mode. The intent is to ensure that only the Security Administrator can restore the TOE, since this is the only administrative role that has the ability to view and configure the most critical configuration parameters. FMT_MOF.1(7) restricts the ability to enroll users to the Enrollment Administrator. Since correctly enrolling users is vital to the TOE's ability to correctly authenticate users, it was felt that it was appropriate that only users that understood the critical nature of enrollment, and were provided the necessary training would be allowed to perform enrollment. These users should be the only individuals assigned to the role of Enrollment Administrator. Since this TOE requires two authentication mechanisms (a biometric, and a non-biometric) that are to be administrated in different fashions, two management functions were deemed necessary. FMT_MOF.1(8) allows the Security Administrator to enable or disable the need for administrators to use the non-biometric authentication mechanism. FMT_MOF.1(9) provides capability to modify the behavior of the biometric authentication mechanism. This includes enabling or

Objective	Requirements Addressing the Objective	Rationale
		disabling of the liveness check, and setting the threshold that affects level of a match required in the comparison of the reference template and live template. This capability is also restricted to the Security Administrator role. FMT_MTD.1(1) is necessary to restrict the ability to modify the cryptographic security data that is used by the TOE to allow it to correctly enforce its security policy. A FIPS 140-2 module that is validated to provide a security level 3 for roles (as is required for this PP) ensures that those users that can manage the cryptographic module are uniquely identified individuals. This requirement levies a restriction, in that that individual most be assigned to the Security Administrative role as well. FMT_MTD.1(2) provides the capability of setting the date and time that is used to generate time stamps to the Security Administrator. These timestamps are critical since they are used in the audit trail to establish the sequence of events that have occurred in the TOE. It is important to allow this functionality, due to clock drift and other circumstances, but the capability must be restricted, and this requirement ensures only the Security Administrator can change the system time and date that generates this timestamp. Since the essence of a biometrics TOE is to perform authentication, FMT_MTD.1(3) ensures that only the Security Administrator has the flexibility to configure the TOE such that it behaves as required by their operational constraints. This requirement goes somewhat hand-in-hand with FMT_MOF.1(9). The CC includes both the management (modifying the behavior) of a security function, and management of TSF data. It is sometimes confusing where to place certain aspects pertaining to the management of a TSF function, since managing TSF data can have an affect on the behavior of a TSF function. FMT_MTD.1(3) identifies TSF data that will have an impact on the behavior of this function and places restrictions on what administrative role can mange that TSF data. This requirement identifies the TSF data the PP authors felt was essential in allowing a Security Administrator to manage the TOE. FMT_MTD_(EXT).1 ensures that if the TOE provides the capability to set the identifier that the function is restricted to the Security Administrator. If the setting of the identifier is a provided capability, the TOE must also provide the Security Administrator with the capability to query the capture devices' identifiers. This capability provides the Security Administrator the ability to read all the capture device identifiers so that when they set an identifier they will know that it is unique. FMT_REV.1 ensures that the Security Administrator has the ability to revoke the assignment of a role to a specific user. This revocation is immediate and applies to all administrative roles identified in this PP. This helps a Security Administrator control what capabilities users have, if any, with respect to managing the TOE. FAU_SAR.1 ensures that the Audit and Security Administrators have the capability to review the audit records and that they are presented in a manner that is suitable for review (e.g., the administrators can construct a sequence of events provided the necessary events were audited). FAU_SAR.2 restricts the ability to read the audit records to the Audit Administrator and Security Administrator. This capability exists for

Objective	Requirements Addressing the Objective	Rationale
		the Security to help facilitate any trouble shooting that they may have to perform. FAU_SAR.3 provides the Security Administrator and Audit Administrator with the ability to selectively review the contents of the audit trail based on established criteria. This capability allows the administrators to focus their audit review to what is pertinent at that time. FAU_STG.1-NIAP-0423 specifies that only the Audit Administrator can backup and delete the audit trail. This prevents the accidental or intentional deletion of the audit trail by administrators acting in another role. FAU_STG.3 provides the Audit Administrator the capability to establish a threshold of audit trail capacity, that when reached an alarm will be generated. If the audit trail becomes full FAU_STG.NIAP-0414-1-NIAP-0429 provides the Audit Administrator the option of having the TOE prevent auditable events from occurring, or having the TOE overwrite the oldest audit records. While the option of overwriting old audit records does not technically prevent audit data loss, it is provided to the Audit Administrator as an option to prevent a possible denial-of-service. FAU_SEL.1-NIAP-0407 provides the Audit Administrator the ability to define what events will be included or excluded from the list of audited events. This allows a site to audit only those events that are of interest to them and reduces the amount of unwanted audit data that is collected.
O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated or upon completion of a function that residual biometric data could not be reused.	FDP_RIP.2 FCS_CKM.4	FDP_RIP.2 is used to ensure the contents of resources are not available once the TSF is finished processing the TSF data, in addition to requiring that the data be made unavailable when reallocated to another subject. The requirement was refined since it is possible that the resource will not be deallocated or reallocated (e.g., memory assigned to a subject, never released and that memory would be used in subsequent authentication attempts. FCS_CKM.4 addresses this objective by ensuring the cryptographic keys are zeroized and are unavailable to unauthorized users.
O.ROBUST_TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate	AVA_VAN.4 FIA_AFL.1-NIAP-0425(1)-(3) FIA_ATD.1 FIA_SOS.1 FIA_SOS.2 FIA_UAU.2 FIA_UAU.5	The AVA_VAN.4 requirement as applied to the local authentication mechanism. The evaluator performs penetration testing, to confirm that the potential vulnerabilities cannot be exploited in the operational environment for the TOE. Penetration testing is performed by the evaluator assuming an attack potential of moderate. This requirement ensures the evaluator has performed an analysis of the authentication mechanism to ensure the probability of guessing a user's authentication data would require a high-attack potential, as defined in Annex B of the CEM. FIA_UID.2 plays a small role in satisfying this objective by ensuring that every user is identified before the TOE performs any mediated functions. A distinction between a verification mode and identification mode TOE is that the user must be identified and the comparison of the live biometric templates is done with the reference template associated with the user provided identity. While an attacker may continue attempting to authenticate by cycling through all the

