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

Information Assurance Directorate

Version 1.1

July 25, 2007

Protection Profile Title:

Biometric Verification Mode Protection Profile for Basic Robustness Environments.

Criteria Version:

This Protection Profile "U.S. Government Biometric Verification Mode Protection Profile for Basic 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.

Constraints:

Targets of Evaluation (TOEs) developed to satisfy this Protection Profile shall conform to CC Part 2 and CC Part 3 and applicable interpretations.

Table of Contents

1.0 INTRODUCTION

This Biometric Verification Mode Protection Profile (PP) for Basic Robustness Environments was sponsored by the Biometrics Management Office (BMO) and the National Security Agency (NSA). 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: Biometric Verification Mode Protection Profile (PP) for Basic Robustness Environments

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

CC Version: Common Criteria (CC) Version 3.1, and applicable interpretations (NIAP as well as internationally approved).

Registration: <to be provided upon registration>

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

Keywords: Protection Profile, Basic Robustness Environments, verification mode, 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 basic 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 levies requirements on the IT environment that are necessary to address critical functionality that must be provided by the IT environment. In some instances the TOE only partially addresses a threat, and relies on the IT environment to completely play a role in addressing a threat. One critical aspect in these IT environment requirements is the protection of the biometrics package (i.e., trusted user identifier, user's reference template(s), and possibly other information). Contrary to the medium robustness biometrics TOE, there is no protection afforded to the biometrics package by the TOE. The acceptable degree of protection (e.g., encryption, access control provided by a database or operating system) provided by the IT environment is a determination that is made by the end-users of the TOE. It is important for integrators and certifiers to ensure that the IT environment satisfies these IT environment requirements, since they are necessary for the TOE to enforce its security policies.

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

1.3 Related Protection Profiles

A medium robustness PP for a biometric TOE operating in verification mode has many of the same functional requirements, and adds additional functionality, including the use of cryptography to protect the biometric packages. Contrary to a basic robustness TOE, the medium robustness TOE has no reliance on the IT environment in order to address some of the threats and to enforce its security policies. The medium robustness PP also has more 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 only operates in 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.

- *U.S. Government Biometric Verification Mode Protection Profile For Medium Robustness Environments.*

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 2.1.4 of Part 2 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 "(EXT)" 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.

2.0 TOE DESCRIPTION

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

Biometric TOEs are unlike many 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) do so in a very limited fashion. 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 sample 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 sample, 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 unlike the TOE defined for medium robustness environments, the TOE for basic robustness environments excludes some security relevant functionality (e.g., audit storage, audit review) and may rely on another IT entity to provide logical protection to components of the TOE (e.g., an underlying OS may provide protection from tampering of software components of the TOE). This means that the comparison software or any capture controller function could execute on an IT product other than the TOE. Figure 1 illustrates an example of a TOE that is integrated into an IT product. 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 storage, comparator function, and any other TOE software resides in the IT product. The capture device transmits the live sample, and possibly other data (e.g., unique device id), to the comparator through a path that is not trusted with respect to the TOE. There is a reliance on the environment to protect this communication path (e.g., physical protection of the communication line, encryption). The comparator retrieves the reference template from storage (in Figure 1, the storage is depicted as residing in the IT product, but the storage could be located elsewhere), which is also protected by the environment. The reference template is included in the biometric package. The comparator compares the templates and generates a match/no match decision, which is then provided to the IT product.

When the TOE is physically separated, the environment is required to maintain confidentiality and to detect modification of the transmitted data. This could be achieved by physically protecting the communication lines, or some form of logical protection (e.g., encryption).

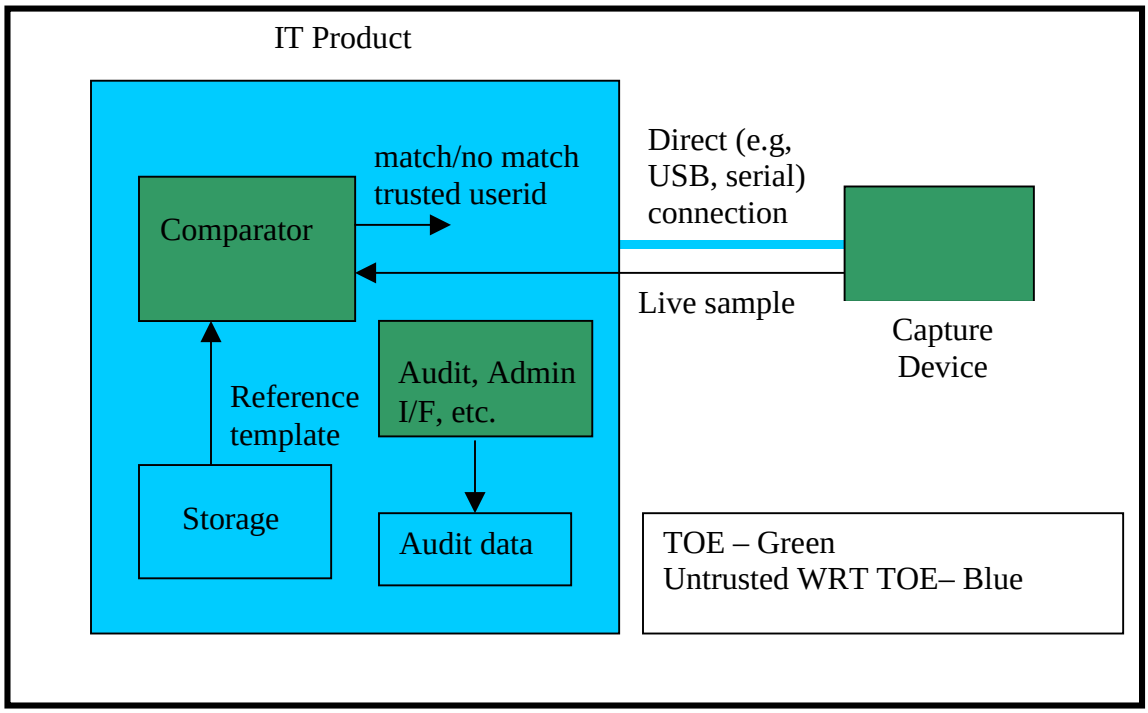


Figure 1. Example of TOE architecture with reliance on the IT environment for protection.

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 authentication (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. The following is a description of each block in the diagram:

- *Capture* –In capture, a sample of the user's biometric is acquired using the required sensor (camera, microphone, fingerprint scanner, etc.).
- *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 reference template. During the authentication process, it is known as the live sample.
- *Package Creation* – Performed only during enrollment. The TOE binds the user's identity and additional information with the biometric template to create a biometric package for storage. It is left to the IT environment to ensure that this binding can be trusted (e.g., protect the storage from unauthorized modification).
- *Comparison* – Performed only during authentication. Matches the live sample and reference template(s). 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 include setting of the threshold, and determining audit events. The ability to review audit information is levied on the IT environment.

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

The fundamental 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 reference template, and associates this template with the enrollee's identity for storage.

During authentication, the biometric device can be used for identification or verification of the person's identity. In identification mode, the biometric device 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 mode, 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 PP considers a biometric TOE operating only in the verification mode.

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

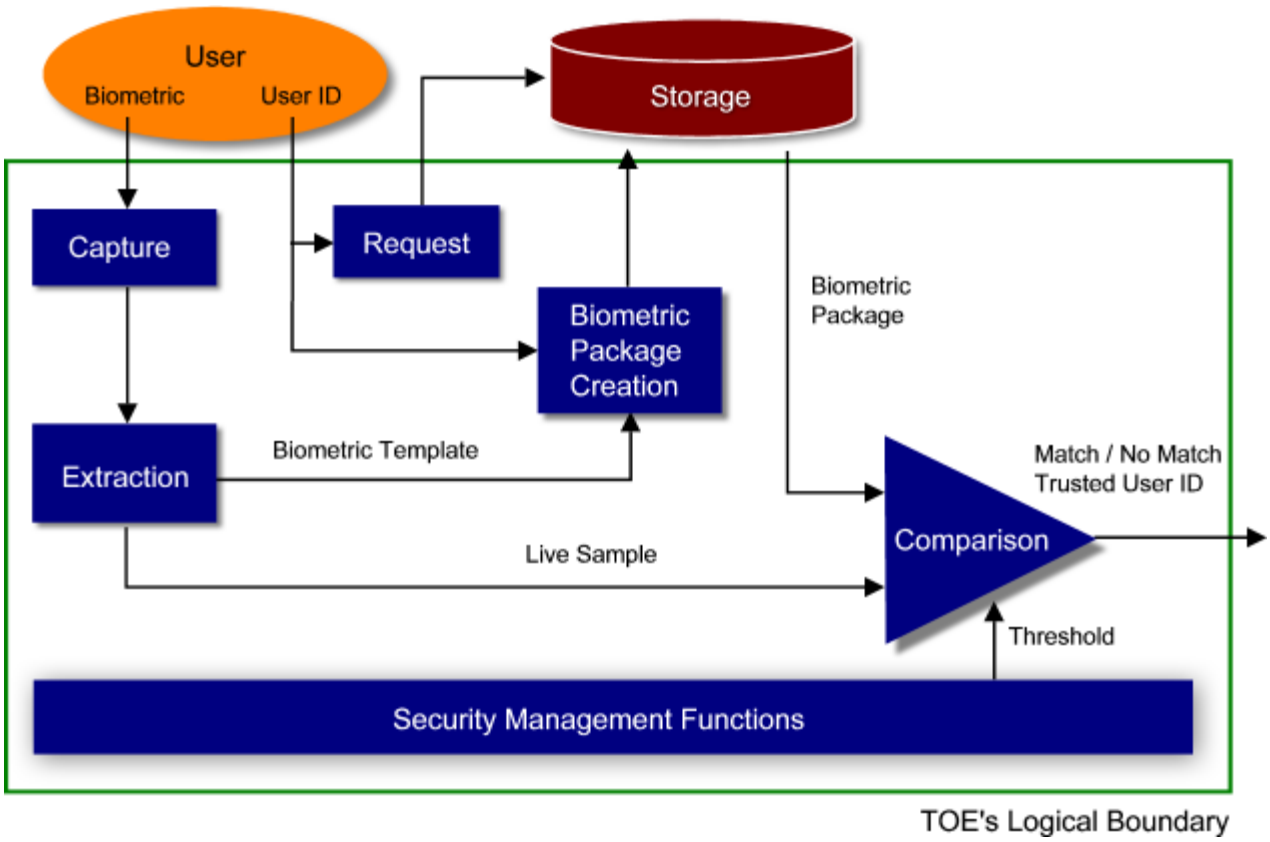


Figure 2. TOE functional block diagram

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 an administrator can enroll users in a biometric TOE. The TOE's administrative guidance provides administrators guidance about acceptable quality metrics in regards to the quality of the biometric template.

