# Functional Package for Authentication Protocols

Version: 1.0
2022-01-20
**National Information Assurance Partnership**

## Revision History

| Version | Date | Comment |
|---|---|---|
| 1.0 | 2022-01-20 | Start of first draft. |

## Contents


1 Introduction
1.1 Overview
1.2 Terms
1.2.1 Common Criteria Terms
1.2.2 Technical Terms
1.3 Compliant Targets of Evaluation
1.4 Use Cases
2 Conformance Claims
3 Security Functional Requirements
3.1 Auditable Events for Mandatory SFRs
3.2 Cryptographic Support (FCS)
3.3 Identification and Authentication (FIA)
Appendix A - Implementation-Dependent Requirements
Appendix B - Use Case Templates
B.1 EAP
B.2 Pre-Shared Keys
B.3 X.509 Certificates
Appendix C - Acronyms
Appendix D - Bibliography

# 1 Introduction

## 1.1 Overview

This Functional Package for Authentication Protocols gathers requirements for several protocols and standards that are commonly used in NIAP Protection Profiles for authentication in network communications. It includes requirements for the Extensible Authentication Protocol (EAP), requirements for the composition and exchange of pre-shared keys (PSK), and requirements for X.509 certificates. These protocols and standards are often used in setting up trusted channels, but may also be used for other purposes, such as authenticating wireless communications.

More specialized protocols, such as 802.1X, RADIUS, Enrollment over Secure Transoort (EST), and Certificate Management over CMS (CMC) are not currently included.

This Functional Package is organized as a collection of requirements so that a Security Target can claim one or more "base requirements." Additional SFRs must be claimed based on dependencies in the intially claimed SFRs.

## 1.2 Terms

The following sections list Common Criteria and technology terms used in this document.

### 1.2.1 Common Criteria Terms

| | |
|---|---|
| Assurance | Grounds for confidence that a TOE meets the SFRs [CC]. |
| Base Protection Profile (Base-PP) | Protection Profile used as a basis to build a PP-Configuration. |
| Collaborative Protection Profile (cPP) | A Protection Profile developed by international technical communities and approved by multiple schemes |
| Common Criteria (CC) | Common Criteria for Information Technology Security Evaluation (International Standard ISO/IEC 15408). |
| Common Criteria Testing Laboratory | Within the context of the Common Criteria Evaluation and Validation Scheme (CCEVS), an IT security evaluation facility, accredited by the National Voluntary Laboratory Accreditation Program (NVLAP) and approved by the NIAP Validation Body to conduct Common Criteria-based evaluations. |
| Common Evaluation Methodology (CEM) | Common Evaluation Methodology for Information Technology Security Evaluation. |
| Distributed TOE | A TOE composed of multiple components operating as a logical whole. |
| Extended Package (EP) | A deprecated document form for collecting SFRs that implement a particular protocol, technology, or functionality. See Functional Packages. |
| Functional Package (FP) | A document that collects SFRs for a particular protocol, technology, or functionality. |
| Operational Environment (OE) | Hardware and software that are outside the TOE boundary that support the TOE functionality and security policy. |
| Protection Profile (PP) | An implementation-independent set of security requirements for a category of products. |
| Protection Profile Configuration (PP-Configuration) | A comprehensive set of security requirements for a product type that consists of at least one Base-PP and at least one PP-Module. |
| Protection Profile Module (PP-Module) | An implementation-independent statement of security needs for a TOE type complementary to one or more Base Protection Profiles. |

| | |
|---|---|
| Security Assurance Requirement (SAR) | A requirement to assure the security of the TOE. |
| Security Functional Requirement (SFR) | A requirement for security enforcement by the TOE. |
| Security Target (ST) | A set of implementation-dependent security requirements for a specific product. |
| Target of Evaluation (TOE) | The product under evaluation. |
| TOE Security Functionality (TSF) | The security functionality of the product under evaluation. |
| TOE Summary Specification (TSS) | A description of how a TOE satisfies the SFRs in an ST. |

### 1.2.2 Technical Terms

| | |
|---|---|
| Certificate Authority (CA) | An entity that issues digital certificates. |
| Certificate Signing Request (CSR) | A message sent from an applicant to a registration authority of the public key infrastructure in order to apply for a digital identity certificate. |
| Distinguished Name (DN) | A field in an X.509 certificate that uniquely identifies a person, organization, or business. |
| Eliptic Curve group modulo a Prime (ECP) | Eliptic Curve Group Modulo a Prime. |
| Encapsulating Security Payload (ESP) | The protocol used by IPsec to transport encrypted and integrity-protected communications across the network. |
| Extended Authentication (XAUTH) | An authentication scheme that supports an additional level of authentication by allowing the IPSec gateway to request extended authentication from remote users. |
| Extended Sequence Number (ESN) | An extension to the standard that allows IPsec to use 64-bit sequence numbers. |
| Extensible Authentication Protocol (EAP) | A framework for adding arbitrary authentication methods in a standardized way to any protocol. The most common EAP method used with IKEv2 is EAP-TLS. |
| Fully Qualified Domain Name (FQDN) | A domain name that specifies its exact location in the hierarchy of the Domain Name System (DNS). |
| Internet Control Message Protocol (ICMP) | A supporting protocol in the Internet Protocol suite. It is used by network devices to send error messages and operational information indicating success or failure when communicating with another IP address. |
| Internet Key Exchange (IKE) | The protocol used by IPsec to set up and manage IPsec connections. This includes negotiating IPsec connection settings, authenticating endpoints to each other, defining the security parameters of IPsec-protected connections, and negotiating session keys. IKEv2 is the current version. |
| Internet Protocol Security (IPsec) | A suite of open standards for ensuring private communications over public networks. |
| Internet Security Association and Key Management Protocol (ISAKMP) | A protocol defined by RFC 2408 for establishing Security association (SA) and cryptographic keys in an Internet environment. |