Objective	Requirements Addressing the Objective	Rationale
	FIA_UAU.7 FIA_UID.2 FTA_TSE.1 FTA_SSL.3	user identifiers (in essence manually performing what an identification mode TOE performs automatically). FIA_AFL is used to address this threat. In the context of this objective, the key is ensuring that an untrusted user cannot access an administrative account. FIA_AFL.1-NIAP-0425 has three iterations that provide a detection mechanism for unsuccessful authentication attempts for failed attempts against a single user identifier, consecutive failed attempts against any user identifiers, and failed attempts against an administrator account. For this objective, the third iteration is what plays a role in partially meeting the objective. The requirement enables a Security Administrator settable threshold that prevents unauthorized users from gaining access to an administrators account by locking the targeted account until the Security Administrator takes some action (e.g., re-enables the account) or for some Security Administrator defined time period. Thus, limiting an unauthorized user's ability to gain unauthorized access to the TOE. FIA_ATD.1 defines the attributes of users, including a user identifier that is used to by the TOE to determine a user's identity and enforce what type of access the user has to the TOE (e.g., the TOE associates a user identifier with any role(s) they may assume). This requirement allows a human user to have more than one user identity assigned, so that a single human user could assume all the roles necessary to manage the TOE. This requirement ensures that untrusted users cannot be associated with a role and reduces the possibility of a user obtaining administrative privileges. This TOE is somewhat unique in that it requires two authentication mechanisms, a biometric authentication mechanism and a non-biometric authentication mechanism for administrative access. The required use of these two authentication mechanisms is dictated at the option of the Security Administrator. If the Security Administrator desires, the non-biometric authentication is mandatory for administrative authentication. The FIA_SOS.1 requirement prescribes the metrics that must be satisfied when using this mechanism. The PP authors intentionally did not dictate that a password mechanism be required and allowed for other types of mechanisms (e.g. a PIN, Token). In any case, FIA_SOS.1 requires that the non-biometric authentication mechanism provide the ability for administrators to choose their "secret" in a space that cannot be guessed at random in less than probability of one in 1 x 10^8. It was thought that a PIN that consisted of 8 digits (0-9) could satisfy this requirement. Since this function is used solely for administrators, the intention is that administrators would be able to select their "secret" from this space. Since administrators may be responsible for administering a number of TOEs, it was deemed impractical to have the TOE generate the secrets and require the administrators to remember them. FIA_SOS.2 is directly related to the ability of the TOE to "generate" a secret based on a user's biometric characteristic. The PP authors believe that the TOE essentially generates a secret used to authenticate users based upon proprietary algorithms used by developers to generate a reference template and subsequent live templates for comparison. This authentication is optional, at the Security Administrator's discretion, for administrative users. The thinking is that if the capture device experience problems, the

Objective	Requirements Addressing the Objective	Rationale
		Security Administrator may want to have an account that can administer the TOE that does not rely on the biometric authentication mechanism. The PP authors struggled with trying to define a quality metric that they could impose on the TOE, but given the nature of the various technologies, it was felt that the FAR and FRR numbers would have to suffice in ensuring the TOE generates acceptable reference templates, which plays a significant role in the quality of the generated secret. The authors understand that the FAR and FRR numbers are dependent on other factors (e.g., the population of users enrolled, the quality of the biometric characteristic, the number of users enrolled), but this specification was felt the best that could be done at this time given the nature of biometric technologies and their application. FIA_UAU.2 simple requires that administrative users are authenticated before they perform any administrative actions. This is an unusual TOE, in that the only users of the TOE are administrative users. Untrusted users have no access to the resources resident in the TOE and have no interaction with the TOE other to authenticate themselves for access to a portal, or for possible mediation performed by another IT entity, therefore this requirement was refined to address only administrative users. FIA_UAU.5 provides the Security Administrator with the flexibility to determine the degree of authentication that is required of users that have access to the TOE itself (i.e. administrative users). This requirement provides the necessary rules for both biometric and non-biometric authentication mechanisms. The ability to configure the biometric authentication mechanism, and to require the use of the non-biometric authentication mechanism affords the Security Administrator the ability to dictate the degree of user authentication necessary to perform administrative activities. FIA_UAU.7 ensures that no feedback that affects their ability to circumvent the biometric authentication mechanism is presented to the user when they attempt to authenticate. The TOE is allowed to provide information that would allow the user to use the authentication mechanism in a correct manner (e.g., center your finger and press firmly, speak louder and slowly), but not provide information that may allow alteration to their presentation that would thwart the mechanism (e.g., you failed the liveness check, your comparison failed to pass the threshold by a factor of X). FTA_TSE.1 is used to control the ability of an administrator to establish a session with the TOE. The ability of a the Security Administrator to determine which users are able to administrate the TOE at a specific range of time, and from a specific location (this may only apply in a networked TOE) affords the TOE the ability to limit the exposure of the TOE to an attacker attempting to establish an administrative session. For example, if Security Administrator Joe, is only allowed to establish a session from 8-5, M-F, an attacker attempting to establish a session other than those hours would not succeed, regardless of them possessing the administrator’s authentication data. FTA_SSL.3 contributes to satisfying this objective by limiting the exposure of an administrative session that is inactive for whatever reason. If an administrative session becomes inactive for a Security Administrator defined period, the session is terminated. This requirement applies both to remote and direct connections to the

Objective	Requirements Addressing the Objective	Rationale
		TOE.

O.SELF_PROTECTION
The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.
	ADV_ARC.1
	FPT_ITT.1(1)
	FPT_ITT.1(2)
	FPT_TST_(EXT).1
	FPT_TST1(1)
	FPT_TST1(2)
	FPT_PHP_(EXT).1
	FPT_PHP.3
		ADV_ARC.1 provides the security architecture description of the security domains maintained by the TSF that are consistent with the SFRs**.** Since self-protection is a property of the TSF that is achieved through the design of the TOE and TSF, and enforced by the correct implementation of that design, self-protection will be achieved by that design and implementation.
		FPT_ITT.1(1) is necessary to satisfy this objective because it ensures that TSF data that is transmitted between components of the TOE is encrypted to prevent the disclosure of information. This data would include the live template as it leaves the capture device, or as it is transmitted between other parts of the TOE. This would also include any TSF data that is sent from an administrative console to the TOE if that console is “networked” with the TOE. This would not apply to TSF data that is configured from a console that is connected via a communication path (e.g., serial cable, USB port) that ensures the data cannot be disclosed. The disclosure of TSF data could create an opportunity for the TOE to be rendered ineffective in enforcing its security policies.
		FPT_ITT.1(2) ensures the integrity of the TSF data is maintained as it is transmitted between various parts of the TOE. Ensuring the integrity of the TSF data is crucial in order to ensure the TSF can enforce its security policies.
		FPT_TST_(EXT).1 provides capability for the administrators to ensure that the TSF hardware is operating correctly, and that the resident TSF data and TSF software have not been corrupted. This aspect is critical in the administrator’s determination that the TSF can indeed protect itself, or of the fact that something has happened to bring into question the TSF’s ability to protect itself.
		FPT_TST1(1) and FPT_TST1(2) are used to ensure that the components used in generating cryptographic keys are working correctly. Since cryptography plays an important role in the TSF’s ability to enforce security policy, this requirement contributes significantly to this objective.
		FPT_PHP_(EXT).1 plays a diminished role in satisfying this objective in that it can generate an alarm and audit record notifying the Security Administrator and Audit Administrator that a potential physical attack has been mounted against the TOE. This notification affords the administrators the opportunity to inspect the TOE and determine if the TOE has been physically compromised.
		FPT_PHP.3 goes one step further than FPT_PHP_(EXT).1 since it causes the TOE to ensure that if it is physically compromised that the security policies cannot be circumvented. FPT_PHP.3 is important, since due to the nature of biometric TOE installations it might not be possible to react to a physical attack even given a notification as required by FPT_PHP_(EXT).1.