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, it is left to the environment to ensure confidentiality of the biometric package is maintained, and to detect modification of the package while in storage or in transit.

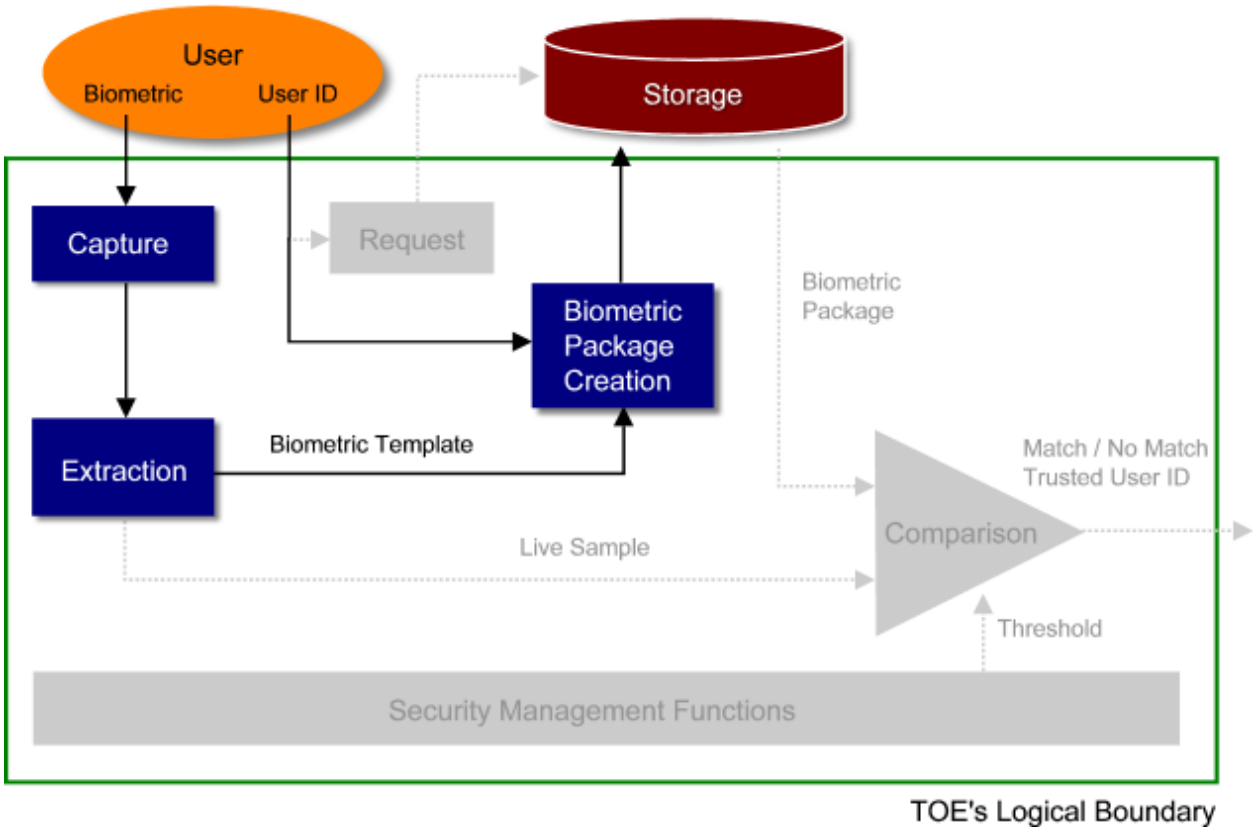


Figure 3. Block diagram of the enrollment process.

2.1.2 The Verification Process

Figure 4 highlights the components of a TOE involved during the verification process. This verification process essentially defines the mode of biometric authentication, which in the case of this TOE is verification mode. The TOE retrieves the biometric package of the user's claimed identity from storage.

The biometric template(s) in the biometric package is then matched against a live sample captured from the user and a match/no-match result is generated. The 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.

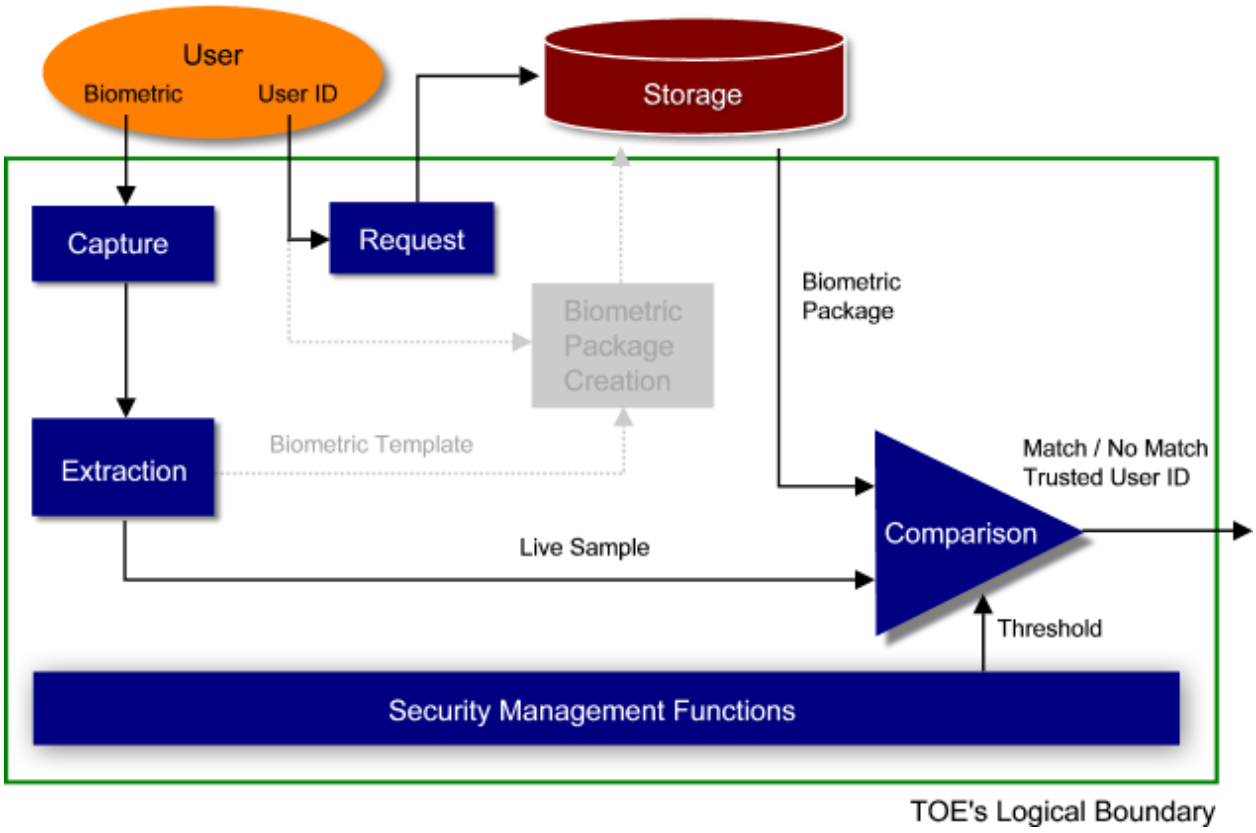


Figure 4. Verification process.

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 the next section, 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 firewall may have "low value" data itself, but it might protect an enclave with high value data. If the firewall 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 (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 was authorized to certain data on a TOE, yet they connect 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.

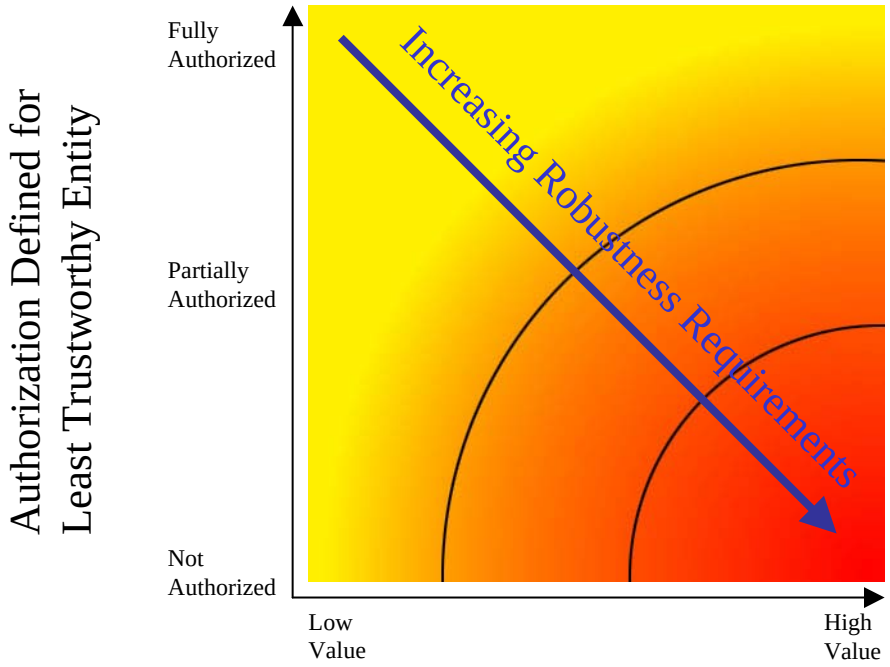
It is critical to note that several combinations of the environmental factors will result in environments in which the likelihood of an attempted security policy compromise is similar. Consider the following two cases:

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.