| | |
|---|---|
| Online Certificate Status Protocol (OCSP) | An Internet protocol used for obtaining the revocation status of an X.509 digital certificate. |
| Pre-Shared Key (PSK) | A secret that was previously shared between two parties before it needs to be used. |
| Security Association (SA) | The establishment of shared security attributes between two network entities to support secure communication. |
| Security Policy Database (SPD) | A set of rules that determines whether a packet is subject to IPsec processing. Each entry in the SPD represents a policy that defines how the set of traffic covered under the policy will be processed. |
| User Datagram Protocol (UDP) | A communications protocol that is primarily used to establish low-latency and loss-tolerating connections between applications on the internet. |
| Virtual Private Network (VPN) | An extension of a private network across a public or shared network that allows users to exchange data as though they were connected directly to the private network. |

## 1.3 Compliant Targets of Evaluation

The Target of Evaluation (TOE) in this Functional Package (FP) is an IT product that uses pre-shared keys, X.509 certificates, or the Extensible Authentication Protocol (EAP). Typically these protocols would be used in products that implement trusted channels or VPNs for communications with other IT entities, or implement wireless security protocols. This FP describes the security functionality of these protocols and standards in terms of [CC].

The contents of this FP must be appropriately incorporated into a PP, cPP, or PP-Module. When this Package is so incorporated, the ST must include selection-based requirements in accordance with the selections or assignments indicated in the incorporating document.

The PP, cPP, or PP-Module that instantiates this Package must typically include the below components in order to satisfy dependencies of this Package. It is the responsibility of the PP, cPP, or PP-Module author who incorporates this FP to ensure that dependence on these components is satisfied, either by the TOE or by assumptions about its Operational Environment.

An ST must identify the applicable version of the PP, cPP, or PP-Module and this Functional Package in its conformance claims.

| Component | Explanation |
|---|---|
| FCS_CKM.1 | To support key generation for IPsec, the incorporating document must include FCS_CKM.1 and specify the corresponding algorithms. |
| FCS_CKM.2 | To support key establishment for IPsec, the incorporating document must include FCS_CKM.2 and specify the corresponding algorithms. |
| FCS_CKM_EXT.5 | To support key derivation for IPsec, the incorporating document may need to include FCS_CKM_EXT.5 and specify the corresponding key derivation algorithms. |
| FCS_COP.1 | To support the cryptography needed for IPsec communications, the incorporating document must include FCS_COP.1 (iterating as needed) to specify AES with corresponding key sizes and modes, digital signature generation and verification function (at least one of RSA or ECDSA), a cryptographic hash function, and a keyed-hash message authentication function. In particular, this Package requires that the TOE support AES-GCM-128 and AES-GCM-256 for ESP, and AES-CBC-128 and AES-CBC-256 for IKE. |
| FCS_RBG_EXT.1 | To support random bit generation needed for IPsec key generation, the incorporating document must include a requirement that specifies the TOE's ability to invoke or provide random bit generation services, commonly identified as FCS_RBG_EXT.1. |
| FIA_X509_EXT.2 | To support establishment of IPsec communications using a public key algorithm that includes X.509, the incorporating document must include FIA_X509_EXT.2 to specify the reasons for using X.509 certificates. But is this really a dependency for this package? I don't think so. |
| FPT_STM.1 | To support establishment of IPSec communications using a public key algorithm that includes X.509, the incorporating document must include FPT_STM.1 or some other requirement that ensures reliable system time. Note however that support for time-based rekey thresholds is selectable and not mandatory. |

## 1.4 Use Cases

Although focused on IPsec, this Package contains requirements for several authentication protocols used by IPsec IKE protocols. If the TOE does not implement IPsec, but it does implement one of the other protocols, then this Package can be used to integrate requirements for the other protocols into the ST.

**[USE CASE 1] EAP**

This use case adds physical protections to the base requirements for server-class hardware. Additional physical protections are required because the platform us assumed to be minimally protected by the by the operational environment. This use case can also be invoked for servers in data centers where there are enhanced security requirements.

This use case adds requirements for audit, physical protections, and Administrator authentication to the base mandatory SFRs. It removes the assumption that the TOE is physically protected by the OE.

For changes to included SFRs, selections, and assignments required for this use case, see B.1 EAP.

**[USE CASE 2] Pre-Shared Keys**

This use case defines the base requirements for portable clients.

This use case adds no requirements to the base mandatory SFRs.