Objective	Requirements Addressing the Objective	Rationale
O.SOUND_DESIGN The design of the TOE will be the result of sound design principles and techniques; the design of the TOE, as well as the design principles and techniques, are adequately and accurately documented.	ADV_ARC.1 ADV_INT.1 ADV_FSP.5 ADV_TDS.4	There are two different perspectives for this objective. One is from the developer's point of view and the other is from the evaluator's. The ADV class of requirements is levied to aide in the understanding of the design for both parties, which ultimately helps to ensure the design is sound. ADV_ARC.1 The security architecture description will be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document (ADV_TDS.4). It will describe the security domains maintained by the TSF consistently with the SFRs as well as how the TSF initialization process is secure. The security architecture description will demonstrate that the TSF protects itself from tampering and that the TSF prevents bypass of the SFR-enforcing functionality. ADV_INT.1ensures that the design of the TOE has been performed using good software engineering design principles that require a modular design of the TSF. Modular code increases the developer's understanding of the interactions within the TSF, which in turn, potentially reduces the amount of errors in the design. Having a modular design is imperative for evaluator's to gain an appropriate level of understanding of the TOE's design in a relatively short amount of time. The appropriate level of understanding is dictated by other assurance requirements in this PP (e.g., ATE_DPT.2, AVA_CCA_(EXT).2, AVA_VAN.4). ADV_FSP.5 requires that the interfaces to the TSF be completely specified. In this TOE, the interface consists of the interface presented to the untrusted user (i.e., the capture device), as well as the interface presented to administrators (e.g., administrative commands). If the TOE provides a network interface, a specification of the network interface (including the network interface hardware) is critical in understanding what functionality is presented to untrusted users and how that functionality fits into the enforcement of security policies. Some network protocols have inherent flaws and users have the ability to provide the TOE with network packets crafted to take advantage of these flaws. The routines/functions that process the fields in the network protocols allowed (e.g., TCP, UPD, ICMP, any application level) must fully specified: the acceptable parameters, the errors that can be generated, and what, if any, exceptions exist in the processing. The functional specification of the hardware interface (e.g., network interface card) is also extremely critical. Any processing that is externally visible performed by NIC must be specified in the functional specification. .Having a complete understanding of what is available at the TSF interface allows one to analyze this functionality in the context of design flaws.

Objective	Requirements Addressing the Objective	Rationale
		TDS.4 - Provides a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design. The design will describe: the structure of the TOE in terms of subsystems; the TSF in terms of modules; identify all subsystems of the TSF; provide a description of each subsystem of the TSF; a description of the interactions among all subsystems of the TSF; a mapping from the subsystems of the TSF to the modules of the TSF; describe each SFR-enforcing module in terms of its purpose; describe each SFR-enforcing module in terms of its SFR-related interfaces; return values from those interfaces, and called interfaces to other modules; describe each SFR-supporting or SFR-non-interfering module in terms of its purpose and interaction with other modules; the mapping shall demonstrate that all behavior described in the TOE design is mapped to the TSFIs that invoke it. The design, as required by ADV_TDS.4 , provides the evaluator with the details of the TOE's design and describes at a module level how the design of the TOE addresses the SFRs. This level of description provides the detail of how modules interact within the TOE and if a flaw exists in the TOE's design. This requirement also mandates that the interfaces presented by modules be specified. Having knowledge of the parameters a module accepts, the errors that can be returned and a description of how the module works to support the security policies allows the design to be understood at its lowest level. ADV_TDS.4 also o ensures that the levels of decomposition of the TOE's design are consistent with one another. This is important, since design decisions that are analyzed and made at one level (e.g., functional specification) that are not correctly designed at a lower level may lead to a design flaw. This requirement helps in the design analysis to ensure design decisions are realized at all levels of the design.
O.SOUND_IMPLEMENTATION The implementation of the TOE will be an accurate instantiation of its design, and is adequately and accurately documented.	ADV_TDS.4 ADV_IMP.1 ADV_INT.1 ALC_TAT.1	TDS.4 - Provides a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design. The design will describe: the structure of the TOE in terms of subsystems; the TSF in terms of modules; identify all subsystems of the TSF; provide a description of each subsystem of the TSF; a description of the interactions among all subsystems of the TSF; a mapping from the subsystems of the TSF to the modules of the TSF; describe each SFR-enforcing module in terms of its purpose; describe each SFR-enforcing module in terms of its SFR-related interfaces; return values from those interfaces, and called interfaces to other modules; describe each SFR-supporting or SFR-non-interfering module in terms of its purpose and interaction with other modules; the mapping shall demonstrate that all behavior described in the TOE design is mapped to the TSFIs that invoke it. The design, as required by ADV_TDS.4 , provides the evaluator with the details of the TOE's design and describes at a module level how the design of the TOE addresses the SFRs. This level of description provides the detail of how modules interact within the TOE and if a flaw exists in the TOE's design. This requirement also mandates that the interfaces presented by modules be specified. Having knowledge of the parameters a module accepts, the errors that can be returned and a description of how the module works to support the security policies allows the design to be understood at its lowest level. ADV_TDS.4 also o ensures that the levels of decomposition of the TOE's design are consistent with one another. This is important, since design decisions that are analyzed and made at one level (e.g., functional specification) that are not correctly designed at a lower level may lead to a design flaw. This requirement helps in the design

