

Evaluation of Cryptographic Protocols and Implementation
CCEVS Guidance
Version 1.0
5 January 2007

Purpose: This paper provides guidance on how cryptographic protocols are to be evaluated in accordance with CCEVS expectations. It is meant to ensure that CCEVS evaluations appropriately analyze and test a TOE's use of cryptographic protocols, as well as ensure evaluations of such protocols are consistent across products and CCTLs.

Background: As the submission of cryptographic enabled products increases, it is important that CCTL's evaluate products consistently and thoroughly. This includes evaluation of the implementation, integration, and potentially, the design. Cryptographic protocols typically present an interface to the TSF and offer an avenue of attack. In many cases, implementation errors are hard to detect since communicating client-server or communicating peers often use the same tools or libraries, masking implementation errors that an attacker can exploit.

Evaluations performed under CCEVS do not replace testing or requirements FIPS 140-1/2 or any guidance provided by the Cryptographic Module Validation Program (CMVP) as related to FIPS 140-1/2, the associated Derived Test Requirements, cryptographic module testing, NIST's Cryptographic Algorithm Testing Program, and other general implementation guidance. Additionally, guidance is not intended to replace other requirements NIAP may have in the area of cryptography (e.g., interpretations, specific requirements in a PP, etc.).

Evaluation:

The evaluation activities associated with assessing the compliance of a TOE's design and resulting implementation of cryptographic protocols against the claimed standard have been inconsistently applied across CCEVS evaluations, this is in part due to ill-defined specifications of the protocols. The approved list of cryptographic protocols also plays into the expectations of evaluators when evaluating a TOE's implementation of cryptographic protocols, in that CCEVS does not expect evaluators to determine the adequacy of a given protocol against replay, disclosure, modification or other attacks against the protocol. It will be assumed that these approved protocols sufficiently suit their intended purpose. The evaluators are responsible for ensuring the protocol(s) is correctly implemented, that the interfaces are implemented securely, and how keys and critical data relied upon by these protocols are initialized and that they are sufficiently protected.

The determination of whether a cryptographic protocol is correctly implemented and its components are adequately protected can be achieved through a combination of testing and analysis. These activities must be performed to achieve the same level of assurance as commensurate with the overall EAL of the TOE.

Evaluation teams are expected to identify in their evaluation workplan the methodology that will be used in evaluating the TOE's cryptographic protocol(s). CCTLs that are evaluating products that contain these protocols are required to develop a methodology for each cryptographic protocol that describes the activities the evaluation team members will perform in determining whether the implementation satisfies the SFRs. This methodology should include analysis activities (including the examination of source code when necessary), and testing activities (to include interoperability testing).

While it is hard to generalize the protocols, the following are some of the characteristics that should be addressed in the methodology with respect to implementation:

- Random seed with proper entropy
- Pseudo-randomization of the seed
- Credential exchange and verification (e.g., validation of certification path)
- Exchange of random value
- Calculation of session keys
- Verification of session keys
- Modes of operation – protocols may have different modes of operation to enhance performance (e.g., Quick Mode vs. Aggressive Mode)

It is important that an implementation has commensurate entropy for the seed; pseudo randomizes the seed, verifies the credentials properly, securely exchanges the random value for domain parameter generation, and calculates the keys correctly. As part of the standard, it is also important to examine the primality in order to understand how the numbers are generated and if they are acceptable. These issues are distinct from interoperability and need to be carefully considered in a CCEVS evaluation.

The testing of whether a vendor's implementation is compliant with a standard is an issue that is problematic, as there do not appear to be any well-defined "correct" references from which test suites could be developed. In addition to addressing the inherent security in a developer's implementation, another important aspect is whether it is interoperable with other implementations. Some aspects of testing that must be addressed in an evaluation team's methodology include:

- definition of the products used in the interoperability testing
- how they are chosen?
- how many products does a product have to interoperate with?
- what happens when a product fails to interoperate with one, but works with others?
- what if no other products implement an aspect of a standard (e.g., D-H group 14)?

There is another aspect to this that goes beyond interoperability, and that is that the common failing of "well it works, so it must be okay" mindset. Just because an implementation works with other implementations, does not mean it is secure.

Interoperability testing is important, but one needs to ensure the implementation is not flawed in other ways. The evaluation team's methodology must address how the implementation of the cryptographic protocol will be addressed in their vulnerability analysis (AVA_VLA) and (AVA_MSU)¹ activities.

Evaluation teams must perform analysis and testing of the integration to ensure the modules/algorithms are implemented securely. Analysis must also be targeted against keys and critical data that the protocols require to ensure they are initialized and are sufficiently protected. This includes areas such as:

- Trust anchors (i.e., root certificates, self-signed certificates, roots), to ensure they can not be added or modified by any one other than an authorized administrator.
- Authentication mechanisms, to ensure they cannot be bypassed or corrupted.
- Private and secret keys, to ensure are only generated and invoked by authorized users to whom they belong.
- Protections provided by the TOE for the algorithm, module, keys, or authentication data to prevent unauthorized disclosure and modification (while in storage, transit, and processing).
- Implementation of the interfaces within and at the TOE boundary.

¹ This is not meant to imply that the evaluation is limited to AVA activities. Integration of the cryptography should also be considered in other activities, where appropriate. For example, analysis could fall within ADV_ARC.