Medium Robustness: CCv2.3 → CCv3.1

This collection of tables presents a mapping between the requirements of Medium Robustness (as defined in the CIM) and the requirements of CCv3.1. The right column cites the corresponding requirements from CCv3.1.

ACM_AUT.1 (v2.3)	ALC_CMC.4 (v3.1)
ACM_AUT.1.1C - The CM system shall provide an automated means by which only authorized changes are made to the TOE implementation representation.	ALC_CMC.4.4C - The CM system shall provide automated measures such that only authorized changes are made to the configuration items.
ACM_AUT.1.2C - The CM system shall provide an automated means to support the generation of the TOE.	ALC_CMC.4.5C - The CM system shall support the production of the TOE by automated means.
ACM_AUT.1.3C - The CM plan shall describe the automated tools used in the CM system.	ALC_CMC.4.7C - The CM plan shall describe how the CM system is used for the development of the TOE.
ACM_AUT.1.4C - The CM plan shall describe how the automated tools are used in the CM system.	

ACM_CAP.4 (v2.3)	ALC_CMC.4; ALC_CMS.4 (v3.1)
ACM_CAP.4.1C - The reference for the TOE shall be unique to each version of the TOE.	ALC_CMC.4.1C - The TOE shall be labeled with its unique reference.
ACM_CAP.4.2C - The TOE shall be labeled with its reference.	
ACM_CAP.4.3C - The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.	ALC_CMS.4.1C - The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation representation; and security flaw reports and resolution status. ALC_CMC.4.6C - The CM documentation shall include a CM plan. ALC_CMC.4.8C - The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.
ACM_CAP.4.4C - The configuration list shall uniquely identify all configuration items that comprise the TOE.	ALC_CMS.4.2C - The configuration list shall uniquely identify the configuration items.
ACM_CAP.4.5C - The configuration list shall describe the configuration items that comprise the TOE.	ALC_CMS.4.1C - The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the

	implementation representation; and security flaw reports and resolution status.
ACM_CAP.4.6C - The CM documentation shall describe the method used to uniquely identify the configuration items that comprise the TOE.	ALC_CMC.4.2C - The CM documentation shall describe the method used to uniquely identify the configuration items.
ACM_CAP.4.7C - The CM system shall uniquely identify all configuration items that comprise the TOE.	ALC_CMC.4.3C - The CM system shall uniquely identify all configuration items.
ACM_CAP.4.8C - The CM plan shall describe how the CM system is used.	ALC_CMC.4.7C - The CM plan shall describe how the CM system is used for the development of the TOE.
ACM_CAP.4.9C - The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.	ALC_CMC.410C - The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.
ACM_CAP.4.10C - The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.	ALC_CMC.4.9C - The evidence shall demonstrate that all configuration items are being maintained under the CM system.
ACM_CAP.4.11C - The CM system shall provide measures such that only authorised changes are made to the configuration items.	ALC_CMS.4.3C - For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.
ACM_CAP.4.12C - The CM system shall support the generation of the TOE.	ALC_CMC.4.7C The CM plan shall describe how the CM system is used for the development of the TOE. ALC_CMC.4.8C The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE. Generation includes development and production.
ACM_CAP.4.13C - The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.	ALC_CMC.4.8C - The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

ACM_SCP.2 (v2.3)	ALC_CMS.4 (v3.1)
ACM_SCP.2.1C - The list of configuration items shall include the following: implementation representation; security flaws; and the evaluation evidence required by the assurance components in the ST.	ALC_CMS.4.1C - The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation representation; and security flaw reports and resolution status.

ADO_DEL.2 (v2.3)	ALC_DEL.1; AGD_PRE.1 (v3.1)
ADO_DEL.2.1C - The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.	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.
ADO_DEL.2.2C - The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.	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.
ADO_DEL.2.3C - The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.	