Objective	Requirements Addressing the Objective	Rationale
		analysis to ensure design decisions are realized at all levels of the design. ADV_IMP.1was chosen to ensure evaluators have full access to the source code. If the evaluators are limited in their ability to analyze source code they may not be able to determine the accuracy of the implementation or the adequacy of the documentation. Often times it is difficult for an evaluator to identify the complete sample of code they wish to analyze. Often times looking at code in one subsystem may lead the evaluator to discover code they should look at in another subsystem. Rather than require the evaluator to "re-negotiate" another sample of code, the complete implementation representation is required. When performing the activities associated with the ADV_INT.1 requirement, the evaluators will ensure that the architecture of the implementation is modular and consistent with the architecture presented in the low-level design. Having a modular implementation provides the evaluators with the ability to more easily assess the accuracy of the implementation, with respect to the design. If the implementation is overly complex (e.g., circular dependencies, not well understood coupling, reliance on side-effects) the evaluator may not have the ability to assess the accuracy of the implementation. ALC_TAT.1 provides evaluators with information necessary to understand the implementation representation and what the resulting implementation will consist of. Critical areas (e.g., the use of libraries, what definitions are used, compiler options) are documented so the evaluator can determine how the implementation representation is to be analyzed.
O.TIME_STAMPS The TOE shall provide reliable time stamps and the capability for the administrator to set the time used for these time stamps.	FPT_STM.1 FMT_MTD.1(2)	FPT_STM.1 requires that the TOE be able to provide reliable time stamps for its own use and therefore, partially satisfies this objective. Time stamps include date and time and are reliable in that they are always available to the TOE, and the clock must be monotonically increasing. FMT_MTD.1(2) satisfies the rest of this objective by providing the capability to set the time used for generating time stamps to the Security Administrator. This functionality allows the Security Administrator to ensure the time and date are correctly set, while restricting this function from unauthorized use.
O.VULNERABILITY_ANALYSIS_TEST The TOE will undergo appropriate independent vulnerability analysis and penetration testing to demonstrate the design and implementation of the TOE does not allow attackers with medium attack potential to violate the TOE's security policies.	AVA_VAN.4	AVA_VAN.4 requires the evaluator to perform a search of public domain sources to identify potential vulnerabilities in the TOE. The evaluator will perform an independent, methodical vulnerability analysis of the OE using the guidance documentation, functional specification, TOE design, security architecture description and implementation representation to identify potential vulnerabilities in the TOE. The evaluator will conduct penetration testing based on the identified potential vulnerabilities to determine that the TOE is resistant to attacks performed by an attacker possessing **Moderate** attack potential. For those vulnerabilities that are not eliminated by the developer, a rationale must be provided that describes why these vulnerabilities cannot be exploited by a threat agent with a moderate attack potential, which is in keeping with the desired assurance level of this TOE.

Objective	Requirements Addressing the Objective	Rationale

6.4 Rationale for Assurance Requirements

The assurance selection was based on:

- recommendations documented in the GIG;
- DoD Instruction 8500.1; and
- the postulated threat environment.

The EAL definitions and assurance requirements in Part 3 of the CC were reviewed and the *Medium Robustness Assurance Package* as defined in Section 5.2 was believed to best achieve the goal of addressing circumstances where developers and users require a moderate to high level of independently assured security in commercial products. This collection of assurance requirements require TOE developers to gain assurance from good software engineering development practices which, though rigorous, do not require substantial specialist knowledge, skills, and other resources. Rationale for individual assurance requirements is provided in Table 6.2.

The Government's guidance in the GIG was consulted and found to also support the chosen assurance package. Specifically, the GIG states that medium robustness security services and mechanisms provide for additional safeguards above the DoD minimum and require good assurance security design as specified in EAL3 or greater.

The postulated threat environment specified in Section 3 of this PP was used in conjunction with the Information Assurance Technical Framework (IATF) Robustness Strategy guidance to derive the chosen assurance level.

These three factors were taken into consideration and the conclusion was that the medium robustness assurance package was the appropriate level of assurance.

6.5 Rationale for Not Satisfying All Dependencies

Each functional requirement, including **extended** requirements was analyzed to determine that all dependencies were satisfied. All requirements were then analyzed to determine that no additional dependencies were introduced as a result of completing each operation.

Table 6.3 identifies the functional requirement, its correspondent dependency and the analysis and rationale for not supporting the CC defined dependency in this PP.

Table 6.3 - Broken Dependency Rationale

Requirement	Dependency	Dependency Analysis and Rationale
FIA_UAU.1 FIA_UAU.2 FMT_SMR.2	FIA_UID.1	This dependency is satisfied with the inclusion of requirement FIA_UID.2. This requirement is hierarchical to FIA_UID.1 and is sufficient to satisfy the dependency for these requirements.
FMT_MOF.1 FMT_MTD.1 FMT_REV.1	FMT_SMR.1	This dependency is satisfied with the inclusion of requirement FMT_SMR.2. This requirement is hierarchical to FMT_SMR.1 and is sufficient to satisfy the dependency for these requirements.
FCS_CKM.1 FCS_CKM.4	FMT_MSA.2	This dependency is satisfied by placing strict requirements on the values of attributes of the cryptographic module in the associated FCS requirements. Therefore, FMT_MSA.2 is not necessary to satisfy the requirement of only secure values being assigned to secure attributes.
FMT_MOF.1 FMT_MTD.1	FMT_SMF.1	The requirements FMT_MOF.1, and FMT_MTD.1 express the functionality required by the TSF to provide the specified functions to manage TSF data, security attributes, and management functions. These requirements make clear that the TSF has to provide the functions to manage the identified data, attributes, and functions. Therefore, FMT_SMF.1 is not necessary.

6.6 Rationale for Extended requirements

The rationale for the inclusion of the extended requirements found in this PP is presented in Table 6.4. Due to the unique nature of biometric technologies the PP authors found it necessary to write extended requirements since the existing CC requirement did not capture the security functionality desired.

Table 6.4 Rationale for Extended Requirements

Extended Requirement	Identifier	Rationale
FAU_ENROLL_(EXT).1	Enrollment	This requirement is necessary because the CC does not contain an SFR that addresses the desired security functionality required for the enrollment of a user in a biometrics TOE. This requirement specifically states what is minimally required in a biometrics package and the constraints regarding access and modification of the biometrics package.
FCS_BCM_(EXT).1	Baseline cryptographic module	This extended requirement is necessary since the CC does not provide a means to specify a cryptographic baseline of implementation.
FCS_CKM_(EXT).2	Cryptographic Key Handling and Storage	This extended requirement is necessary since the CC does not provide a means to specify a cryptographic key handling and storage implementation.
FCS_COP_(EXT).1	Random Number Generation	This extended requirement is necessary since the CC cryptographic operation components are focused on specific algorithm types and operations requiring specific key sizes
FMT_MTD_(EXT).1	Management of TSF data (Capture device unique identifier)	This extended requirement is necessary because the PP authors did not want to require the TOE to provide the capability to query and set the capture device identifier if the TOE developer uses some means to ensure the capture device identifiers are unique (e.g., serial number). The CC does not contain an existing requirement that captures the intent of this extended requirement.
FPT_ITC_(EXT). 1	TSF confidentiality	This extended requirement is necessary because the CC does not contain a requirement that specifies the desired functionality. The PP authors did not want to require the storage device be aware of the cryptography used or to have the cryptographic keys to decrypt the data.