For changes to included SFRs, selections, and assignments required for this use case, see B.2 Pre-Shared Keys.

**[USE CASE 3] X.509 Certificates**

This use case defines the base requirements for portable clients.

This use case adds no requirements to the base mandatory SFRs.

For changes to included SFRs, selections, and assignments required for this use case, see B.3 X.509 Certificates.

# 2 Conformance Claims

**Conformance Statement**

An ST must claim exact conformance to this Package, as defined in the CC and CEM addenda for Exact Conformance, Selection-Based SFRs, and Optional SFRs (dated May 2017).

**CC Conformance Claims**

This Package is conformant to Parts 2 (extended) and 3 (conformant) of Common Criteria Version 3.1, Revision 5.

**PP Claim**

This Package does not claim conformance to any Protection Profile.

**Package Claim**

This Package does not claim conformance to any packages.

# 3 Security Functional Requirements

This chapter describes the security requirements which have to be fulfilled by the product under evaluation. Those requirements comprise functional components from Part 2 of [CC]. The following conventions are used for the completion of operations:

- **Refinement** operation (denoted by **bold text** or ~~strikethrough text~~): is used to add details to a requirement (including replacing an assignment with a more restrictive selection) or to remove part of the requirement that is made irrelevant through the completion of another operation, and thus further restricts a requirement.
- **Selection** (denoted by *italicized text*): is used to select one or more options provided by the [CC] in stating a requirement.
- **Assignment** operation (denoted by *italicized text*): is used to assign a specific value to an unspecified parameter, such as the length of a password. Showing the value in square brackets indicates assignment.
- **Iteration** operation: is indicated by appending the SFR name with a slash and unique identifier suggesting the purpose of the operation, e.g. "/EXAMPLE1."

## 3.1 Auditable Events for Mandatory SFRs

The auditable events specified in this Package are included in a Security Target if the incorporating PP, cPP, or PP-Module supports audit event reporting through FAU_GEN.1 and all other criteria in the incorporating document are met.

**Table 1: Auditable Events for Mandatory Requirements**

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FCS_EAP_EXT.1 | No events specified. | N/A |
| FIA_PSK_EXT.1 | No events specified. | N/A |

## 3.2 Cryptographic Support (FCS)

### FCS_EAP_EXT.1 EAP-TLS

FCS_EAP_EXT.1.1

The TSF shall implement [**selection**: *EAP-TLS protocol as specified in [RFC 5216]*, *EAP-TTLS as specified in [RFC 5281]*] as updated by [RFC 8996] with TLS implemented using mutual authentication in accordance with the [Functional Package for TLS].

FCS_EAP_EXT.1.2

The TSF shall generate random values used in the [**selection**: *EAP-TLS*, *EAP-TTLS*] exchange using the RBG specified in FCS_RBG_EXT.1.

FCS_EAP_EXT.1.3

The TSF shall support peer authentication using certificates and [**selection**: *PSK*, *HOTP*, *TOTP*, *[**assignment**: other Authentication-verification protocols]*, *no other authentication*] as updated by [RFC 8996] with TLS implemented using mutual authentication in accordance with the [Functional Package for TLS].

FCS_EAP_EXT.1.4

The TSF shall not forward a EAP-success response if the client certificate is not valid according to FIA_X509_EXT.1.

FCS_EAP_EXT.1.5

The TSF shall use the MSK from the [**selection**: *EAP-TLS*, *EAP-TTLS*] response as the IKEv2 shared secret in the authentication payload.

**Evaluation Activities**

*FCS_EAP_EXT.1*
***TSS***
*The evaluator shall verify that the TSS describes the use of EAP options for each of the selected peer authentication mechanisms, that TLS with mutual authentication is used, that the random values are from an appropriate source, and that the EAP MSK is derived from the TLS master key and is used as the IKEv2 shared key.*
***Guidance***
*The evaluator shall verify that the guidance documents describe any configurable features of the EAP or TLS functionality, including instructions for configuration of the authenticators and registration processes for clients.*

***Tests***
*Testing for TLS functionality is in accordance with the TLS package.*

*For each supported EAP method claimed in FCS_EAP_EXT.1.1 and for each authentication method claimed in FCS_EAP_EXT.1.3, the evaluator shall perform the following tests:*

- ***Test 1:*** *The evaluator shall follow AGD guidance to configure the TSF to use the EAP method claimed. The evaluator shall follow AGD guidance to configure the TSF to use the authentication method claimed and, for EAP-TTLS, register an endpoint with appropriate key material required for the authentication method. The evaluator shall establish an IPsec connection using a test endpoint with a valid certificate and, for EAP-TTLS, configured to provide a correct value for the configured authenticator. The evaluator shall observe that the IPsec connection is successful.*
- ***Test 2:*** *[conditional] If EAP-TTLS is supported, the evaluator shall cause the test endpoint with a valid certificate to send an invalid authenticator for the claimed authentication method:*
  - *For HOTP, replay the HOTP value sent previously.*
  - *For TOTP or PSK, modify a byte of the properly constructed value,and observe that the TSF aborts the connection.*
- ***Test 3:*** *The evaluator shall establish a new, valid certificate for a test endpoint using an identifier not corresponding to a registered user. For EAP-TTLS, the evaluator shall cause the test endpoint using this certificate to send a correct authenticator value for the registered user. The evaluator shall initiate an IPsec connection from the test endpoint to the TSF and observe that the TSF aborts the connection.*
- ***Test 4:*** *The evaluator shall follow AGD guidance to configure the TSF to use a supported EAP method and register the user with key material for required for a supported authentication method. The evaluator shall configure a test endpoint to respond to an IKEv2 exchange with EAP-request, providing valid phase 1 handshake and valid TLS handshake, but computing the phase 2 shared key using standard (non-EAP) methods. The evaluator shall initiate an IPsec connection between the test endpoint and the TSF, and observe that the TSF aborts the connection.*