ADO_IGS.1 (v2.3)	AGD_PRE.1 (v3.1)
ADO_IGS.1.1C - The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.	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.
ADO_IGS.1.2E - The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.	AGD_PRE.1.2E - The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

ADV_ARC_EXP (MR)	ADV_ARC.1 (v3.1)
ADV_ARC_EXP.1.1C - The presentation of the architectural design of the TSF shall be informal.	The requirement for informal presentation has been dropped because it is the lowest level; anything more formal will still meet the requirement.
ADV_ARC_EXP.1.2C - The architectural design shall be internally consistent.	There is no longer an explicit check for inconsistency; if it is inconsistent, it doesn't make sense, so it can't fulfill ANY of the requirements.
ADV_ARC_EXP.1.3C - The architectural	ADV_ARC.1.2C - The security architecture

design shall describe the design of the TSF self-protection mechanisms.	description shall describe the security domains maintained by the TSF consistently with the SFRs.
ADV_ARC_EXP.1.4C - The architectural design shall describe the design of the TSF in detail sufficient to determine that the security enforcing mechanisms cannot be bypassed.	ADV_ARC.1.5C - The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.
ADV_ARC_EXP.1.5C - The architectural design shall justify that the design of the TSF achieves the self-protection function.	ADV_ARC.1.4C - The security architecture description shall demonstrate that the TSF protects itself from tampering.
ADV_ARC_EXP.1.2E - The evaluator shall analyze the architectural design and dependent documentation to determine that FPT_SEP and FPT_RVM are accurately implemented in the TSF.	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.
Added description of the level of detail. Also, Medium Robustness defined no requirements to ensure secure initialization.	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.1.3C - The security architecture description shall describe how the TSF initialisation process is secure.

ADV_FSP_EXP (MR)	ADV_FSP.5 (v3.1)
ADV_FSP_EXP.1.1C - The functional specification shall completely represent the TSF.	ADV_FSP.5.1C - The functional specification shall completely represent the TSF.
ADV_FSP_EXP.1.2C - The functional specification shall be internally consistent.	There is no longer an explicit check for inconsistency; if it is inconsistent, it doesn't make sense, so it can't fulfill ANY of the requirements.
ADV_FSP_EXP.1.3C - The functional specification shall describe the external TSF interfaces (TSFIs) using an informal style.	ADV_FSP.5.2C - The functional specification shall describe the TSFI using a semi-formal style.
ADV_FSP_EXP.1.4C - The functional specification shall designate each external TSFI as security enforcing or security supporting.	Explicit designation seen as unnecessary; the status will be apparent from the information provided.
ADV_FSP_EXP.1.5C - The functional specification shall describe the purpose and method of use for each external TSFI.	ADV_FSP.5.3C - The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP_EXP.1.6C - The functional specification shall identify and describe all parameters associated with each external TSFI.	ADV_FSP.5.4C - The functional specification shall identify and describe all parameters associated with each TSFI.
ADV_FSP_EXP.1.7C - For security enforcing external TSFIs, the functional specification shall describe the security enforcing effects and security enforcing exceptions.	ADV_FSP.5.5C - The functional specification shall describe all actions associated with each TSFI. The additional requirement of a description (rather than a specification of any kind of detail) is non-zero, but comparatively small
ADV_FSP_EXP.1.8C - For security enforcing external TSFIs, the functional specification shall describe direct error messages resulting from security enforcing effects and exceptions.	ADV_FSP.5.6C - The functional specification shall describe all direct error messages that may result an invocation of each TSFI. ADV_FSP.5.7C - The functional specification shall describe all error messages that do not result an invocation of each TSFI. ADV_FSP.5.8C - The functional specification shall provide a rationale for each error message contained in the TSF implementation yet does not result from an invocation of a TSFI. This requirement expands the scope of error messages to meet the medium robustness requirements of ATE_DPT.3.
This mapping used to be in RCR (see RCR, below)	ADV_FSP.5.9C - The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.
ADV_FSP_EXP.1.2E - The evaluator shall determine that the functional specification is an accurate and complete instantiation of the user-visible TOE security functional requirements. "User-visible" excludes FPT_SEP, RVM	ADV_FSP.5.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence. ADV_FSP.5.2E - The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs. FPT_SEP, RVM no longer exist