Extended Requirement	Identifier	Rationale
FPT_ITI_(EXT).1	TSF detection of modification	This extended requirement is necessary because the CC does not contain a requirement that specifies the desired functionality. The PP authors did not want to require the storage device be aware of the cryptography used or to have the cryptographic keys to sign the biometric package.
FPT_PHP_(EXT).1	Detection of physical attack	This extended requirement is necessary because the existing CC requirements do not allow for identifying the specific scenarios the TOE must detect.
FPT_TST_(EXT).1	TSF testing	This extended requirement is necessary to capture the notion of the TOE to verify the integrity of the TSF software. Additionally, the TSF data set that is subject to these tests was reduced to address the notion that it does not make sense to test the integrity of some TSF data (e.g., audit data) and this extended requirement address that.
AVA_CCA_(EXT).2	Systematic Cryptographic Module Covert Channel Analysis	This extended requirement is necessary since the CC does not have requirements to perform a covert channel analysis on information that does not have an information flow control policy. This requirement ensures that the bandwidth of critical security parameters (e.g., keys) associated with the cryptographic module is documented.

7.0 REFERENCES

1) *Common Criteria for Information Technology Security Evaluation,* CCIB-98-031 Version 2.1, August 1999.

   [1a] Common Criteria Implementation Board, Common Criteria for Information Technology Security Evaluation, CCMB-2006-09-003, Version 3.1, September 2006.

2) *BioAPI Specification,* Version 1.1, March 16, 2001.

3) *Department of Defense Chief Information Officer Guidance and Policy Memorandum No. 6-8510,* Guidance and Policy for the Department of Defense Global Information Grid Information Assurance (GIG), *June 2000.*

4) *Department of Defense Directive, Information Assurance,* 8500.1, October 24, 2002.

5) *Department of Defense Instruction, Information Assurance Implementation*, 8500.2, February 6, 2003.

6) *Federal Information Processing Standard Publication (FIPS-PUB) 140-2,* Security Requirements for Cryptographic Modules, *May 25, 2001.*

7) *Federal Information Processing Standard Publication (FIPS-PUB) 197,* Specification for the Advanced Encryption Standard (AES), November 26, 2001.

8) *Information Assurance Technical Framework*, Version 3.0, September 2000.

8.0 TERMINOLOGY

8.1 Specific Biometrics Terminology

Attack -- An act attempting to violate the security policy of an IT system.

Attacker - An attacker is any individual who is attempting to subvert the operation of the biometric system. The intention may be either to subsequently gain illegal entry to the portal or to deny entry to legitimate users.

Attempt – The submission of a biometric sample to a biometric system for identification or verification.

Authentication/Authenticate, Biometric – The biometric process of either identifying or verifying a user.

Authorization -- Permission, granted by an entity authorized to do so, to perform functions and access data.

Authorized user -- An authenticated user who may, in accordance with a Target of Evaluation Security Policy, perform an operation.

Best Match – The biometric presented is not 100% exactly the same as the reference user template but is the closest match.

Biometric – Measurable physical characteristic or personal behavioral trait used to recognize the identity or verify the claimed identity of an individual.

Biometric Data – The extracted information taken from the biometric sample and used either to build a reference template or to compare against a previously created reference template.

Biometric Package – Record created by the biometric TOE that cryptographically bind the user's identity and additional information with the biometric template for storage.

Biometric Raw Data -- The initial data from a biometric sensor device from which a biometric template is derived.

Biometric Record -- The biometric raw data, biometric sample, and/or the biometric template of an individual.

Biometric Sample – Data representing a biometric characteristic of a user as captured by a biometric system.

Biometric System – An automated system capable of capturing a biometric sample from a user, extracting biometric data from that sample, comparing the biometric data with that contained in one or more reference templates, deciding how well they match, and indicating whether or not an authentication of identity has been achieved.

Capture – The process of taking a biometric sample from the user.

Claimed user identifier - The name or index of a claimed user identity, used by a biometric system for verification.

Comparison – The process of comparing biometric data with a previously stored reference template or templates.

Enrollee – A person who has a biometric reference template stored in a biometric package.

Enrollment – The process of collecting biometric samples from a user and the subsequent preparation, encryption, and storage of biometric reference templates representing that person's identity.

Exact Match – The biometric presented is 100% exactly the same as the reference user template.

Failure to Acquire -- Failure of a biometric system to capture and extract biometric data.

Failure to Acquire Rate -- The frequency of a failure to acquire.

Failure-to-Enroll – Any irrecoverable failure in the enrollment process.

Failure-to-Enroll Rate - The probability that a biometric system will have a failure-to-enroll.

False Acceptance – When a biometric system incorrectly identifies an individual or incorrectly authenticates an impostor against a claimed identity.

False Acceptance Rate (FAR) – The probability that a biometric system will incorrectly identify an individual or will fail to reject an imposter. It is stated as follows:

FAR = NFA/NIIA or FAR=NFA/NIVA

Where **FAR** is the false acceptance rate

Where **NFA** is the number of false acceptances

Where **NIIA** is the number of imposter identification attempts

Where **NIVA** is the number of imposter verification attempts

False Rejection – When a biometric system fails to identify an enrollee or fails to verify the legitimate claimed identity of an enrollee.

False Rejection Rate (FRR) – The probability that a biometric system will fail to identify an enrollee, or verify the legitimate claimed identity of an enrollee. It is stated as follows:

FRR=NFR/NEIA or **FRR=NFR/NEVA**

Where **FRR** is the false rejection rate

Where **NFR** is the number of false rejections

Where **NEIA** is the number of enrollee identification attempts

Where **NEVA** is the number of enrollee verification attempts

Identification/Identify, Biometric – The one-to-many process of comparing a submitted biometric sample against all of the biometric reference templates on file to determine whether it matches any of the templates and, if so, the identity of the enrollee whose template was matched. The biometric system using the one-to-many approach is seeking to find an identity amongst a database rather than authenticate a claimed identity. Contrast with "Authentication".

Identity -- A representation (e.g., a string) uniquely identifying an authorized user.

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

Match Score – A numeric value or set of values derived from the comparison by the biometric system of a biometric sample with a template.

Matching -- The process of comparing a biometric sample against a previously stored template and scoring the level of similarity.

Portal – The logical or physical point beyond which the protected assets reside. For example, a physical portal may be the locking mechanism on a door. A logical portal may be an authentication measure taken prior to gaining access to a computer.