## 3.3 Identification and Authentication (FIA)

The TOE may support pre-shared keys for use in establishing an IPsec connection. PSK in the context of this document refer to generated values, memorized values subject to conditioning, one time passwords, and combinations of the above as described in FIA_PSK_EXT.1.2.

### FIA_HOTP_EXT.1 HMAC-Based One-Time Password Pre-Shared Keys

***This is a selection-based component. Its inclusion depends upon selection from FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2, FCS_EAP_EXT.1.3.***

FIA_HOTP_EXT.1.1

The TSF shall support HMAC-Based One-Time Password authentication (HOTP) in accordance with [RFC 4226] to authenticate the user before establishing an IPsec connection.

FIA_HOTP_EXT.1.2

The TSF shall generate a HOTP seed of [**selection**: *128*, *256*] bits in accordance with FCS_RBG_EXT.1.

FIA_HOTP_EXT.1.3

The TSF shall generate a new HOTP seed value for each IPsec connection.

FIA_HOTP_EXT.1.4

The TSF shall utilize [**selection**: *HMAC-SHA-1*, *HMAC-SHA-256*, *HMAC-SHA-384*, *HMAC-SHA-512*] with key sizes [**assignment**: *key size (in bits) used in HMAC*] to derive a HOTP hash from the HOTP seed and counter.

**Application Note:** The assignment must be consistent with the key sizes supported by the relevant iteration of FCS_COP.1.

FIA_HOTP_EXT.1.5

The TSF shall truncate the HOTP hash per FIA_HOTP_EXT.1.4 to create a HOTP of [**selection, choose one of**:

- *administrator configurable character length of at least 6*,
- *preset character length of [**selection, choose one of**: 6, 7, 8, 9, 10]*

].

**Application Note:** The ST author may either provide a configurable character length of at least 6 or a preset size between 6 and 10.

FIA_HOTP_EXT.1.6

The TSF shall [**selection**:

- *throttle invalid requests to [**selection**: administrator configurable value, [**assignment**: value less than 10]] per minute ,*
- *lock the associated account after [**selection**: administrator configurable value, [**assignment**: value less than 10]] failed attempts until [**selection**: an administrator unlocks the account, a configurable time period]*

].

**Application Note:** The ST author may select throttle requests, account lockout, or both.

FIA_HOTP_EXT.1.7

The TSF shall not verify HOTP attempts outside of the counter look ahead window of [**selection**: *a configurable value, [**assignment**: a value less than or equal to 3]*] for resynchronization.

FIA_HOTP_EXT.1.8

The TSF shall increment the counter after each successful authentication.

**Application Note:** The HOTP seed and all derived values are considered secret keys for purposes of protection. This requirement must be claimed if "*verify the HOTP*" is selected in FIA_PSK_EXT.4.2.

### Evaluation Activities

*FIA_HOTP_EXT.1*
***TSS***
*The evaluator shall confirm the TSS describes how the TOE complies with [RFC 4226].*

*The evaluator shall confirm the TSS describes how the HOTP seed is generated and ensure it aligns with FCS_RBG_EXT.1.*

*The evaluator shall confirm the TSS describes how the HOTP seed is protected and ensure it aligns with the storage requirements of the base PP.*

*The evaluator shall confirm the TSS describes how a new HOTP seed is assigned for each client and how each client is uniquely identified.*

*The evaluator shall confirm the TSS describes how the HOTP seed is conditioned into a HOTP hash and verify it matches the selection in FIA_HOTP_EXT.1.4.*

*The evaluator shall confirm the TSS describes how the HOTP hash is truncated and verify it matches the selection in FIA_HOTP_EXT.1.5.*

*The evaluator shall confirm the TSS describes how the TOE handles multiple incoming invalid requests and verify it provides anti-hammer mechanism that match the selections FIA_HOTP_EXT.1.6.*

*The evaluator shall confirm the TSS describes how the TOE handles resynchronization and how it rejects attempts outside of the look-ahead window selected in FIA_TOTP_EXT.1.7.*

*The evaluator shall confirm the TSS describes how the TOE how the counter is incremented after each successful authentication.*

***Guidance***
*The evaluator shall verify the operational guidance contains all configuration guidance for setting any administrative value that is configurable in the FIA_HOTP_EXT.1 requirements.*
***Tests***
*The evaluator shall configure the TOE to use a supported HOTP factor then:*