ADV_HLD_EXP (MR); ADV_LLD_EXP (MR)	ADV_TDS.4 (v3.1)
ADV_HLD_EXP.1.1C - The high-level design shall describe the structure of the TOE in terms of subsystems.	ADV_TDS.4.1C - The design shall describe the structure of the TOE in terms of subsystems.
ADV_HLD_EXP.1.2C - The high-level design shall be internally consistent.	There is no longer an explicit check for inconsistency; if it is inconsistent, it doesn't make sense, so it can't fulfill ANY of the requirements.
ADV_HLD_EXP.1.3C - The high level design	The requirement for informal presentation has

shall describe the subsystems using an informal style.	been dropped because it is the lowest level; anything more formal will still meet the requirement.
ADV_HLD_EXP.1.4C - The high-level design shall describe the design of the TOE in sufficient detail to determine what subsystems of the TOE are part of the TSF.	ADV_TDS.4.3C - The design shall identify all subsystems of the TSF. Explicit designation seen as unnecessary; the status will be apparent from the information provided.
ADV_HLD_EXP.1.5C - The high-level design shall identify all subsystems in the TSF, and designate them as either security enforcing or security supporting.	
ADV_HLD_EXP.1.6C - The high-level design shall describe the structure of the security-enforcing subsystems.	ADV_TDS.4.4C - The design shall provide a description of each subsystem of the TSF.
ADV_HLD_EXP.1.7C - For security-enforcing subsystems, the high-level design shall describe the design of the security-enforcing behavior.	Emphasis shifts away from subsystems to modules. These elements are somewhat covered by 3.4C. If the details of subsystems are still required, it should be done as an explicitly-stated requirement.
ADV_HLD_EXP.1.8C - For security-enforcing subsystems, the high-level design shall summarize any non-security-enforcing behavior.	
ADV_HLD_EXP.1.9C - The high-level design shall summarize the behavior for security-supporting subsystems.	
ADV_HLD_EXP.1.10C - The high-level design shall summarize all interactions between subsystems of the TSF.	ADV_TDS.4.5C - The design shall provide a description of the interactions among all subsystems of the TSF.
ADV_HLD_EXP.1.11C - The high-level design shall describe any interactions between the security-enforcing subsystems of the TSF.	
ADV_HLD_EXP.1.2E - The evaluator shall determine that the high-level design is an accurate and complete instantiation of all user-visible TOE security functional requirements with the exception of FPT_SEP and FPT_RVM.	ADV_TDS.4.1E - The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence. ADV_TDS.4.2E - The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements. FPT_SEP, RVM no longer exist
ADV_LLD_EXP.1.1C - The presentation of the low-level design shall be informal.	The requirement for informal presentation has been dropped because it is the lowest level; anything more formal will still meet the requirement.