Physical/Physiological Biometric – A biometric that is characterized by a physical characteristic rather than a behavioral trait.

Replay attack – An attack in which a valid data transmission is maliciously or fraudulently repeated, either by the originator or by an adversary who intercepts the data and retransmits it, possibly as part of an imposter attack.

Secure State – A condition of normalcy, which occurs when all functions operate securely, as designed.

Template – Data that represents the biometric measurement of an enrollee, used by a biometric system for comparison against subsequently submitted biometric samples.

Threshold – The acceptance or rejection of biometric data is dependent on the match score falling above or below a defined limit. The threshold may be adjustable so that the biometric system can be more or less strict, depending on the requirements of any given biometric application.

Trusted user identifier – The name or index of a user identity that is derived from a trusted source.

User -- Any entity (human user or external IT entity) outside a Target of Evaluation that interacts with the Target of Evaluation.

Verification, Biometric – The one-to-one process of comparing a submitted biometric sample against the biometric reference template of a single enrollee whose identity is being claimed, to determine whether it matches the enrollee's template. Contrast with Biometric "Identification".

Zero Effort Forgery – An arbitrary attack on a specific enrollee identity in which the imposter masquerades as the claimed enrollee using his or her own biometric sample.

8.2 Common Protection Profile Terminology

In the Common Criteria, many terms are defined in Section 2.3 of Part 1. The following are a definitions of terms some of which are used in this PP, and are common to other DoD PPs.

Access -- Interaction between an entity and an object that results in the flow or modification of data.

Access Control -- Security service that controls the use of resources[7] and the disclosure and modification of data.[8]

Accountability -- Property that allows activities in an IT system to be traced to the entity responsible for the activity.

Administrator -- A user who has been specifically granted the authority to manage some portion or all of the TOE and whose actions may affect the TSP. Administrators may possess special privileges that provide capabilities to override portions of the TSP.

Assurance -- A measure of confidence that the security features of an IT system are sufficient to enforce its' security policy.

[7] Hardware and software.

[8] Stored or communicated.

Asymmetric Cryptographic System -- A system involving two related transformations; one determined by a public key (the public transformation), and another determined by a private key (the private transformation) with the property that it is computationally infeasible to determine the private transformation (or the private key) from knowledge of the public transformation (and the public key).

Asymmetric Key -- The corresponding public/private key pair needed to determine the behavior of the public/private transformations that comprise an asymmetric cryptographic system.

Attack -- An intentional act attempting to violate the security policy of an IT system.

Authentication -- Security measure that verifies a claimed identity.

Authentication data -- Information used to verify a claimed identity.

Authorization -- Permission, granted by an entity authorized to do so, to perform functions and access data.

Authorized user -- An authenticated user who may, in accordance with the TSP, perform an operation.

Availability -- Timely[9], reliable access to IT resources.

Compromise -- Violation of a security policy.

Confidentiality -- A security policy pertaining to disclosure of data.

Critical Security Parameters (CSP) -- Security-related information (e.g., cryptographic keys, authentication data such as passwords and pins, and cryptographic seeds) appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a cryptographic module or the security of the information protected by the module.

Cryptographic Administrator -- An authorized user who has been granted the authority to perform cryptographic initialization and management functions. These users are expected to use this authority only in the manner prescribed by the guidance given to them.

Cryptographic boundary -- An explicitly defined contiguous perimeter that establishes the physical bounds (for hardware) or logical bounds (for software) of a cryptographic module.

Cryptographic key (key) -- A parameter used in conjunction with a cryptographic algorithm that determines [7]:

- the transformation of plaintext data into ciphertext data,

[9] According to a defined metric.

- the transformation of cipher text data into plaintext data,
- a digital signature computed from data,
- the verification of a digital signature computed from data, or
- a data authentication code computed from data.

Cryptographic Module -- The set of hardware, software, firmware, or some combination thereof that implements cryptographic logic or processes, including cryptographic algorithms, and is contained within the cryptographic boundary of the module.

Cryptographic Module Security Policy -- A precise specification of the security rules under which a cryptographic module must operate, including the rules derived from the requirements of this PP and additional rules imposed by the vendor.

Defense-in-Depth (DID) -- A security design strategy whereby layers of protection are utilized to establish an adequate security posture for an IT system.

Discretionary Access Control (DAC) -- A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. These controls are discretionary in the sense that a subject with certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.

DMZ -- A Demilitarized Zone (DMZ) is a network that is mediated by the TOE but, as a result of less stringent access controls, provides access to publicly available services, such as web servers.

Embedded Cryptographic Module -- One that is built as an integral part of a larger and more general surrounding system (i.e., one that is not easily removable from the surrounding system).

Enclave -- A collection of entities under the control of a single authority and having a homogeneous security policy. They may be logical, or may be based on physical location and proximity.

Entity -- A subject, object, user or another IT device, which interacts with TOE objects, data, or resources.

External IT entity -- Any trusted Information Technology (IT) product or system, outside of the TOE, which may, in accordance with the TSP, perform an operation.

Identity -- A representation (e.g., a string) uniquely identifying an authorized user, which can either be the full or abbreviated name of that user or a pseudonym.

Integrity -- A security policy pertaining to the corruption of data and TSF mechanisms.

Integrity label -- A security attribute that represents the integrity level of a subject or an object. The TOE uses integrity labels as the basis for mandatory integrity control decisions.

Integrity level -- The combination of a hierarchical level and an optional set of non-hierarchical categories that represent the integrity of data.

Mandatory Access Control (MAC) -- A means of restricting access to objects based on subject and object sensitivity labels.[10]

Mandatory Integrity Control (MIC) -- A means of restricting access to objects based on subject and object integrity labels.

Multilevel -- The ability to simultaneously handle (e.g., share, process) multiple levels of data, while allowing users at different sensitivity levels to access the system concurrently. The system permits each user to access only the data to which they are authorized access.

Named Object -- An object that exhibits all of the following characteristics:

- The object may be used to transfer information between subjects of differing user identities within the TSF.
- Subjects in the TOE must be able to request a specific instance of the object.
- The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user identities to request the same instance of the object.

Non-Repudiation -- A security policy pertaining to providing one or more of the following:

- To the sender of data, proof of delivery to the intended recipient,
- To the recipient of data, proof of the identity of the user who sent the data.

Object -- An entity within the TSC that contains or receives information and upon which subjects perform operations.

Operating Environment -- The total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative and personnel controls.

[10] The Bell LaPadula model is an example of Mandatory Access Control

Operating System (OS) -- An entity within the TSC that causes operations to be performed. Subjects can come in two forms: trusted and untrusted. Trusted subjects are exempt from part or all of the TOE security policies. Untrusted subjects are bound by all TOE security policies.