- ***Test 1:** Attempt to establish a connection using a factor from a different client, the test passes if the client fails to connect.*
- ***Test 2:** Attempt multiple connections outside the boundary set in FIA_HOTP_EXT.1.6 and verify the remediation is triggered. The test passes if remediation is triggered as defined in the selections and assignments.*
- ***Test 3:** Attempt to use a HOTP that is outside of the value allowed with for resynchronization. The test passes if the client fails to connect.*
- ***Test 4:** Attempt to connect with a valid HOTP, disconnect and attempt to authenticate again with the same HOTP value. The test passes if the client connects the first time and fails to connect the second time. If the HOTP generated is duplicated the test may be repeated.*

## FIA_PSK_EXT.1 Pre-Shared Key Composition

FIA_PSK_EXT.1.1

The TSF shall be capable of using pre-shared keys for establishing IPsec

connections.

FIA_PSK_EXT.1.2

The TSF shall be able to accept the following as pre-shared keys: [**selection**: *Generated bit-based*, *Password-based*, *HMAC based one time password*, *Time based one time password*, *Combination of a generated bit-based and HMAC-based one-time password*, *Combination of a generated bit-based and time-based one-time password*, *Combination of a password-based and HMAC-based one-time password*, *Combination of a password-based and time-based one-time password*] keys.

**Application Note:** If any selection including "*generated bit-based*" keys is selected, then FIA_PSK_EXT.2 must be claimed.

If any selection including "*password-based*" keys is selected then FIA_PSK_EXT.3 must be claimed.

If any selection including "*HMAC-based one-time password*" keys is selected then FIA_PSK_EXT.4 must be claimed.

If any selection including "*time-based one-time password*" is selected then FIA_PSK_EXT.5 must be claimed.

This requirement must be claimed if "*Pre-shared keys*" is selected in FCS_IPSEC_EXT.1.11.

**Evaluation Activities**

*FIA_PSK_EXT.1*
***TSS***
*The evaluator shall examine the TSS to ensure that it identifies that IPsec connections may use pre-shared keys. The evaluator shall confirm that the TSS states which pre-shared key selections are supported.*
***Guidance***
*The evaluator shall examine the operational guidance to determine that it provides guidance to administrators on how to configure all selected pre-shared key options if any configuration is required.*
***Tests***
*The evaluator shall also perform the following tests for each protocol (or instantiation of a protocol, if performed by a different implementation on the TOE).*

- ***Test 1:*** *For each mechanism selected in FIA_PSK_EXT.1.2 the evaluator shall attempt to establish a connection and confirm that the connection requires the selected factors in the PSK to establish the connection.*

### FIA_PSK_EXT.2 Generated Pre-Shared Keys

***This is a selection-based component. Its inclusion depends upon selection from FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2.***

FIA_PSK_EXT.2.1

The TSF shall be able to [**selection**:

- *accept externally generated*,
- *generate [**selection**: 128, 256] bit-based pre-shared keys via FCS_RBG_EXT.1.*

]

**Application Note:** Generated PSKs are expected to be shared between components via an out of band mechanism. This requirement must be claimed if any selection in FIA_PSK_EXT.1.2 includes "*generated bit-based*" keys.

**Evaluation Activities**

*FIA_PSK_EXT.2*
***TSS***
*If "generate" is selected the evaluator shall confirm that this process uses the RBG specified in FCS_RBG_EXT.1 and the output matches the size selected in FIA_PSK_EXT.2.1.*

***Guidance***
*The evaluator shall confirm the operational guidance contains instructions for entering*

*generated pre-shared keys.*
***Tests***

- ***Test 1:*** *[conditional] If "generate" was selected the evaluator shall generate a pre-shared key and confirm the output matches the size selected in FIA_PSK_EXT.2.1.*

### FIA_PSK_EXT.3 Password-Based Pre-Shared Keys

***This is a selection-based component. Its inclusion depends upon selection from , FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2.***

FIA_PSK_EXT.3.1

The TSF shall support a PSK of up to [**assignment**: *positive integer of 64 or more*] characters.

**Application Note:** The ST author assigns the maximum size of the PSK it supports; it must support at least 64 characters or a length defined by the platform.

FIA_PSK_EXT.3.2

The TSF shall allow PSKs to be composed of any combination of upper case characters, lower case characters, numbers, and the following special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")", and [**selection**: *[**assignment**: other supported special characters], no other characters*]

**Application Note:** The ST author assigns any other supported characters; if there are no other supported characters, then "*no other characters*" should be selected.

FIA_PSK_EXT.3.3

The TSF shall perform Password-based Key Derivation Functions in accordance with a specified cryptographic algorithm [**selection**: *HMAC-SHA-256*, *HMAC-SHA-384*, *HMAC-SHA-512*], with [**assignment**: *positive integer of 4096 or more*] iterations], and output cryptographic key sizes [**selection**: *128*, *256*] that meet the following: [NIST SP 800-132 Part 1].

**Application Note:** The ST author selects the parameters based on the PBKDF used by the TSF.

FIA_PSK_EXT.3.4

The TSF shall not accept PSKs less than [**selection**: *a value settable by the administrator*, *[**assignment**: minimum PSK length accepted by the TOE, must be >= 6]*] and greater than the maximum PSK length defined in FIA_PSK_EXT.3.1.