ADV_LLD_EXP.1.2C - The presentation of the low-level design shall be separate from the implementation representation.	The prohibition for collocation was seen as unnecessary.
ADV_LLD_EXP.1.3C - The low-level design shall be internally consistent.	There is no longer an explicit check for inconsistency; if it is inconsistent, it doesn't make sense, so it can't fulfill ANY of the requirements.
ADV_LLD_EXP.1.4C - The low-level design shall describe the TSF in terms of modules, designating each module as either security enforcing or security-supporting.	ADV_TDS.4.2C - The design shall describe the TSF in terms of modules, designating each module as SFR-enforcing, SFR-supporting, or SFR-non-interfering. Explicit designation seen as unnecessary; the status will be apparent from the information provided.
ADV_LLD_EXP.1.5C - The low-level design shall identify and describe data that are common to more than one module, where any of the modules is a security-enforcing module.	This was removed because it was seen as being too software-specific
ADV_LLD_EXP.1.6C - The low level design shall describe each security-enforcing module in terms of its purpose, interfaces, return values, called interfaces to other modules, and global variables.	ADV_TDS.4.7C - The design shall describe each SFR-enforcing and SFR-supporting module in terms of its purpose. ADV_TDS.4.8C - The design shall describe each SFR-enforcing and SFR-supporting module in terms of its SFR-related interfaces, return values from those interfaces, and called interfaces to other modules. "Global variables" was removed because it was too software-specific.
ADV_LLD_EXP.1.7C - For each security-enforcing module, the low level design shall provide an algorithmic description detailed enough to represent the TSF implementation. *Application Note: An algorithmic description contains sufficient detail such that two different programmers would produce functionally equivalent code, although data structures, programming methods, etc. may differ.*	"Algorithmic description seen as being unnecessary.
ADV_LLD_EXP.1.8C - The low level design shall describe each security-supporting module in terms of its purpose and interaction with other modules.	ADV_TDS.4.9C - The design shall describe each SFR-non-interfering module in terms of its purpose and interaction with other modules.
ADV_LLD_EXP.1.2E - The evaluator shall determine that the low-level design is an accurate and complete instantiation of all TOE security functional requirements, with the	ADV_TDS.4.1E - The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

exception of FPT_SEP and FPT_RVM.	ADV_TDS.4.2E - The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements. FPT_SEP, RVM no longer exist
This mapping used to be in RCR (see RCR, below). The developer provides it.	ADV_TDS.4.6C - The design shall provide a mapping from the subsystems of the TSF to the modules of the TSF.
	ADV_TDS.4.10C - The mapping shall demonstrate that all behaviour described in the TOE design is mapped to the TSFIs that invoke it.

ADV_IMP.2 (v2.3)	ADV_IMP.1 (v3.1)
ADV_IMP.2.1C - The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.	ADV_IMP.1.1C - The implementation representation shall define the TSF to a level of detail such that the TSF can be generated without further design decisions.
ADV_IMP.2.2C - The implementation representation shall be internally consistent.	There is no longer an explicit check for inconsistency; if it is inconsistent, it doesn't make sense, so it can't fulfill ANY of the requirements.
ADV_IMP.2.3C - The implementation representation shall describe the relationships between all portions of the implementation.	This should already be covered by the TDS descriptions.
ADV_IMP.2.2E - The evaluator shall determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements.	ADV_IMP.1.3C - The mapping between the TOE design description and the sample of the implementation representation shall demonstrate their correspondence.
This prevents giving a pretty-printed version of the source. It was added to ensure that tools used for navigating the code would work. It adds no work for the developer.	ADV_IMP.1.2C - The implementation representation shall be in the form used by the development personnel.

ADV_INT_EXP (MR)	ADV_INT.3 (v3.1)
ADV_INT_EXP.1.1C - The software architectural description shall identify the SFP-enforcing and non-SFP-enforcing modules.	The approach in v2.3 was centered on software only. For example, the metrics described and used as the basis for "good practice" come from software engineering literature. In order to make the criteria relevant to all parts of all TOEs, the approach was changed to identify and describe the metrics used to judge "good practice", and then to measure against
ADV_INT_EXP.1.2C - The TSF modules shall be identical to those described by the low level design (ADV_LLD_EXP.1.4C).	
ADV_INT_EXP.1.3C - The software architectural description shall provide a	