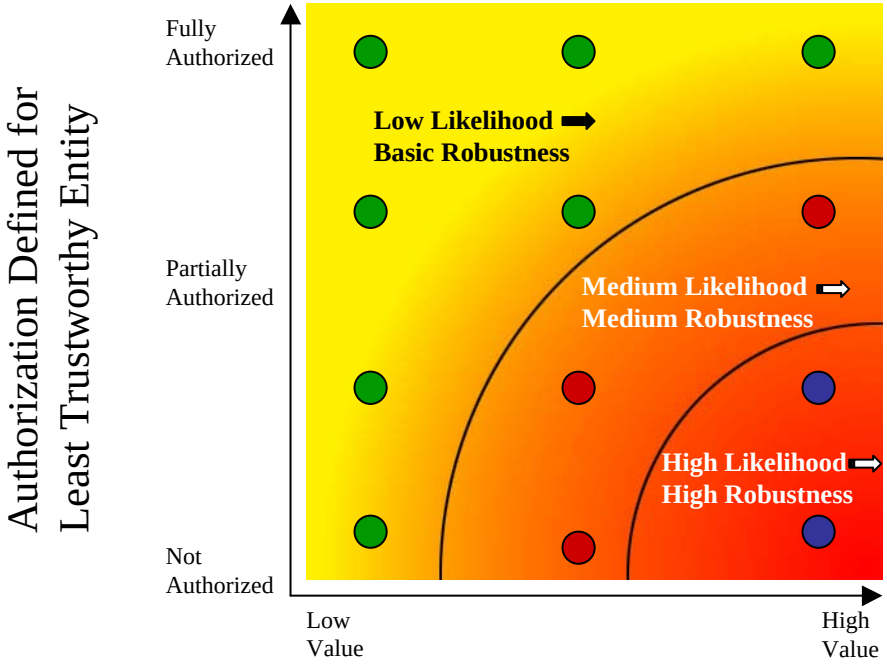
The preceding examples demonstrated that it is possible for radically different combinations of entity authorization/resource values to result in a similar likelihood of an attempted compromise. As mentioned earlier, the robustness of a system is an indication of the protection being provided to counter 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 not 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 3.5 of this PP, the targeted threat level for a basic 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 basic robustness PP are appropriate for their intended application of a compliant biometric authentication device.

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 x 10-5) for this TOE is weaker than the probability that a password can be guessed (1 x 10-6 for the non-biometric authentication mechanism in this PP). But it may be much more difficult to prepare and present 105 different biometric samples than it is to enter 106 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.

3.5 Assumptions

The specific conditions below are assumed to exist in a PP-compliant TOE environment. Typically, assumptions are not used to specify expected behavior of IT devices if IT environment requirements can be used to express the required behavior. However, in this instance A.BIOMETRIC_PACKAGE_PROTECT is used to indicate a need for the IT environment to protect the biometrics package since there are a number of ways in which the package could be protected (e.g., access control mechanisms provided by an operating system or database management system, encryption of the package). These requirements could have been expressed using CC requirements such as FDP_ACC, FDP_ACF, FCS_COP, FPT_ITC, FPT_ITI but the PP authors wanted to allow product developers flexibility in their implementation and end-users flexibility is integrating the TOE into their system. Thus, the suitability of the protection afforded by the combination of the TOE, the IT environment and any procedural control is left as an exercise to the accreditation authority. This was determined to be an acceptable approach given the level of assurance provided by the TOE.

A.COMM_PROTECT	The communication paths between physically separate parts of the TOE and between the TOE and environment (IT and non-IT) are protected (e.g., physically, encrypted).
A.BIOMETRICS_PACKAGE_PROTECT	The biometrics package (i.e., reference template, and its binding to a user identifier) is protected

	from disclosure and modification while in storage and during transmission between the IT environment and the TOE.
A.ENROLLMENT_APPROVAL	It is assumed that sites follow appropriate procedures for validating the identity of enrolled individuals.
A.NO_EVIL	Administrators are non-hostile, appropriately trained and follow all administrator guidance.
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.[1]
A.OPERATING_RANGE	The TOE is placed in an environment that does not exceed its normal operating range 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.

---

[1] The TOE can reside on or be integrated into an IT product that has general purpose computing capabilities. In fact, it is expected. This assumption merely states that the TOE itself does not offer this type of capability.

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:

- The motivation for the threat agent defines the upper bound with respect to the level of robustness required for the TOE.
- A threat agent's expertise and/or resources that is "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).
- 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 Medium 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 interpreted with the accompanying rationale provided in Section 6.1; 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. ACCIDENTAL_ADMIN_ERROR	An administrator may mistakenly incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.BYPASS	An attacker may bypass any component of the biometric product and gain unauthorized authentication.
T.ARTIFACT	An attacker may use an 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.POOR_DESIGN	Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_IMPLEMENTATION	Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous 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 to gain unauthorized access.

T.RESIDUAL_DATA	Residual biometric authentication data from a previous valid user if not cleared may allow an attacker 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.TSF_ COMPROMISE	A 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, the security state of the TOE may 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.

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.ADMIN_GUIDANCE	The TOE will provide administrators with the necessary information for secure delivery and management.
O.ADMIN_ROLE	The TOE will provide an administrator role to isolate administrative actions from untrusted user actions.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security-relevant events associated with users.
O.ALARM_GENERATION	The TOE will provide the capability to detect and alert an administrator of a potential security violation.
O.AUTHENTICATION	The TOE will provide a biometric authentication mechanism to authenticate users for the IT environment or non-IT environment.
O.CONFIGURATION_IDENTIFICATION	The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly,
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.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE.
O.DOCUMENTED_DESIGN	The design of the TOE is adequately and accurately documented.
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.PARTIAL_FUNCTIONAL_TESTING	The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements.
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.PARTIAL_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, through its own interfaces.
O.TOE_ACCESS	The TOE will provide mechanisms that control an administrator's logical access to the TOE.
O.VULNERABILITY_ANALYSIS	The TOE will undergo some vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.

4.2 Security Objectives for the Operating Environment

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

OE.AUDIT_TRAIL_REVIEW	The capability to selectively view audit information generated by the TOE is provided by the IT environment.
OE.AUDIT_PROTECTION	The IT Environment protects the audit information generated by the TOE from modification, disclosure and loss.
OE.BIOMETRICS_PACKAGE_PROTECT	The biometrics package (i.e., reference template, and its binding to a user identifier) is protected from disclosure and modification while in storage

	and during transmission between the IT environment and the TOE.
OE.COMM_PROTECT	The communication paths between physically separate parts of the TOE and between the TOE and environment (IT and non-IT) are protected (e.g., physically, encrypted).
OE.ENROLLMENT_APPROVAL	Sites follow appropriate procedures for validating the identity of enrolled individuals.
OE.NO_EVIL	Administrators are non-hostile, appropriately trained and follow all administrator guidance.
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.NON_BYPASS	The IT environment shall ensure that the TOE cannot be bypassed and is always invoked, unless otherwise directed by an administrator (e.g., fallback procedures for users unable to use the TOE) to perform user authentication.
OE.TOE_PROTECT	The IT environment shall protect the TOE's executable code from tampering.
OE.TIME_STAMPS	The IT environment shall provide reliable time stamps and the capability for the administrator to set the time used for these time stamps.
OE.OPERATING_RANGE	The TOE is placed in an environment that does not exceed its normal operating range as defined by the vendor.

5.0 IT SECURITY REQUIREMENTS

The security requirements that are levied on the TOE and the IT environment are specified in this section of the PP. An ST Author addresses the requirements levied on the TOE, and ensures the TOE interacts with an instantiation of the IT environment that satisfies the IT environment requirements. An ST may include the IT environment requirements in their TOE requirements if they desire.

5.1 TOE Security Functional Requirements

This section provides functional and assurance requirements that must be satisfied by a PP-compliant TOE. These requirements consist of functional components from Part 2, NIAP interpretations, extended functional requirements and assurance components from Part 3 of the CC. 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, and Table 5.3 identifies the functional requirements that the TOE relies on the IT environment to support in order for the TOE to enforce its security policies.