Operational key -- Key intended for protection of operational information or for the production or secure electrical transmissions of key streams.

Peer TOEs -- Mutually authenticated TOEs that interact to enforce a common security policy.

Public Object -- An object for which the TSF unconditionally permits all entities "read" access. Only the TSF or authorized administrators may create, delete, or modify the public objects.

Robustness -- A characterization of the strength of a security function, mechanism, service or solution, and the assurance (or confidence) that it is implemented and functioning correctly. DoD has three levels of robustness:

- **Basic:** Security services and mechanisms that equate to good commercial practices.

- **Medium:** Security services and mechanisms that provide for layering of additional safeguards above good commercial practices. ADV_INT.1

- **High:** Security services and mechanisms that provide the most stringent protection and rigorous security countermeasures.

Secure State -- Condition in which all TOE security policies are enforced.

Security attributes -- TSF data associated with subjects, objects, and users that is used for the enforcement of the TSP.

Security level -- The combination of a hierarchical classification and a set of non-hierarchical categories that represent the sensitivity on the information [10].

Sensitivity label -- A security attribute that represents the security level of an object and that describes the sensitivity (e.g. Classification) of the data in the object. Sensitivity labels are used by the TOE as the basis for mandatory access control decisions [10].

Split key -- A variable that consists of two or more components that must be combined to form the operational key variable. The combining process excludes concatenation or interleaving of component variables.

Subject -- An entity within the TSC that causes operations to be performed.

Symmetric key -- A single, secret key used for both encryption and decryption in symmetric cryptographic algorithms.

Threat -- Capabilities, intentions and attack methods of adversaries, or any circumstance or event, with the potential to violate the TOE security policy.

Threat Agent - Any human user or Information Technology (IT) product or system which may attempt to violate the TSP and perform an unauthorized operation with the TOE.

User -- Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

Vulnerability -- A weakness that can be exploited to violate the TOE security policy.

9.0 ACRONYMS

The following abbreviations from the Common Criteria are used in this Protection Profile:

AES	Advanced Encryption Standard
CC	Common Criteria for Information Technology Security Evaluation
DoD	Department of Defense
EAL	Evaluation Assurance Level
FIPS PUB	Federal Information Processing Standard Publication
GIG	Global Information Grid
I&A	Identification and Authentication
IATF	Information Assurance Technical Framework
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IT	Information Technology
MRE	Medium Robustness Environment
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
NSA	National Security Agency
PKI	Public Key Infrastructure
PP	Protection Profile
RNG	Random Number Generator
SFP	Security Function Policy
ST	Security Target
TOE	Target of Evaluation
TSE	TOE Security Environment
TSF	TOE Security Function
TSP	TOE Security Policy

10.0 REFINEMENTS

This section contains refinements where text was omitted. Omitted text is shown as bold text within parenthesis. The actual text of the functional requirements as presented in Section 5 has been retained.

FAU_ARP.1 Security alarms

FAU_ARP.1.1 – **Refinement**: The TSF shall **(take)** [immediately display an alarm message, identifying the potential security violation and make accessible the audit record contents associated with the auditable event(s) that generated the alarm, at the:

1. local console,
2. remote administrator sessions that exist, and;
3. remote administrator sessions that are initiated before the alarm has been acknowledged, and;
4. at the option of the Security Administrator, generate an audible alarm, and;
5. [assignment: other methods determined by the ST author]]

upon detection of a potential security violation.

FAU_GEN.1-NIAP-0410 Audit data generation

FAU_GEN.1.1-NIAP-0410 – **Refinement**: The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events **(for the [selection] level of audit)** listed in Table 5.3; and

c) [selection: [assignment: events at a basic level of audit introduced by the inclusion of additional SFRs determined by the ST Author], [assignment: events commensurate with a basic level of audit introduced by the inclusion of extended requirements determined by the ST Author], no additional events].

FAU_GEN.1.2-NIAP-0410 – **Refinement**: The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event (if applicable); and

b) For each audit event type **(time)**, based on the auditable event definitions of the functional components included in the PP/ST, [information specified in column three in Table 5.3].

FAU_SAA.1-NIAP-0407 Potential violation analysis

FAU_SAA.1.2-NIAP-0407 – **Refinement**: The TSF shall enforce the following rules for monitoring audited events:

a) Accumulation **(or combination)** of [

- a Security Administrator specified number of authentication failures against a single non-administrative user identifier,
- a Security Administrator specified number of consecutive failed authentication attempts,
- a Security Administrator specified number of authentication failures against an administrative user identifier;

b) Any failure of the cryptographic self-tests;

c) Any failure of the other TSF self-tests;

d) Any failure to generate a cryptographic key;

e) Detection of physical attack;

f) Any failure to decrypt a biometrics package;

g) Detection of modification of a biometrics package] **(known to indicate a potential security violation)**;

h) *[selection: [assignment: any other rules], "no additional rules"]].*

FAU_SAR.1 Audit review

FAU_SAR.1.2 – **Refinement**: The TSF shall provide the audit records in a manner suitable for the **(user)** Audit Administrator and Security Administrator to interpret the information.

FAU_SAR.2 Restricted audit review

FAU_SAR.2.1 – **Refinement**: The TSF shall prohibit all users read access to the audit records, except **(those users that have been granted explicit read-access)** the Audit Administrator and Security Administrator.

FAU_SEL.1-NIAP-0407 Selective Audit

FAU_SEL.1.1-NIAP-0407 - **Refinement**: The TSF shall **(be able)** allow only the Audit Administrator to include or exclude auditable events from the set of audited events based on the following attributes:

a) [user identifier;

b) event type;

c) success of auditable events;

d) failure of auditable events; and

e) [selection: [assignment: list of additional criteria that audit selectivity is based upon], no additional criteria]].

FAU_STG.1-NIAP-0423 Protected audit trail storage

FAU_STG.1.1-NIAP-0423 – **Refinement**: The TSF shall **(protect)** restrict the backup and deletion of stored audit records in the audit trail to the Audit Administrator.

FAU_STG.1.2-NIAP-0423 - **Refinement**: The TSF shall **(be able to)** prevent **(unauthorised)** modifications to the audit records in the audit trail.

FAU_STG.3 Action in case of possible audit data loss

FAU_STG.3.1 - **Refinement**: The TSF shall **(take)** [generate an alarm by [assignment: method determined by the ST Author to generate the alarm]], if the audit trail exceeds [an Audit Administrator settable percentage of storage capacity].