justification for the designation of non-SFP-enforcing modules that interact with the SFP-enforcing module(s). ADV_INT_EXP.1.4C - The software architectural description shall describe the process used for modular decomposition. ADV_INT_EXP.1.5C - The software architectural description shall describe how the TSF design is a reflection of the modular decomposition process. ADV_INT_EXP.1.6C - The software architectural description shall include the coding standards used in the development of the TSF. ADV_INT_EXP.1.7C - The software architectural description shall provide a justification, on a per module basis, of any deviations from the coding standards. ADV_INT_EXP.1.8C - The software architectural description shall include a coupling analysis that describes intermodule coupling for the SFP-enforcing modules. ADV_INT_EXP.1.9C - The software architectural description shall provide a justification, on a per module basis, for any coupling or cohesion exhibited by SFP-enforcing modules, other than those permitted. ADV_INT_EXP.1.10C - The software architectural description shall provide a justification, on a per module basis, that the SFP-enforcing modules are not overly complex.	those metrics. Since PP authors by definition know what kind of TOE is being described, if they have an idea of specific metrics that should be used to define "good practice", these can be incorporated into the requirements as refinements or through the use of explicitly-stated requirements. ADV_INT.3.1C - The justification shall explain the characteristics used to judge the meaning of "well-structured" and "complex". ADV_INT.3.2C - The TSF internals description shall demonstrate that the entire TSF is well-structured. ADV_INT.3.3C - The TSF internals description shall demonstrate that the entire TSF is well-structured and is not overly complex.
ADV_INT_EXP.1.2E - The evaluator shall perform a cohesion analysis for the modules that substantiates the type of cohesion claimed for a subset of SFP-enforcing modules. ADV_INT_EXP.1.3E - The evaluator shall perform a complexity analysis for a subset of TSF modules.	ADV_INT.3.2E - The evaluator shall perform an internals analysis on the entire TSF.

ADV_RCR.1 (v2.3)	CCv3
ADV_RCR.1.1C - For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security	The correspondence between adjacent pairs (i.e. the RCR family) was distributed among each of the different levels of abstraction: each

functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.	examines a correspondence to the previous (see specifics below). This distributed approach reflects the approach in CCv1. ADV_FSP.5.9C - The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification. ADV_TDS.4.10C - The mapping shall demonstrate that all behaviour described in the TOE design is mapped to the TSFIs that invoke it.

ADV_SPM.1 (v2.3)	CCv3
ADV_SPM.1.1C - The TSP model shall be informal. ADV_SPM.1.2C - The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled. ADV_SPM.1.3C - The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled. ADV_SPM.1.4C - The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.	Version 3 presumes the collection of security objectives constitutes an informal model of the TOE's security behavior. That is, there are no CC v3 requirements for any kind of security policy model other than formal. The v2 approach of policies that "can be modeled" is also replaced by the assignment that explicitly identified the policies that are being modelled. ADV_SPM.1.1C - The model shall be in a formal style, supported by explanatory text as required, and identify the security policies of the TSF that are modelled. ADV_SPM.1.2C - For all policies that are modelled, the model shall define security for the TOE and provide a formal proof that the TOE cannot reach a state that is not secure. ADV_SPM.1.3C - The correspondence between the model and the functional specification shall be at the correct level of formality. ADV_SPM.1.4C - The correspondence shall show that the functional specification is consistent and complete with respect to the model. ADV_SPM.1.5C - The demonstration of correspondence shall show that the interfaces in the functional specification are consistent and complete with respect to the policies in the

	ADV_SPM.1.1D assignment.

AGD_ADM.1 (v2.3)	AGD_OPE.1 (v3.1)
AGD_ADM.1.1C - The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.	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_ADM.1.2C - The administrator guidance shall describe how to administer the TOE in a secure manner.	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.1.5C - 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_ADM.1.3C - The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.	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_ADM.1.4C - The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.	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_ADM.1.5C - The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.	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_ADM.1.6C - The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.	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_ADM.1.7C - The administrator guidance shall be consistent with all other documentation supplied for evaluation.	AGD_OPE.1.7C - The operational user guidance shall be clear and reasonable.
AGD_ADM.1.8C - The administrator guidance	AGD_OPE.1.6C - The operational user

shall describe all security requirements for the IT environment that are relevant to the administrator.	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_USR.1 (v2.3)	AGD_OPE.1 (v3.1)
AGD_USR.1.1C - The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.	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_USR.1.2C - The user guidance shall describe the use of user-accessible security functions provided by the TOE.	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_USR.1.3C - The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.	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_USR.1.4C - The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.	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.1.4C - 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_USR.1.5C - The user guidance shall be consistent with all other documentation supplied for evaluation.	AGD_OPE.1.7C - The operational user guidance shall be clear and reasonable.
AGD_USR.1.6C - The user guidance shall describe all security requirements for the IT environment that are relevant to the user.	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.