Table 5.1 - Security Functional Requirements

Functional Components (from CC Part 2 and NIAP Interpretations)	
FAU_ARP.1	Security alarms
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
FDP_RIP.2	Full residual information protection
FIA_AFL.1(1)	Authentication failure handling (Against a single non-administrative user identifier)
FIA_AFL.1(2)	Authentication failure handling (Consecutive failed attempts)
FIA_AFL.1(3)	Authentication failure handling (Administrator Users)
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

Functional Components (from CC Part 2 and NIAP Interpretations)	
FIA_UAU.5	Multiple authentication mechanisms
FIA_UAU.7	Protected authentication feedback
FIA_UID.2	User identification before any action
FIA_USB.1	User-subject binding
FMT_MOF.1(1)	Management of security functions behavior (Audit)
FMT_MOF.1(2)	Management of security functions behavior (Alarms)
FMT_MOF.1(3)	Management of security functions behavior (Self-test)
FMT_MOF.1(4)	Management of security functions behavior (Maintenance Mode)
FMT_MOF.1(5)	Management of security functions behavior (Enrollment)
FMT_MOF.1(6)	Management of security functions behavior (non-biometric Authentication Mechanism)
FMT_MOF.1(7)	Management of security functions behavior (Biometric Authentication Mechanism)
FMT_MTD.1	Management of TSF data (Authentication Mechanism Data)
FMT_REV.1	Revocation
FMT_SMR.1	Security roles
FPT_RCV.2	Recovery from Failure
FPT_TST.1	TSF Testing
FTA_SSL.3	TSF-initiated termination
FTA_TAB.1	Default TOE access banners

Table 5.2 - Extended Security Functional Requirements

Extended Functional Components	
FIA_ENROLL_(EXT).1	Enrollment
FPT_PHP_(EXT).1	Detection of physical attack

Table 5.3 – IT Environment Security Functional Requirements

IT Environment Functional Components (from CC Part 2 and NIAP Interpretations)	
FAU_SAR.1	Audit review
FAU_SAR.2	Restricted audit review
FAU_SAR.3	Selectable audit review
FAU_STG.1	Protected audit trail storage
FAU_STG.3	Action in case of possible audit data loss
FPT_STM.1	Reliable time stamps

5.1.1 Security Audit Requirements (FAU)

FAU_ARP.1 Security alarms

Hierarchical to: No other components.

Dependencies: FAU_SAA.1 Potential violation analysis

FAU_ARP.1.1 – **Refinement:** The TSF shall take **the following action**:[generate an alarm condition to the IT environment by [assignment: method determined by the ST Author to generate the alarm]] upon detection of a potential security violation.

Application Note: The TOE generates a signal indicating an alarm condition to the environment by a method determined by the ST Author. Acceptable methods may include sending an interrupt or message to the IT environment. The TOE could satisfy this requirement by indicating an alarm without interaction with the environment (e.g., an LED or audible indication that indicates an alarm condition. 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).

FAU_GEN.1-NIAP-0410 Audit data generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

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 5.4**; 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 5.4 -- 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 identity of the administrator performing the function
FAU_SEL.1-NIAP-0407	Attempts to modify the audit configuration	The identity of the administrator performing the function
FDP_RIP.2	None	

Requirement	Auditable Events	Additional Audit Record Contents
FIA_AFL.1(1)	The reaching of the threshold for the unsuccessful authentication attempts The actions (e.g. disabling of an account, timeout) taken The subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of an account)	Identity of the unsuccessfully authenticated user; Identity of the administrator (if applicable) that took action to re-enable an account; Period of timeout (if applicable)
FIA_AFL.1(2)	The reaching of the threshold for the unsuccessful authentication attempts The actions (e.g. disabling of an account, timeout) taken The subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of an account)	Identity of the unsuccessfully authenticated users; Identity of the administrator (if applicable) that took action to re-enable an account; Period of timeout (if applicable)
FIA_AFL.1(3)	The reaching of the threshold for the unsuccessful authentication attempts The actions (e.g. disabling of an account, timeout) taken The subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of an account)	Identity of the unsuccessfully authenticated administrator; Identity of the 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.	
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; Comparison score of a non-match decision; Claimed identity of the administrator attempting to authenticate using the non-biometric authentication mechanism (if applicable);
FIA_UAU.7	None.	

Requirement	Auditable Events	Additional Audit Record Contents
FIA_UID.2	All use of the user identification mechanism, including the user identity provided	
FIA_USB.1	Success and failure of binding of user security attributes to a subject	The identity of the user whose attributes are attempting to be bound
FMT_MOF.1(1)	All attempts to enable, disable, determine, or modify the behavior of the audit generation functions in the TSF	The identity of the administrator performing the function
FMT_MOF.1(2)	All attempts to modify the behavior of the alarm and analysis functions in the TSF	The identity of the administrator performing the function
FMT_MOF.1(3)	All attempts to invoke and modify the behavior of the self-tests functions in the TSF	The identity of the administrator performing the function
FMT_MOF.1(4)	None	
FMT_MOF.1(5)	All attempts to determine, or modify the behavior of the enrollment functions in the TSF	The identity of the administrator performing the function
FMT_MOF.1(6)	All attempts to enable and disable the non-biometric authentication mechanism	The identity of the administrator performing the function
FMT_MOF.1(7)	All attempts to modify or determine the behavior of the biometric authentication mechanism	The identity of the administrator performing the function
FMT_MTD.1	All attempts to query and set the authentication mechanism data	The 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 identity of the administrator performing the function
FMT_SMR.1	All attempts to modify the group of users that are associated with a role	User identifiers that are associated with the modifications The identity of the administrator performing the function

Requirement	Auditable Events	Additional Audit Record Contents
FPT_RCV.2	The fact that a failure or service discontinuity occurred; Resumption of the regular operation;	Type of failure or service discontinuity
FTA_SSL.3	The termination of a remote session by the session locking mechanism	The identity of the administrator associated with the session that was terminated
FTA_TAB.1	None	
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;	Identity of the administrator attempting to create/modify a reference template; The enrolled user's user identifier.
FPT_PHP_(EXT).1	Detection of physical attack	
FPT_TST.1	Any failure of self-tests, including detection of corrupted TSF data or software	Self-test that failed; The affected TSF components

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 5.4*].

Application Note: A subject identity is distinct from a user identifier. 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 administrator valuable information.

FAU_GEN.2-NIAP-410 User Identity Association

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FIA_UID.1 Timing of identification

FAU_GEN.2.1-NIAP-410 - 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 user identifier may not be associated with a biometrics package (e.g., an invalid user identifier was presented), however, the supplied 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 user identifier that is supplied when a user attempts to authenticate. 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 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

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_SAA.1.1-NIAP-0407 – The TSF shall be able to apply a set of rules in monitoring the audited 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 audited events:

a) Accumulation of [

- An administrator specified number of authentication failures against a single non-administrative user identifier,
- An administrator specified number of consecutive failed authentication attempts,
- An administrator specified number of authentication failures against an administrative user identifier];

b) [Any failure of the TSF self-tests

c) Any detection of physical tampering;

d) *[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. An 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(1) – (3).

The failure of TSF self-tests in (c) include failures of FPT_TST.1.

FAU_SEL.1-NIAP-0407 Selective Audit

Hierarchical to:	No other components.
Dependencies:	FAU_GEN.1 Audit data generation
	FMT_MTD.1 Management of TSF data

FAU_SEL.1.1-NIAP-0407 - Refinement: The TSF shall allow only the 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 security events;*

d) failure of auditable security 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 administrator has the capability to "pre-select" audit events, this does not mean that this capability implicitly disables alarm events (FAU_SAA.1). If the administrator de-selects an audit event that is listed in FAU_SAA.1 that event will still generate an alarm if an administrator has enabled that event(s) to generate an alarm.

5.1.2 User Data Protection (FDP)

FDP_RIP.2 Full residual information protection

Hierarchical to: FDP_RIP.1 Subset residual information protection

Dependencies: No dependencies.

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 of the completion of a function, is that once the TSF 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.

5.1.3 Identification and Authentication (FIA)

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

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1(1) - Refinement: The TSF shall detect when *an administrator configurable positive integer within* [*a range from 1 to 3*] 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(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 an administrator defined time period for non-administrative users has elapsed, or an action is taken by an administrator].

Application Note: The intent of these requirements is to allow an administrator to set the number of unsuccessful authentication attempts that are associated with a user identifier that is ***not*** *associated with an administrative role. An administrator also has the option of configuring the TOE so further authentication attempts associated with the user identifier are ignored until an administrator takes an action (e.g., re-enables the account) or to ignore further authentication attempts associated with the user identifier until an administrator configured time period for non-administrative users has elapsed (e.g., the TOE will not authenticate a user associated with that non-administrative 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 an administrator configured setting, then an alarm is generated as required by FAU_SAA.1.*

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

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1(2) - The TSF shall detect when *an administrator configurable positive integer within* [*a range from 1 to 3*] of unsuccessful authentication attempts occur related to [consecutive failed biometric authentication attempts].

FIA_AFL.1.2(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 Administrator defined time period for consecutive failed authentication attempts has elapsed, or an action is taken by the Administrator].

Application Note: The intent of this requirement is to provide an administrator the capability to set the number of consecutive failed authentication attempts, regardless of the user identifier. This configurable number is different than that specified in FIA_AFL.1. For example, an administrator may decide to set the failed number of authentication attempts against a non-administrative user identifier to be three, and may

set the failed number of consecutive failed authentication attempts to six. An administrator defined time period is also distinct from the non-administrative user defined period defined in FIA_AFL.1(1). For example, an 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 an administrator configured setting, then an alarm is generated as required by FAU_SAA.1.

FIA_AFL.1(3) Authentication failure handling (Administrator Users)

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1(3) - The TSF shall detect when *an administrator configurable positive integer within* [*a range from 1 to 3*] of unsuccessful authentication attempts occur related to [an administrators use of any of the authentication mechanisms].

FIA_AFL.1.2(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 an administrator defined time period for administrative users has elapsed, or an action is taken by an administrator].

Application Note: This iteration of FIA_AFL.1 applies to user identifiers associated with an administrative role. The Administrator configurable number is distinct from the configurable number specified in the previous two iterations, as is the 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 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

Hierarchical to: No other components.

Dependencies: No dependencies.

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

- [user identifier,
- [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 Administrator.

Application Note: The TOE only associates security attributes with administrative users. Untrusted users do not have any interaction with the TOE that requires the association of

security attributes.