**Application Note:** If the minimum length is settable, then ST author chooses "*a value settable by the administrator*". If the minimum length is not settable, the ST author fills in the assignment with the minimum length the PSK must be. This requirement is to ensure bounds work properly.

FIA_PSK_EXT.3.5

The TSF shall generate all salts using an RBG that meets FCS_RBG_EXT.1 and with entropy corresponding to the key size selected for PBKDF in FIA_PSK_EXT.3.3.

FIA_PSK_EXT.3.6

The TSF shall require the PSK to be entered before every initiated connection.

FIA_PSK_EXT.3.7

The TSF shall [**selection**: *provide a password strength meter*, *check the password against a blacklist*, *perform no action to assist the user in choosing a strong password*].

**Application Note:** For FIA_PSK_EXT.3.7, the ST author may select one, both, or neither of the functions in alignment with [NIST SP 800-63B].

This requirement is selection dependent on FIA_PSK_EXT.1.

### Evaluation Activities

*FIA_PSK_EXT.3*
***TSS***
*The evaluator shall examine the TSS to ensure it describes the process by which the bit-based*

*pre-shared keys used. If generated is selected the evaluator shall confirm that this process uses the RBG specified in FCS_RBG_EXT.1.*

*Support for length: The evaluators shall check to ensure that the TSS describes the allowable ranges for PSK lengths, and that at least 64 characters or a length defined by the platform may be specified by the user.*

*Support for character set: The evaluator shall check to ensure that the TSS describes the allowable character set and that it contains the characters listed in the SFR.*

*Support for PBKDF: The evaluator shall examine the TSS to ensure that the use of PBKDF2 is described and that the key sizes match that described by the ST author.*

*The evaluator shall check that the TSS describes the method by which the PSK is first encoded and then fed to the hash algorithm. The settings for the algorithm (padding, blocking, etc.) shall be described, and the evaluator shall verify that these are supported by the selections in this component as well as the selections concerning the hash function itself.*

*For the NIST SP 800-132-based conditioning of the PSK, the required evaluation activities will be performed when doing the evaluation activities for the appropriate requirements (e.g., FCS_COP.1/KeyedHash).*

*The evaluator shall confirm that the minimum length is described.*

*The ST author shall provide a description in the TSS regarding the salt generation. The evaluator shall confirm that the salt is generated using an RBG described in FCS_RBG_EXT.1.*

*[conditional] If "password strength meter" or "password blacklist" is selected, the evaluator shall examine the TSS to ensure any password checking functionality provided by the TSF is described and contains details on how the function operates.*

***Guidance***
*The evaluator shall confirm the operational guidance contains instructions for entering bit-based pre-shared keys for each protocol identified in the requirement, or generating a bit-based pre-shared key (or both). The evaluator shall confirm that any configuration management functions related to pre-shared keys that are performed by the TOE are specified in the operational guidance.*

*The guidance must specify the allowable characters for pre-shared keys, and that list must include, at minimum, the same items contained in FIA_PSK_EXT.3.2.*

*The evaluator shall confirm the operational guidance contains any necessary instructions for enabling and configurating password checking functionality.*

***Tests***
*Support for Password/Passphrase characteristics: In addition to the analysis above, the evaluator shall also perform the following tests on a TOE configured according to the Operational Guidance:*

- ***Test 1:** The evaluator shall compose a pre-shared key of at least 64 characters that contains a combination of the allowed characters in accordance with the FIA_PSK_EXT.1.3 and verify that a successful protocol negotiation can be performed with the key.*
- ***Test 2:** [conditional] If the TOE supports pre-shared keys of multiple lengths, the evaluator shall repeat Test 1 using the minimum length and invalid lengths that are below the minimum length, above the maximum length, null length, empty length, or zero length. The minimum test should be successful, and the invalid lengths must be rejected by the TOE.*
- ***Test 3:** [conditional] If the TOE initiates connections, initiate and establish a remote connection, disconnect from the connection, verify that the PSK is required when initiating the connection a second time.*
- ***Test 4:** [conditional] If the TOE supports a password meter, the evaluator shall enter a password and verify the password checker responds per the description in the TSS.*
- ***Test 5:** [conditional] If the TOE supports a password blacklist, the evaluator shall enter a blacklisted password and verify that the password is rejected or flagged as such.*

### FIA_PSK_EXT.4 HMAC-Based One-Time Password Pre-shared Keys Support

***This is a selection-based component. Its inclusion depends upon selection from FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2.***

FIA_PSK_EXT.4.1

The TSF shall accept and send a HOTP while initiating an IPsec connection.

FIA_PSK_EXT.4.2

The TSF shall [**selection, choose one of**: *verify the HOTP*, *verify the HOTP via an external authentication server*] before establishing an incoming connection.

**Application Note:** This requirement must be claimed if "*HMAC-based one-time password*" is included in a selection in FIA_PSK_EXT.1.2.