ALC_DVS.1 (v2.3)	ALC_DVS.1 (v3.1)
ALC_DVS.1.1C - The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.	ALC_DVS.1.1C - The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.
ALC_DVS.1.2C - The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.	Documentation doesn't really provide confidence that procedures were followed. This is better handled by the evaluator verifying for himself, as part of 1.2E.
ALC_DVS.1.2E - The evaluator shall confirm that the security measures are being applied.	ALC_DVS.1.2E - The evaluator shall confirm that the security measures are being applied.

ALC_FLR.2 (v2.3)	ALC_FLR.2 (v3.1) – no change
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.2.1C - 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.2.2C - 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.2.3C - 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.2.4C - 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.2.5C - 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 corrected and the correction issued to TOE users.	ALC_FLR.2.6C - The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.
ALC_FLR.2.7C - The procedures for processing reported security flaws shall provide	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.	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.	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.

ALC_LCD.1 (v2.3)	ALC_LCD.1 (v3.1) – no change
ALC_LCD.1.1C - The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.	ALC_LCD.1.1C - The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.
ALC_LCD.1.2C - The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.	ALC_LCD.1.2C - The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

ALC_TAT.1 (v2.3)	ALC_TAT.1 (v3.1)
ALC_TAT.1.1C - All development tools used for implementation shall be well-defined.	ALC_TAT.1.1C - Each development tool used for implementation shall be well-defined.
ALC_TAT.1.2C - The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.	ALC_TAT.1.2C - The documentation of each development tool shall unambiguously define the meaning of all statements as well as all conventions and directives used in the implementation.
ALC_TAT.1.3C - The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.	ALC_TAT.1.3C - The documentation of each development tool shall unambiguously define the meaning of all implementation-dependent options.

ATE_COV.2 (v2.3)	ATE_COV.2 (v3.1)
ATE_COV.2.1C - The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.	ATE_COV.2.1C - The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.
ATE_COV.2.2C - The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.	ATE_COV.2.2C -The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.
ATE_COV_EXP.2.2E - For cryptographic portions of the TOE, an NSA evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.	If PP authors need added testing of crypto, it should be done as an explicitly-stated requirement.

ATE_DPT.2 (v2.3)	ATE_DPT.3 (v3.1)
ATE_DPT.2.1C - The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and low-level design.	ATE_DPT.3.1C - The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems and modules in the TOE design. ATE_DPT.3.2C - The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested. ATE_DPT.3.3C - The analysis of the depth of testing shall demonstrate that all modules in the TOE design have been tested. Since DPT is now more closely tied to TDS, added explicit wording to ensure all subsystems/modules get tested.

ATE_FUN.1 (v2.3)	ATE_FUN.1 (v3.1)
ATE_FUN.1.1C - The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.	ATE_FUN.1.1C - The test documentation shall consist of test plans, expected test results and actual test results. The difference between test plans and test procedures was vague, so the two were combined. It's the content, not the name, that is important.
ATE_FUN.1.2C - The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.	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 test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.	
ATE_FUN.1.4C - The expected test results shall show the anticipated outputs from a successful execution of the tests.	ATE_FUN.1.3C - The expected test results shall show the anticipated outputs from a successful execution of the tests.
ATE_FUN.1.5C - The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.	ATE_FUN.1.4C - The actual test results shall be consistent with the expected test results.