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:
   - [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, while it is under the control of the TOE;[2]
c) Enrollment (e.g., initial, refreshing reference templates, adding additional reference templates[3]) is performed by the 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 administrator is provided a quality metric of the newly created reference template;
f) [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 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 while it is in the TOE's scope of control. The IT environment must ensure the reference template is not modified once it leaves the TOE's scope of control.

Item (d) requires that the 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

[2] 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. Once the reference template leaves the TOE's scope of control the environment is responsible for protecting the reference template from modification.

[3] 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).

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 administrator what are acceptable quality metrics. This allows the 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.

FIA_SOS.1 Verification of secrets

Hierarchical to: No other components.

Dependencies: No dependencies.

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 1×10^6].

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

Hierarchical to: No other components.

Dependencies: No dependencies.

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

a) For each attempt to use the biometric authentication mechanism, the False Acceptance Rate shall be in an administrator settable range with a minimum value of: [assignment: rate assigned by ST Author] to a maximum value of: 1 in 10,000, and
b) False Rejection Rate shall be in an 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 10,000. 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 10,000 it is

acceptable for the ST author to use the rate 1 in 10,000 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 better 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

Hierarchical to:	FIA_UAU.1 Timing of authentication
Dependencies:	FIA_UID.1 Timing of identification

FIA_UAU.2.1 – Refinement: The TSF shall require the administrators to be successfully authenticated before allowing any other TSF-mediated actions on behalf of the administrator.

Application Notes: This requirement applies to 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

Hierarchical to:	No other components.
Dependencies:	No dependencies.

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). It should be noted that a PIN by itself does not constitute an authentication mechanism. If a product uses a PIN as an identifier, that PIN cannot be consider authentication data. In order to qualify as an authentication mechanism, the mechanism must require the user to provide an identifier, as well as some form of authentication data. The non-biometric authentication mechanism is to be used, at the option of an 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 [selection: the IT environment with the user identifier and a match/non-match decision, the non-IT environment with a match/no match decision] according to the following rules:
    a) 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;
    b) at the option of the administrator, the TOE will not successfully authenticate the same user identifier consecutively in a time duration specified by the administrator;
    c) [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.

*For item (b), the administrator has the ability to configure the TOE to prevent the same user from successfully authenticating consecutively at the same capture device in an administrator defined period of time. For example, the 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 administrator can choose that these users require authentication only by the biometric authentication mechanism(s), only by the non-biometric authentication mechanism as required in UAU.5.1, 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:

- 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;
- at the option of the administrator, the TOE will not successfully authenticate the same user identifier consecutively in a time duration specified by the administrator;

a) [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 after a failed authentication attempts, such that there can be no more than a 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.

FIA_UAU.7 Protected authentication feedback

Hierarchical to:	No other components.
Dependencies:	FIA_UAU.1 Timing of authentication

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 that might help the attacker to fool the device in subsequent authentication attempts. Additionally the biometric system must not inform the user if the claimed user identifier could not be found, so as to aid an attacker guessing user identifiers. Instructional information includes positioning information, volume, etc.

FIA_UID.2 User identification before any action

Hierarchical to: FIA_UID.1 Timing of identification

Dependencies: No dependencies.

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

Application Note: This requirement ensures that users are required to identify themselves before the TOE will perform authentication.

FIA_USB.1 User-subject binding

Hierarchical to: No other components.

Dependencies: FIA_ATD.1 User attribute definition

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

FIA_USB.1.2: The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [assignment: rules for the initial association of attributes].

FIA_USB.1.3: The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [assignment: rules for the changing of attributes].

5.1.4 Security Management Requirements (FMT)

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

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

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

- [Security Audit (FAU_SEL)]

to [an 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 (alarms)

FMT_MOF.1.1(2) - 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 [an Administrator].

Application Note: This requirement ensures only an 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. The ST author describes how the administrator is alerted by the TOE in FAU_ARP.1 (e.g., notify the administrator via a pager) and the ST author should consider how "modify the behavior" applies to that functionality.

FMT_MOF.1(3) - Management of security functions behavior (Self-test)

FMT_MOF.1.1(3) – **Refinement:** The TSF shall restrict the ability to **invoke**, *modify the behavior of* the functions:

- [TSF Self-Test (FPT_TST.1)]

to [the 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. "Invoke" refers to running the self-tests.

FMT_MOF.1(4) Management of security functions behavior (Maintenance Mode)

FMT_MOF.1.1(4) - The TSF shall restrict the ability to *enable* the functions [to restore the TOE to a secure state from maintenance mode (FPT_RCV.1.1)] to the [Administrator].

FMT_MOF.1(5) Management of security functions behavior (Enrollment)

FMT_MOF.1.1(5) - **Refinement:** The TSF shall restrict the ability to **perform**, ***determine* and** *modify the behavior of* the function [enrollment (FIA_ENROLL_(EXT).1)] to [the Administrator].

Application Notes: The Administrator is the only user that is allowed to perform the enrollment function. "Determine the behavior" refers to the ability of the 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 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(6) Management of security functions behavior (non-biometric Authentication Mechanism)

FMT_MOF.1.1(6) - The TSF shall restrict the ability to *enable and disable* the functions:[

- non-biometric authentication mechanism]

to the [Administrator].

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

FMT_MOF.1(7) Management of security functions behavior (Biometric Authentication Mechanism)

FMT_MOF.1.1(7) - The TSF shall restrict the ability to *determine and modify the behavior of* the functions:[

- biometric authentication mechanism]

to the [Administrator].

Application Note: The Administrator has the ability to modify the behavior of biometric authentication mechanism by adjusting the threshold. Determine in this requirement applies to the Administrator being able to query the threshold setting.

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

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1 - The TSF shall restrict the ability to *query, [and set]* the:

- [value of the 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)
- Time duration restricting the authentication of the same user identifier consecutively;
- Administrator configurable number of attempts per minute (FAU_UAU.5.2);
- [selection: [assignment: other data determined by the ST Author], none]];

to [the Administrator].

FMT_REV.1 Revocation

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

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 Administrator].

FMT_REV.1.2 - The TSF shall enforce the rules:

- [revocation of a user's administrative 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.1 Security roles

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification

FMT_SMR.1.1 The TSF shall maintain the roles [administrator].

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

Application Note: The administering of the TOE is limited to the capabilities associated with the administrative role.

5.1.5 Protection of TSF (FPT)

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: loss of continuity in the TOE's physical housing, [*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). The intent of this requirement is to detect if someone has "opened" the TOE's physical housing. Exposing the internal components by "cutting" through the housing or other means of disturbing the integrity of the housing are not addressed by the loss of continuity aspect of this requirement. The ST author is free to address this type of physical tampering by filling in the open assignment. One method of detecting physical tampering could be an interlock switch. When detection of physical tampering occurs an audit record and alarm are generated.

FPT_RCV.2 Recovery from Failure

Hierarchical to:	FPT_RCV.1 Manual recovery
Dependencies:	AGD_OPE.1 Operational user guidance

FPT_RCV.2.1 When automated recovery from [assignment: list of failures/service discontinuities] is not possible, the TSF shall enter a maintenance mode where the ability to return the TOE to a secure state is provided.

FPT_RCV.2.2 For [power failures], the TSF shall ensure the return of the TOE to a secure state using automated procedures.

Application Note: For FPT_RCV.2.1, the ST author fills in the assignment with all the events for which the TOE provides an automated recovery.

The administrative guidance provides an 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_TST.1 TSF testing (for the TSF)

Hierarchical to: No other components.

Dependencies: FPT_AMT.1 Abstract machine testing

FPT_TST_1.1 – **Refinement:** The TSF shall run a suite of self-tests *during initial start-up, periodically during normal operation as specified by the authorized user, and at the request of the authorized user* to demonstrate the correct operation of [the hardware portions of the TSF].

FPT_TST_1.2 – The TSF shall provide authorized users with the capability to verify the integrity of [threshold setting, parameters under the control of the administrators that are used to enforce the security policies, [selection: [assignment: other TSF data as determined by the ST Author], none]].

FPT_TST_1.3 - The TSF shall provide authorized users with the capability to verify the integrity of stored TSF executable code.

5.1.6 TOE Access (FTA)

FTA_TAB.1 Default TOE access banners

Hierarchical to: No other components.

Dependencies: No dependencies.

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_SSL.3 TSF-initiated termination

Hierarchical to: No other components.

Dependencies: No dependencies.

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

5.2 IT Environment Requirements

5.2.1 Security Audit (FAU)

FAU_SAR.1 Audit review

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_SAR.1.1 - **Refinement:** The **IT environment** shall provide an [administrator] with the capability to read [audit information] from the audit records.

FAU_SAR.1.2 – **Refinement:** The **IT environment** shall provide the audit records in a manner suitable for an Administrator to interpret the information.

FAU_SAR.2 Restricted audit review

FAU_SAR.2.1 – **Refinement:** The **IT environment** shall prohibit all users read access to the audit records, except an administrator that have been granted explicit read-access.

FAU_SAR.3 Selectable audit review

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

a) [user identifier;

b) reference template creation;

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

d) events that generate an alarm].

Application Note: An administrator is the only user who can perform these functions, since they are the only users with read access to all of the audit records in the audit trail. Audit data should be capable of being searched and sorted on all criteria specified in a – c, 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 an administrator may want all the audit records for a specified time period presented together to facilitate their audit review. In item (d), these are the events specified in FAU_SAA.1

FAU_STG.1 Protected audit trail storage

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.1.1 – **Refinement:** The **IT environment** shall protect the stored audit records from unauthorized deletion.

FAU_STG.1.2 - **Refinement:** The **IT environment** shall be able to *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 **IT environment** shall [generate an alarm] if the audit trail exceeds [an administrator settable percentage of storage capacity].