**Evaluation Activities**

*FIA_PSK_EXT.4*
***TSS***
*The evaluator shall verify the TSS describes how the HOTP is input into the client and how that value is sent to the server.*

*The evaluator shall verify the TSS describes how the HOTP is accepted from an incoming connection and how that value is verified, either by the TOE or by an external authentication server.*

***Guidance***
*The evaluator shall verify the operational guidance contains any configuration necessary to enable HOTP.*

***Tests***

- ***Test 1:*** *The evaluator shall configure the TOE to use a supported HOTP factor, then attempt to establish a connection using that factor.*
- ***Test 2:*** *The evaluator shall verify the client prompts the user for the HOTP before initiating the connection.*
- ***Test 3:*** *The evaluator shall verify the server validates the HOTP or receives confirmation from an authentication server before establishing the channel.*

### FIA_PSK_EXT.5 Time Based One Time Password Pre-shared Keys Support

***This is a selection-based component. Its inclusion depends upon selection from FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2.***

FIA_PSK_EXT.5.1

The TSF shall accept and send a TOTP while initiating an IPsec connection.

FIA_PSK_EXT.5.2

The TSF shall [**selection, choose one of**: *verify the TOTP*, *verify the TOTP via an external authentication server*] before establishing an incoming connection.

**Application Note:** If "*verify the TOTP*" is selected then FIA_TOTP_EXT.1 must be claimed.

This requirement must be claimed if "*Time-based one-time password*" is included in a selection in FIA_PSK_EXT.1.2.

**Evaluation Activities**

*FIA_PSK_EXT.5*
***TSS***
*The evaluator shall verify the TSS describes how the TOTP is input into the client and how that value is sent to the server.*

*The evaluator shall verify the TSS describes how the TOTP is accepted from an incoming connection and how that value is verified, either by the TOE or by an external authentication server.*

***Guidance***
*The evaluator shall verify the operational guidance contains any configuration necessary to enable TOTP.*
***Tests***

- ***Test 1:*** *The evaluator shall configure the TOE to use a supported TOTP factor, then attempt to establish a connection using that factor.*
- ***Test 2:*** *The evaluator shall verify the client prompts the user for the TOTP before initiating the connection.*
- ***Test 3:*** *The evaluator shall verify the server validates the TOTP or receives confirmation from an authentication server before establishing the channel.*

### FIA_TOTP_EXT.1 Time-Based One-Time Password Pre-Shared Keys

***This is a selection-based component. Its inclusion depends upon selection from FCS_EAP_EXT.1.3, FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2.***

FIA_TOTP_EXT.1.1

The TSF shall support Time-Based One-Time Password (TOTP) authentication in accordance with [RFC 6238] to authenticate the user before establishing an IPsec connection.

FIA_TOTP_EXT.1.2

The TSF shall generate a TOTP seed according to FCS_RBG_EXT.1 of [**selection**: *128*, *256*] bits.

FIA_TOTP_EXT.1.3

The TSF shall generate a new TOTP seed for each IPsec connection.

FIA_TOTP_EXT.1.4

The TSF shall utilize [**selection**: *HMAC-SHA-1*, *HMAC-SHA-256*, *HMAC-SHA-384*, *HMAC-SHA-512*] with key sizes [**assignment**: *key size (in bits) used in HMAC*] to derive a TOTP hash from the TOTP seed and current time provided by NTP.

**Application Note:** The selection must be consistent with the key sizes permitted by the relevant iteraton of FCS_COP.1.

FIA_TOTP_EXT.1.5

The TSF shall truncate the TOTP hash per FIA_TOTP_EXT.1.4 to create a TOTP of [**selection**:

- *administrator configurable character length of at least 6*,
- *preset character length of [**selection, choose one of**: 6, 7, 8, 9, 10]*

].

**Application Note:** The ST author may either provide a configurable character length of at least 6 or a preset size between 6 and 10.

FIA_TOTP_EXT.1.6

The TSF shall [**selection**:

- *throttle invalid requests to [**selection**: administrator configurable value, [**assignment**: value less than 10]] per minute*,
- *lock the associated account after [**selection**: administrator configurable value, [**assignment**: value less than 10]] failed attempts until [**selection**: an administrator unlocks the account, a configurable time period]*

].

**Application Note:** The ST may select throttle requests, account lockout, or both.

FIA_TOTP_EXT.1.7

The TSF shall set a time-step size of [**selection**: *a configurable value*, *[**assignment**: a value less than or equal to 30]*] seconds.

FIA_TOTP_EXT.1.8

The TSF shall not validate a drift of more than [**selection**: *a configurable value*, *[**assignment**: a value less than or equal to 3]*] time-steps.

FIA_TOTP_EXT.1.9

The TSF shall [**selection, choose one of**: *allow resynchronization by recording time drift within the limit of FIA_TOTP_EXT.2.8*, *not permit resynchronization*].

**Application Note:** The TOTP seed and all derived values are considered secret keys for purposes of protection. This requirement must be claimed if "verify the TOTP" is selected in FCS_PSK_EXT.5.2.