ATE_IND.2 (v2.3)	ATE_IND.2 (v3.1)
ATE_IND.2.1C - The TOE shall be suitable for	ATE_IND.2.1C - The TOE shall be suitable for

testing.	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.	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.
ATE_IND.2.2E - The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.	ATE_IND.2.3E - The evaluator shall test a subset of the TSF interfaces to confirm that the TSF operates as specified.
ATE_IND.2.3E - The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.	ATE_IND.2.2E - The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

AVA_CCA_EXP (MR)	v3.1
AVA_CCA_EXP.2.1C - The analysis documentation shall identify covert channels in the cryptographic module and estimate their capacity.	The covert channel analysis was incorporated into the AVA_VAN family, as a specific kind of vulnerability. If PP authors need specifics of CCA spelled out, it should be done as an explicitly-stated requirement.
AVA_CCA_EXP.2.2C - The analysis documentation shall describe the procedures used for determining the existence of covert channels in the cryptographic module, and the information needed to carry out the covert channel analysis.	
AVA_CCA_EXP.2.3C - The analysis documentation shall describe all assumptions made during the covert channel analysis.	
AVA_CCA_EXP.2.4C - The analysis documentation shall describe the method used for estimating channel capacity, based on worst case scenarios.	
AVA_CCA_EXP.2.5C - The analysis documentation shall describe the worst case exploitation scenario for each identified covert channel.	
AVA_CCA_EXP.2.6C - The analysis documentation shall provide evidence that the method used to identify covert channels is systematic.	
AVA_CCA_EXP.2.2E - The NSA evaluator shall confirm that the results of the covert channel analysis show that the cryptographic module meets its functional requirements.	
AVA_CCA_EXP.2.3E - The NSA evaluator shall selectively validate the covert channel analysis through independent analysis and testing.	

AVA_MSU.2 (v2.3)	v3.1
AVA_MSU.1.1C - The guidance documentation 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.	The entire misuse analysis (i.e. the MSU family) was incorporated into the AGD families that address the documents subjected to such analysis. If PP authors need specific details of misuse analysis spelled out, it should be done as an explicitly-stated requirement.
AVA_MSU.1.2C - The guidance documentation shall be complete, clear, consistent and reasonable.	
AVA_MSU.1.3C - The guidance documentation shall list all assumptions about the intended environment.	
AVA_MSU.1.4C - The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).	
AVA_MSU.2.5C - The analysis documentation shall demonstrate that the guidance documentation is complete.	
AVA_MSU.2.2E - The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.	
AVA_MSU.1.3E - The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.	
AVA_MSU.2.4E - The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.	

AVA_SOF.1 (v2.3)	v3.1
AVA_SOF.1.1C - For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.	The strength-of-function analysis (i.e. the SOF family) was incorporated into the AVA_VAN family as part of the vulnerability analysis. There is no more SOF claim made in the ST.
AVA_SOF.1.2C - For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the	

specific strength of function metric defined in the PP/ST.	
AVA_SOF.1.2E - The evaluator shall confirm that the strength claims are correct.	

AVA_VLA.3 (v2.3)	AVA_VAN.4 (v2.3)
AVA_VLA.3.1C - The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.	AVA_VAN.4.1C - The TOE shall be suitable for testing. AVA_VAN.4.2E - The evaluator *shall perform* a search of public domain sources to identify potential vulnerabilities in the TOE. (The bulk of the penetration testing work is now done by the evaluator, not the developer.)
AVA_VLA.3.2C - The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.	AVA_VAN.4.3E - The evaluator *shall perform* an independent, methodical vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design, security architecture description and implementation representation to identify potential vulnerabilities in the TOE.
AVA_VLA.3.3C - The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.	
AVA_VLA.3.4C - The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.	
AVA_VLA.3.5C - The vulnerability analysis documentation shall show that the search for vulnerabilities is systematic.	
AVA_VLA.3.2E - The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.	AVA_VAN.4.4E - The evaluator shall conduct penetration testing based on the identified potential vulnerabilities to determine that the TOE is resistant to attacks performed by an attacker possessing Moderate attack potential.
AVA_VLA.3.3E - The evaluator shall perform an independent vulnerability analysis.	
AVA_VLA.3.4E - The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.	
AVA_VLA.3.5E - The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a moderate attack potential.	