5.2.2 Protection of IT Environment (FPT)

FPT_STM.1 Reliable time stamps

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_STM.1.1 – **Refinement:** The **IT environment** shall be able to provide reliable time stamps for the TOE's use.

5.3 Security Assurance Requirements

Assurance Class	ASSURANCE COMPONENTS	ASSURANCE COMPONENTS DESCRIPTION
DEVELOPMENT	ADV_ARC.1	Architectural Design with domain separation and non-bypassability
	ADV_FSP.2	Security-enforcing Functional Specification
	ADV_TDS.1	Basic design
GUIDANCE DOCUMENTS	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative User guidance
LIFE CYCLE SUPPORT	ALC_CMC.2	Use of a CM system
	ALC_CMS.2	Parts of the TOE CM coverage
	ALC_DEL.1	Delivery procedures
	ALC_FLR.2	**Flaw Reporting Procedures**
TESTS	ATE_COV.1	Evidence of coverage
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - conformance
VULNERABILITY ASSESSMENT	AVA_VAN.2	Vulnerability analysis

Table 2 Assurance Requirements

5.3.1 Class ADV: Development

5.3.1.1 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.

5.3.1.2 ADV_FSP.2 Security-enforcing functional specification

Dependencies: ADV_TDS.1 Basic design

Developer action elements:

ADV_FSP.2.1D The developer shall provide a functional specification.

ADV_FSP.2.2D The developer shall provide a tracing from the functional specification to the SFRs.

Content and presentation elements:

ADV_FSP.2.1C The functional specification shall completely represent the TSF.

ADV_FSP.2.2C The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.2.3C The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.2.4C For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.

ADV_FSP.2.5C For SFR-enforcing TSFIs, the functional specification shall describe direct error messages resulting from processing associated with the SFR-enforcing actions.

ADV_FSP.2.6C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

Evaluator action elements:

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

ADV_FSP.2.2E The evaluator *shall determine* that the functional specification is an accurate and complete instantiation of the SFRs.

5.3.1.3 ADV_TDS.1 Basic design

Dependencies: ADV_FSP.2 Security-enforcing functional specification

Developer action elements:

ADV_TDS.1.1D The developer shall provide the design of the TOE.

ADV_TDS.1.2D 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:

ADV_TDS.1.1C The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.1.2C The design shall identify all subsystems of the TSF.

ADV_TDS.1.3C The design shall describe the behavior of each SFR-supporting or SFR-non-interfering TSF subsystem in sufficient detail to determine that it is not SFR-enforcing.

ADV_TDS.1.4C The design shall summarize the SFR-enforcing behavior of the SFR-enforcing subsystems.

ADV_TDS.1.5C The design shall provide a description of the interactions among SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other subsystems of the TSF.

ADV_TDS.1.6C The mapping shall demonstrate that all behavior described in the TOE design is mapped to the TSFIs that invoke it.

Evaluator action elements:

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

ADV_TDS.1.2E The evaluator *shall determine* that the design is an accurate and complete instantiation of all security functional requirements.

5.3.2 Class AGD: Guidance documents

5.3.2.1 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.

5.3.2.2 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.

5.3.3 Class ALC: Life-cycle support

5.3.3.1 ALC_CMC.2 Use of a CM system

Dependencies: ALC_CMS.1 TOE CM coverage

Developer action elements:

ALC_CMC.2.1D The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.2.2D The developer shall provide the CM documentation.

ALC_CMC.2.3D The developer shall use a CM system.

Content and presentation elements:

ALC_CMC.2.1C The TOE shall be labeled with its unique reference.

ALC_CMC.2.2C The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.2.3C The CM system shall uniquely identify all configuration items.

Evaluator action elements:

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

5.3.3.2 ALC_CMS.2Parts of the TOE CM coverage

Dependencies: No dependencies.

Developer action elements:

ALC_CMS.2.1D The developer shall provide a configuration list for the TOE.

Content and presentation elements:

ALC_CMS.2.1C The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; and the parts that comprise the TOE.

ALC_CMS.2.2C The configuration list shall uniquely identify the configuration items.

ALC_CMS.2.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

Evaluator action elements:

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

5.3.3.3 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.

5.3.3.4 ALC_FLR.2 Flaw 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.

Content and presentation elements:

ALC_FLR.2.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.2.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.2.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.2.5C The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.2.6C The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.

ALC_FLR.2.7C The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.2.8C The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

Evaluator action elements:

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

5.3.4 Class ATE: Tests

5.3.4.1 ATE_COV.1 Evidence of coverage

Dependencies: ADV_FSP.2 Security-enforcing functional specification
ATE_FUN.1 Functional testing

Developer action elements:

ATE_COV.1.1D The developer shall provide evidence of the test coverage.

Content and presentation elements:

ATE_COV.1.1C The evidence of the test coverage shall show the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

Evaluator action elements:

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

5.3.4.2 ATE_FUN.1 Functional 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.

5.3.4.3 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.

5.3.5 Class AVA: Vulnerability assessment

5.3.5.1 AVA_VAN.2 Vulnerability analysis

Dependencies: ADV_ARC.1 Security architecture description
ADV_FSP.1 Basic functional specification
ADV_TDS.1 Basic design
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative procedures

Developer action elements:

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

Content and presentation elements:

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

Evaluator action elements:

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

AVA_VAN.2.2E The evaluator *shall perform* a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.2.3E The evaluator *shall perform* an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.

AVA_VAN.2.4E 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 Basic attack potential.

Application Note: The TOE version used as the basis for testing should include a reference to the specific signature set in place when this activity is conducted.

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. Table 3 illustrates the mapping from Security Objectives to Threats and Policies.

6.1 Rationale for TOE Security Objectives

Table 3 - Security Objectives to Threats and Policies Mappings

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.ACCIDENTAL_ADMIN_ ERROR: An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.	**O.ADMIN_GUIDANCE**: The TOE will provide administrators with the necessary information for secure management.	O.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.
T.BYPASS An attacker may bypass any component of the biometric product and gain unauthorized authentication.	**OE.NON_BYPASS** The IT environment shall ensure that the TOE cannot be bypassed and is always invoked, unless otherwise directed by an administrator (e.g., fallback procedures for users unable to use the TOE) to perform user authentication. **O.PARTIAL_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, through its own interfaces. **OE.TOE_PROTECT** The IT environment shall protect the TOE's executable code from tampering.	OE.NON_BYPASS helps to mitigate this threat by requiring the IT environment to always invoke the TOE to perform user authentication (unless the administrator directs a different procedure). This includes mechanisms or physical protection of the communication path afforded by the environment to protect against an attacker "inserting" data in the communication path between the TOE and the IT environment. O.PARTIAL_SELF_PROTECTION helps to mitigate this threat by requiring that the TOE respond to 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. OE.TOE_PROTECT helps to mitigate this threat by requiring the IT environment to provide mechanisms to protect the TOE's executable code (for example, TOE's executable code is cryptographically signed so the IT environment can verify the source and to detect unauthorized modifications).

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.ARTIFACT An attacker may use an 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 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 provisions that helps to minimize this threat include: enrollment only performed by an administrator, authentication before any TOE-mediated action, strength of secrets and FAR figures specified in this PP.
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. TOE_ACCESS** The TOE will provide mechanisms that control an administrator's logical access to the TOE.	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 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 greater than 1 in 10,000. This threat cannot be totally mitigated and is an inherent weakness in some, if not all, of biometric technologies. O.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 Administrator.

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.POOR_DESIGN: Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.	**O.CONFIGURATION_IDENTIFICATION**: The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly. **O.DOCUMENTED_DESIGN**: The design of the TOE is adequately and accurately documented. **O.VULNERABILITY_ANALYSIS**: The TOE will undergo some vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.	O.CONFIGURATION_IDENTIFICATION help in mitigating this threat by requiring procedures on how to track and address design flaws reported by users. O.DOCUMENTED_DESIGN minimizes this threat, to a degree, by requiring the developer to provide a functional specification describing the TSF and its external interface and a high-level design of the TSF. These evidence aid in the analysis of the TOE in detecting obvious flaws in the design by serving as a reference from which it can be determined if the TOE is an accurate instantiation of the TOE's SFR. O.VULNERABILITY_ANALYSIS_TEST ensures that the design of the TOE is analyzed and tested for obvious flaws that may violate the TSP. Flaws must be characterized, corrected, as appropriate, and documented.
T.POOR_IMPLEMENTATION: Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.	**O.CONFIGURATION_IDENTIFICATION**: The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly., **O.PARTIAL_FUNCTIONAL_TESTING**: The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements. **O.VULNERABILITY_ANALYSIS**: The TOE will undergo some vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.	O.CONFIGURATION_IDENTIFICATION help in mitigating this threat by requiring procedures on how to track and address flaws reported by users when implementing the TOE. O.PARTIAL_FUNCTIONAL_TESTING increases the likelihood that flaws that 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 ensures that the design of the TOE is analyzed and tested for obvious flaws that may violate the TSP. Flaws must be characterized, corrected, as appropriate, and documented.

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
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.PARTIAL_FUNCTIONAL_TESTING**: The TOE will undergo some security functional testing that demonstrates the TSF satisfies the security functional requirements. **O.VULNERABILITY_ANALYSIS**: The TOE will undergo some vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.	O.CORRECT_ TSF_OPERATION helps in mitigating this threat by requiring a suite of self-tests that can be run at the request of an administrator to demonstrate the correct operation of portions of the TSF. O.PARTIAL_FUNCTIONAL_TESTING requires the TOE to go through testing to discover flaws in the design of the TOE. O.VULNERABILITY_ANALYSIS_TEST ensures that the design of the TOE is analyzed and tested for obvious flaws that may violate the TSP. Flaws must be characterized, corrected, as appropriate, and documented.
T.REPLAY_RESIDUAL_IMAGE An attacker may attempt to "reuse" an authorized user's biometric residual characteristic 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 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 may allow an attacker to gain unauthorized authentication.	**O.RESIDUAL_INFORMATION**: The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION counters this threat by ensuring that TSF data is not persistent when resources are released by one function/process and allocated to another function/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.