FAU_STG.NIAP-0414-1-NIAP-0429 Site-Configurable Prevention of Audit Loss

FAU_STG.NIAP-0414-1.1-NIAP-0429 - **Refinement**: The TSF shall provide the Audit Administrator the capability to select one **(or more)** of the following actions: prevent auditable events, except those taken by the Audit Administrator, overwrite the oldest stored audit records or [selection: [assignment: other actions to be taken in case of audit storage failure], no other actions] **(and)** to be taken if the audit trail is full.

FAU_STG.NIAP-0414-1.2-NIAP-0429 **Refinement**: The TSF shall as a default [prevent auditable events, except those taken by the Audit Administrator] if the audit trail is full **(and no other action has been selected)**.

FIA_AFL.1-NIAP-0425(1) Authentication failure handling (Against a single non-administrative user identifier)

FIA_AFL.1.2-NIAP-0425(1) - **Refinement**: When the defined number of consecutive unsuccessful authentication attempts has been met **(or surpassed)**, the TSF shall [ignore any further authentication attempts related to that user until the Security Administrator defined time period for non-administrative users has elapsed, or an action is taken by the Security Administrator].

FIA_AFL.1-NIAP-0425(2) Authentication failure handling (Consecutive failed attempts)

FIA_AFL.1.2-NIAP-0425(2) - **Refinement**: When the defined number of consecutive unsuccessful authentication attempts has been met **(or surpassed)**, the TSF shall [ignore any further authentication attempts related to that user until the Security Administrator defined time period for non-administrative users has elapsed, or an action is taken by the Security Administrator].

FIA_AFL.1-NIAP-0425(3) Authentication failure handling (Administrator Users)

FIA_AFL.1.2-NIAP-0425(3) **– Refinement**: When the defined number of consecutive unsuccessful authentication attempts has been met **(or surpassed)**, the TSF shall [ignore any further authentication attempts related to that user until the Security Administrator defined time period for non-administrative users has elapsed, or an action is taken by the Security Administrator].

FIA_ATD.1 User attribute definition

FIA_ATD.1.1 – **Refinement**: The TSF shall maintain the following list of security attributes belonging to **(individual)** administrative users:

- [trusted user identifier,
- role(s), and
- [selection: [assignment: any other security attributes defined by the ST Author], none.]]

and restrict the ability to assign and modify these security attributes to the Security Administrator.

FIA_UAU.5 Multiple authentication mechanisms

FIA_UAU.5.1 **Refinement**: The TSF shall provide [a biometric authentication mechanism, [assignment: non-biometric authentication mechanism that meets the strength of secrets metric defined in FIA_SOS.1], [selection: [assignment: any other authentication mechanisms defined by the ST Author], none.]] to **(support)** perform user authentication.

FIA_UAU.7 Protected authentication feedback

FIA_UAU.7.1 – **Refinement**: The TSF shall provide only [instructional information] to aid the user **(while the authentication is in progress)** in supplying their biometric characteristic to the TOE.

FPT_ITT.1(2) Basic internal TSF data transfer protection (from undetected modification)

FPT_ITT.1.1(2) **Refinement**: The TSF shall **(protect)** use a cryptographic digital signature to detect modification of TSF data **(from [selection])** when it is transmitted between separate parts of the TOE.

FPT_PHP.3 Resistance to physical attack

FPT_PHP.3.1 **Refinement**: The TSF shall **(resist)** react [to the exposure of internal components] **(to)** of the [biometrics TOE] by responding automatically such that the TSP is not violated.

FTA_SSL.3 TSF-initiated termination

FTA_SSL.3.1 - **Refinement**: The TSF shall terminate **(a user)** an administrative session after a [Security Administrator-configurable time interval of session inactivity].

FTA_TAB.1 Default TOE access banners

FTA_TAB.1.1 - Refinement**: Before establishing** (a user) **an administrative session, the TSF shall display an advisory notice and consent warning message regarding unauthorized use of the TOE.**

[i] A deletion of CC text was performed in FPT_TST.1.1(1). Rationale: The word "TSF" was deleted to allow for the demonstration of the correct operation of a number of cryptographic related self tests.

FPT_TST.1.1(1) **Refinement:** The TSF shall run a suite of self-**tests in accordance with FIPS PUB 140-2, Level 4 (as identified in Table 5.3)** during initial start-up **(on power on)**, at the request of the **cryptographic administrator (on demand)**, **under various conditions,** and periodically **(at least once a day)** to demonstrate the correct operation of the ~~TSF~~ **following …**

[ii] A deletion of CC text was performed in FPT_TST.1.2(2). Rationale: The word "users" was deleted to replace it with the role of " cryptographic administrator". "Only authorized cryptographic administrators should be given the capability to verify the integrity of cryptographically related TSF data.

FPT_TST.1.2(1) **Refinement:** The TSF shall provide authorized ~~users~~ **cryptographic administrators** with the capability to verify the integrity of **TSF data related to the cryptography by using TSF-provided cryptographic functions.**.

[iii] A deletion of CC text was performed in FPT_TST.1.3(1). Rationale: The word “users” was deleted to replace it with the role of " cryptographic administrator". Only authorized cryptographic administrators should be given the capability to verify the integrity of cryptographically related TSF executable code.

FPT_TST.1.3(1) **Refinement:** The TSF shall provide authorized ~~users~~ **cryptographic administrators** with the capability to verify the integrity of stored **cryptographically related** TSF executable code.

[iv] A deletion of CC text was performed in FPT_TST.1.1(2). Rationale: The words "the TSF" was deleted to allow for the demonstration of the correct operation of each key generation component. The word “perform” replaced “run a suite of” for clarity and better flow of the requirement.

FPT_TST.1.1(2) **Refinement:** The TSF shall ~~run a suite of~~ **perform** self-tests **immediately after generation of a key** to demonstrate the correct operation of ~~the TSF~~ **each key generation component**. **If any of these tests fails, that generated key shall not be used, the cryptographic module shall react as required by FIPS PUB 140 for failing a self-test, and this event will be audited.**

[v] A deletion of CC text was performed in FPT_TST.1.2(2). Rationale: The word "users" was deleted to replace it with the role of "cryptographic administrator".

FPT_TST.1.2(2) **Refinement:** The TSF shall provide authorized ~~users~~ **cryptographic administrators** with the capability to verify the integrity of TSF data **related to the key generation**.

[vi] A deletion of CC text was performed in FPT_TST.1.3(2). Rationale: The word “users” was deleted to replace it with the role of "cryptographic administrator".

FPT_TST.1.3(2) **Refinement:** The TSF shall provide authorized ~~users~~ **cryptographic administrators** with the capability to verify the integrity of stored TSF executable code **related to the key generation**.