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
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 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.TAMPER An attacker may modify or otherwise alter the software or hardware components, the connections between them thereby gaining unauthorized authentication.	**O.PARTIAL_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. **OE.TOE_PROTECT** The IT environment shall protect the TOE's executable code from tampering.	O.PARTIAL_SELF_PROTECTION helps in minimizing this threat by requiring that TOE to provide active physical mechanisms, such as mechanical switches, to detect and react to the exposure of the internal TOE components. OE.TOE_PROTECT depends on the IT environment requirements to protect (e.g., physically, encrypted) the communication paths between physically separate parts of the TOE and between the TOE and environment (IT and non-IT).
T.TSF_COMPROMISE: A user or process may cause, through an unsophisticated attack,, TSF data, or executable code to be inappropriately accessed (viewed, modified, or deleted).	**O.RESIDUAL_INFORMATION**: The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated. **O.PARTIAL_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 through its own interfaces. **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. **OE.TOE_PROTECT** The IT environment shall protect the TOE's executable code from tampering.	O.RESIDUAL_INFORMATION helps mitigating this threat by ensuring that the contents of resources are not available once the TSF is finished processing the TSF data. Since the TOE relies on the IT environment to provide some protection of the TSF, O.PARTIAL_SELF_PROTECTION helps minimize this threat by enforcing separation between the security domains of subjects in the TOE's Scope of Control. O.MANAGE helps mitigate this threat by requiring restricted access to functions that manage the TOE to be accessible to administrators only. OE.TOE_PROTECT helps mitigate this threat by requiring the IT environment to protect (e.g., physically, encrypted) the communication paths between physically separate parts of the TOE and between the TOE and environment (IT and non-IT).
T.UNATTENDED_ SESSION: An attacker may gain unauthorized access to an administrator's unattended session.	**O.TOE_ACCESS**: The TOE will provide mechanisms that control a user's logical access to the TOE.	O.TOE_ACCESS helps mitigate this threat by requiring that an administration session terminates after a pre-defined amount of inactive time.

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.UNAUTHORIZED_ ACCESS: A user may gain access to administrative functions for which they are not authorized according to the TOE security policy.	**O. TOE_ACCESS** The TOE will provide mechanisms that control an administrator's logical access to the TOE. **O.ADMIN_ROLE** The TOE will provide an administrator role to isolate administrative actions from untrusted user actions.	O.TOE_ACCESS helps mitigate this threat by requiring the addition of non-biometric authentication mechanisms to authenticate administrators, and by requiring authentication before any administrative action. It also helps mitigate this threat by requiring a maximum number of authentication failures to authenticate administrators. Settings in the FAR performance figures also help in minimizing this threat. O.ADMIN_ROLE is used to ensure there is a mechanism that determines whether a user physically interacting with the TOE has administrative capabilities, and in turn is able to access logical functions of the TOE not intended for use by untrusted users.
T.UNIDENTIFIED_ACTIONS: The administrator may not have the ability to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.	**O.ALARM_GENERATION**: The TOE will provide the capability to detect and alert an administrator of a potential security violation.	O.ALARM_GENERATION is used to mitigate this threat by ensuring the TOE monitors for certain potential security violations by providing the Administrator with a required minimum set of configurable events that could indicate a potential security violation. By configuring these 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.
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.ADMIN_GUIDANCE** The TOE will provide administrators with the necessary information for secure delivery and management.	O.MAINT_MODE helps to mitigate this threat by ensuring that the TOE does not continue to operate in an insecure state when a failure occurs. Upon a power failure, the TOE must attempt to automatically recover from the discontinuity. If the TOE cannot automatically recover from a failure, the TOE enters a state that disallows further biometric authentication attempts and requires the 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 hardware and software 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 These integrity tests are not meant to address the hardware platform that may be underlying the TOE, but rather focus on the hardware portions that are required to be part of the TOE (e.g., capture device). O.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).

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
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 a 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.	**O.AUDIT_GENERATION**: The TOE will provide the capability to detect and create records of security-relevant events associated with users. **OE. AUDIT_TRAIL_REVIEW** The capability to selectively view audit information generated by the TOE is provided by the IT environment. **OE.AUDIT_PROTECTION** The IT Environment protects the audit information generated by the TOE from modification, disclosure and loss. **OE.TIME_STAMPS** The IT environment shall provide reliable time stamps and the capability for the administrator to set the time used for these time stamps.	O.AUDIT_GENERATION addresses this policy by providing the 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). OE. AUDIT_TRAIL_REVIEW contributes to satisfying this policy by requiring the IT environment to provide administrators the capability to review the audit events generated by the TOE is a way that facilitates efficient review of events deemed relevant by the administrators. OE.AUDIT_PROTECTION contributes to satisfying this policy by requiring the IT environment to control the access to the audit trail and not allowing the modification or unauthorized deletion of audit events that could obscure a user's actions. OE.TIME_STAMPS plays a role in supporting this policy by requiring the IT environment to provide a reliable time stamp for auditing. The audit mechanism is required to include the current date and time in each audit record.

6.2 Rationale for the Security objectives for the Environment

The IT environment objectives map to their associated IT environment requirements as follows:

OE.AUDIT_TRAIL_REVIEW - The capability to selectively view audit information generated by the TOE is provided by the IT environment. The IT environment requirements FAU_SAR.1, FAU_SAR.2, FAU_SAR.3 ensure the IT environment provides the administrator with the ability to review the audit trail and base their review on selected criteria.

OE.AUDIT_PROTECTION - The IT environment protects the audit information generated by the TOE from modification, disclosure and loss. The IT environment requirements FAU_ STG.1, and FAU_STG.3 ensure that the IT environment offer suitable protection of the audit trail so that audit data is not maliciously modified or deleted.

OE.TIME_STAMPS - The IT environment shall provide reliable time stamps and the capability for the administrator to set the time used for these time stamps. The IT environment requirement FPT_STM.1 ensures that the IT environment provides the TOE with reliable time so that audit records have a time stamp that ensures the sequence of audit events can be determined.

OE.NON_BYPASS - The IT environment shall ensure that the TOE cannot be bypassed and is always invoked, unless otherwise directed by an administrator (e.g., fallback procedures for users unable to use the TOE) to perform user authentication. The IT environment requirement ADV_ARC.1 ensures that the TOE's architecture incudes the requirement for authenticating users when it is configured to do so by the administrator.

OE.TOE_PROTECT – The IT environment shall protect the TOE's executable from tampering. The IT environment requirement ADV_ARC.1 architecture requires that the IT environment protects the TOE from unprivileged code running on the IT environment from modifying the TOE's software. The IT environment cannot prevent the malicious use of privileged code from tampering with the TOE. In order for this requirement to be satisfied the administrator must install the TOE and configure the IT environment such that untrusted users do not have write access to the TOE's executables.

The non-IT security objectives OE.COMM_PROTECT, OE.ENROLLMENT_APPROVAL, OE.NO_EVIL, OE.NO_GENERAL_PURPOSE and OE.OPERATING_RANGE are simply restatements of their corresponding assumptions and therefore are trivially mapped to those assumptions and are deemed suitable to cover those assumptions.

The objective OE.BIOMETRICS_PACKAGE_PROTECT is different from the other environment objectives in that it can be addressed by a combination of the non-IT environment (i.e., the communication path is physically protected) and the IT environment, or could be completely addressed by IT environment requirements. There are many ways in which IT environment requirements could be applied. Encryption could be used as specified in the medium robustness biometric PP, access control mechanisms could be specified in the IT environment that would control subjects access to the biometrics package, or a combination could be used (e.g., use encryption to protect the package during transmission, and use an access control mechanism to control access to the biometric package when it resides in storage). The PP authors felt the end-user should be aware of the mechanisms that could be employed without dictating a solution. In any case, this objective is suitable to cover the assumption A.BIOMETRICS_PACKAGE_PROTECT since the objective is a simple restatement of the assumption.

6.3 Rationale for TOE Security Requirements

Table 4 - Rationale for TOE Security Requirements

Objectives	Requirements Addressing the Objective	Rationale
O.ADMIN_GUIDANCE: The TOE will provide administrators with the necessary information for secure 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 maintain security during the delivery of the TOE, in whole or in parts. 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 Preparative User Guidance (AGD_PRE)documentation ensures that once the administrator has followed the installation and configuration guidance the result is a TOE in a secure configuration. The AGD assurance requirements family ensures proper documentation for administrators. 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 threshold, and what quality metrics are acceptable when enrolling a user. The documentation also provides a description of how to setup and review the auditing features of the TOE. Although the AGD_OPE.1 also is intended for non-administrative users, it could be used to provide guidance on security that is common to both administrators and non-administrators (e.g., password management guidelines). Since both administrative and non-administrative users of this TOE present their biometric characteristic to the TOE, this document would instruct all users how to correctly supply their characteristic. AGD_OPE.1 AND AGD_PRE.1 analysis during evaluation will ensure that the guidance documentation is complete and can be followed unambiguously to ensure the TOE is not mis-configured in an unsecure state due to confusing guidance.
O.ADMIN_ROLE The TOE will provide an administrator role to isolate administrative actions from untrusted user actions.	FMT_SMR.1	FMT_SMR.1 requires the existence of the administrator role in charge of configuring the TOE's security policies.
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 FAU_SEL.1-NIAP-0407	FAU_GEN.1-NIAP-0410 defines the set of events that the TOE must be capable of recording. This requirement ensures that the 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.

Objectives	Requirements Addressing the Objective	Rationale
		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 Administrator with useful information (e.g., a specific user is targeted by an attacker). 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). FAU_SEL.1-NIAP-0407 allows the 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.
O.ALARM_GENERATION The TOE will provide the capability to detect and alert an administrator of a potential security violation.	FAU_ARP.1 FAU_SAA.1-NIAP-0407	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 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 Administrator did not select the event of biometrics package modification in FAU_SEL, the 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.
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	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 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

Objectives	Requirements Addressing the Objective	Rationale
		establishes the restrictions on the creation/modification of a biometric package (which includes the reference template). This requirement also mandates that the 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.
O.CONFIGURATION_ IDENTIFICATION: The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly.	ALC_CMS.2 ALC_FLR.2	ALC_CMS.2 contributes to this objective by requiring the developer to use a configuration management (CM) system and to provide CM documentation. It is also used to ensure the appropriate items are under CM control. 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.
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.1	This objective is met by using FPT_TST.1, which requires the TOE to provide the administrator with the capability to run a suite of self-tests on request to demonstrate the correct operation of the hardware portions of the TSF, to verify the integrity of TSF data, and to verify the integrity of stored TSF executable code.
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.DOCUMENTED_DESIGN: The design of the TOE is adequately and accurately documented.	ADV_FSP.2 ADV_TDS.1	ADV_FSP.2 contributes to this objective by requiring evidence showing that the external interfaces of the TSF conform to security functional requirements. ADV_TDS.1 contributes to this objective by requiring a description of the TSF in terms of subsystems, describing their purpose and function, and the provision of evidence identifying the security functions contained in each subsystem. Finally, ADV_TDS.1 also contributes to this objective by requiring analytical evidence that the TSF's design is decomposed into correctly between the different levels of design decomposition, and that these levels of decomposition demonstrate an accurate, consistent and complete instantiation of the security functional requirements as expressed in the ST.
O.MAINT_MODE The TOE shall provide a mode from which recovery or initial startup procedures can be performed.	FPT_RCV.2	This objective is met by using the FPT_RCV.2 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.1.1) 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 further biometric authentication and allows for 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

Objectives	Requirements Addressing the Objective	Rationale
		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) FMT_MOF.1(7) FMT_MTD.1 FMT_REV.1	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 administrator to control the security function associated with audit generation. This requirement also allows the Administrator to affect the events that are audited, turn audit off/on, and requires the capability exists that the Administrator can determine/view the configuration settings. FMT_MOF.1(2) provides the Administrator the capability to select the event types, as well as the events that are monitored to generate alerts. FMT_MOF.1(3) provides the Administrator the ability to modify the behavior of the tests. This ensures that the self-tests will run no less than a frequency determined as necessary by the Administrator. FMT_MOF.1(4) is 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 Administrator can restore the TOE. FMT_MOF.1(5) restricts the ability to enroll users to Administrator. Correctly enrolling users is vital to the TOE's ability to correctly authenticate users. 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(6) allows the Administrator to enable or disable the need for administrators to use the non-biometric authentication mechanism. FMT_MOF.1(7) provides capability to modify the behavior of the biometric authentication mechanism. This includes setting the threshold that affects level of a match required in the comparison of the reference template and live template. Since the essence of a biometrics TOE is to perform authentication, FMT_MTD.1 ensures that only the Administrator has the flexibility to configure the TOE such that it behaves as required by their operational constraints. 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. This requirement identifies the TSF data the PP authors felt was essential in allowing a Administrator to manage the TOE. FMT_REV.1 ensures that the Administrator has the ability to revoke the assignment of a role to a specific user. This revocation is immediate.

Objectives	Requirements Addressing the Objective	Rationale
O.PARTIAL_FUNCTIONAL_ TESTING: The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements.	ATE_COV.1 ATE_FUN.1 ATE_IND.2	ATE_COV.1 requires the developer to provide evidence that shows correspondence between test in the documentation and the TSFI. While this component does not require that all TSFI be tested, it is expected that the TSFI associated with authentication are completely tested as that is the main function of this TOE. 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 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_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.RESIDUAL_ INFORMATION: The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.	FDP_RIP.2	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.
O.PARTIAL_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 through its own interfaces.	ADV_ARC.1 FPT_PHP_(EXT).1	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_PHP_(EXT).1 plays a diminished role in satisfying this objective in that it can generate an alarm and audit record notifying the 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. An attacker could disable the power and remove the housing and gain access to the internals of the TOE and render the TOE unable to enforce its security policies. The TOE is not expected to be able to detect this type of attack.
O.TOE_ACCESS: The TOE will provide mechanisms that control a user's logical access to the TOE.	FIA_AFL.1(1) FIA_AFL.1(2) FIA_AFL.1(3) FIA_ATD.1 FIA_UID.2 FIA_SOS.1 FIA_SOS.2 FIA_UAU.2 FIA_UAU.5 FIA_UAU.7 AVA_VAN.2 FTA_SSL.3	FIA_AFL.1 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 the Administrator settable threshold that prevents unauthorized users from gaining access to an administrators account by locking the targeted account until the Administrator takes some action (e.g., re-enables the account) or for some 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)

Objectives	Requirements Addressing the Objective	Rationale
		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. 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 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. 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 Administrator. If the 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 6 . It was thought that a PIN that consisted of 6 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 Administrator’s discretion, for administrative users. The thinking is that if the capture device experience problems, the 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.

Objectives	Requirements Addressing the Objective	Rationale
		FIA_UAU.2 simply 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 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 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., your comparison failed to pass the threshold by a factor of X). The AVA_VAN.2 requirement is applied to the local non-biometric authentication mechanism. For this TOE, the vulnerability analysis is specified for an attack potential of basic. 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 medium-attack potential, as defined in Annex B of the CEM. 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 Administrator defined period, the session is terminated. This requirement applies both to remote and direct connections to the TOE.
O.VULNERABILITY_ANALYSIS: The TOE will undergo some vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.	AVA_VAN.2	The **AVA_VAN.2** component provides the necessary level of confidence that vulnerabilities do not exist in the TOE that could cause the security policies to be violated. AVA_VAN.2 requires the evaluator to perform a search for potential vulnerabilities in all the TOE deliverables. 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 basic attack potential, which is in keeping with the desired assurance level of this TOE. This component provides the confidence that security flaws do not exist in the TOE that could be exploited by a threat agent of basic attack potential to violate the TOE's security policies.

6.4 Rationale for Assurance Requirements

The EAL definitions in Part 3 of the CC were reviewed and the Basic Robustness Assurance Package (EAL2 augmented with assurance requirements ALC_FLR.2) was believed to best achieve this goal. The sponsor concluded that EAL2 augmented is applicable since this PP addresses circumstances where developers and users require a low to moderate level of independently assured security in commercial products. Rationale for individual assurance requirements is provided in Table 4.

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 basic 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 5 identifies the functional requirement, its correspondent dependency and the analysis and rationale for not supporting the dependency in this PP.

Requirement	Dependency	Dependency Analysis and Rationale
FIA_AFL.1(1-2)	FIA_UAU.1	FIA_AFL.1(1) and FIA_AFL.1(2) apply to non-administrative users. These users do not authenticate themselves to the TOE in order to perform actions on the TOE that are to be mediated. FIA_UAU.1 is intended to be used to ensure users must authenticate themselves before they perform any actions to be mediated by the TOE. In this scenario, users must be authenticated so the TOE can mediate their actions based on the users' credentials or rights. Therefore, FIA_UAU.1 is unnecessary and the dependency on these two iterations is broken.
FIA_UAU.2 FIA_UAU.7	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.

Requirement	Dependency	Dependency Analysis and Rationale
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.
FPT_TST.1	FPT_AMT.1	While this TOE does have an underlying abstract machine (the IT environment that supplies an infrastructure and is required to provide support for SFRs) it is unclear what reliance can be placed on the abstract machine's result from "self-tests" on that machine (e.g., if the abstract machine is compromised, it would provide the "expected" results for self-tests). It is the PP author's opinion that requiring the TOE developer to provide testing of the underlying operating system and hardware is inappropriate for the basic robustness level of assurance, and that the FPT_TST.1 requirements levied on the TOE ensure that the hardware mechanisms of the biometric capture device, and the integrity of the TSF data/executables is sufficient for the level of assurance being requested by the PP.

Table 5 - Unsupported Dependency Rationale

6.6 Rationale for Extended requirements

Table 6 presents the rationale for the inclusion of the extended requirements found in this PP.

Extended Requirement	Identifier	Rationale

Extended Requirement	Identifier	Rationale
FIA_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.
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.

Table 6 - Rationale for Extended Requirements

7.0 REFERENCES

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

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) *Information Assurance Technical Framework,* Version 3.1, September 2002.

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 -
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[4] and the disclosure and modification of data.[5]
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.
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.

---

[4] Hardware and software.
[5] Stored or communicated.

Authorized user -- An authenticated user who may, in accordance with the TSP, perform an operation.
Availability -- Timely[6], 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,
- 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.

- ---

[6] According to a defined metric.

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.[7]
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.
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.

- ---

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

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.
- **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:

CC	Common Criteria for Information Technology Security Evaluation
DoD	Department of Defense
EAL	Evaluation Assurance Level
GIG	Global Information Grid
I&A	Identification and Authentication
IATF	Information Assurance Technical Framework
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
PP	Protection Profile
SFP	Security Function Policy
ST	Security Target
TOE	Target of Evaluation
TSE	TOE Security Environment
TSF	TOE Security Function
TSP	TOE Security Policy