### Evaluation Activities

*FIA_TOTP_EXT.1*
***TSS***
*The evaluator shall confirm the TSS describes how the TOE complies with [RFC 6238]*

*. The evaluator shall confirm the TSS describes how the TOTP seed is generated and ensure it aligns with FCS_RBG_EXT.1.*

*The evaluator shall confirm the TSS describes how the TOTP seed is protected and ensure it aligns with the storage requirements of the base PP.*

*The evaluator shall confirm the TSS describes how a new TOTP seed is assigned for each client and how each client is uniquely identified.*

*The evaluator shall confirm the TSS describes how the TOTP seed is conditioned into a TOTP hash and verify it matches the selection in FIA_TOTP_EXT.1.4.*

*The evaluator shall confirm the TSS describes how the TOTP hash is truncated and verify it matches the selection in FIA_TOTP_EXT.1.5.*

*The evaluator shall confirm the TSS describes how the TOE handles multiple incoming requests and verify it provides anti-hammer mechanism that matches the selections FIA_TOTP_EXT.2.6.*

*The evaluator shall confirm the TSS describes how the TOE sets a time-step value and verify it matches the selections in the ST.*

*The evaluator shall confirm the TSS describes how the TOE handles drift and resynchronization and verify it matches the selections. The evaluator shall ensure the TSS describes how time is kept and drift is calculated. If drift is recorded the evaluator shall ensure the TSS how this is done.*

***Guidance***
*The evaluator shall verify the operational guidance contains all configuration guidance for setting any administrative value that is configurable in the FIA_TOTP_EXT.1 requirements.*
***Tests***
*The evaluator shall configure the TOE to use a supported TOTP factor then:*

- ***Test 1:*** *Attempt to establish a connection using a factor from a different client, the test passes if the client fails to connect.*
- ***Test 2:*** *Attempt multiple connections outside the boundary set in FIA_TOTP_EXT.1.6 and verify the remediation is triggered. The test passes if remediation is triggered as defined in the selections and assignments.*
- ***Test 3:*** *Attempt to use a TOTP that is outside of the value allowed with for resynchronization. The test passes if the client fails to connect.*
- ***Test 4:*** *Attempt to connect with a valid TOTP, disconnect and attempt to authenticate again with the same TOTP. The test passes if the client connects the first time and fails to connect the second time. If the TOTP generated is duplicated the test may be repeated.*

### FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1

The application shall [**selection, choose one of**: *invoke platform-provided functionality*, *implement functionality*] to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The application shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The application shall validate that any CA certificate includes caSigning purpose in the key usage field
- The application shall validate the revocation status of the certificate using [**selection**: *OCSP as specified in RFC 6960*, *CRL as specified in RFC 5280 Section 6.3*, *CRL as specified in RFC 8603*, *an OCSP TLS Status Request Extension (OCSP stapling) as specified in RFC 6066*, *OCSP TLS Multi-Certificate Status Request Extension (i.e., OCSP Multi-Stapling) as specified in RFC 6961*].
- The application shall validate the extendedKeyUsage (EKU) field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing Purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the EKU field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
  - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
  - Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field.

**Application Note:** FIA_X509_EXT.1.1 lists the rules for validating certificates. The ST author shall select whether revocation status is verified using OCSP or CRLs. FIA_X509_EXT.2 requires that certificates are used for HTTPS, TLS, and DTLS; this use requires that the extendedKeyUsage rules are verified. If OCSP is not supported the EKU provision for checking the OCSP Signing purpose is met by default.

This requirement is included if the protocol(s) selected in FTP_DIT_EXT.1.1 require the use of certificates. If the TOE implements TLS as a HTTPS/TLS server with no mutual authentication, this requirement is not applicable.

OCSP stapling and OCSP multi-stapling only support TLS server certificate validation. If other certificate types are validated, either OCSP or CRL should be claimed.

Regardless of the selection of "*implement functionality or invoke platform-provided functionality*," the validation is expected to end in a trusted root CA certificate in a root store managed by the platform.

FIA_X509_EXT.1.2

The application shall treat a certificate as a CA certificate only if the basicConstraints extension is present and the CA flag is set to TRUE.

**Application Note:** This requirement applies to certificates that are used and processed by the TSF and restricts the certificates that may be added as trusted CA certificates.

**Evaluation Activities**

*FIA_X509_EXT.1.1*
***TSS***
*The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place. The evaluator ensures the TSS also provides a description of the certificate path validation algorithm.*

***Guidance***
*None.*

***Tests***
*The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created.*

- ***Test 1:*** *The evaluator shall demonstrate that validating a certificate without a valid certification path results in the function failing, for each of the following reasons, in turn:*
  - *by establishing a certificate path in which one of the issuing certificates is not a CA certificate,*
  - *by omitting the basicConstraints field in one of the issuing certificates,*
  - *by setting the basicConstraints field in an issuing certificate to have CA=False,*
  - *by omitting the CA signing bit of the key usage field in an issuing certificate, and*
  - *by setting the path length field of a valid CA field to a value strictly less than the certificate path.*

  *The evaluator shall then establish a valid certificate path consisting of valid CA certificates, and demonstrate that the function succeeds. The evaluator shall then remove trust in one of the CA certificates, and show that the function fails.*
- ***Test 2:*** *The evaluator shall demonstrate that validating an expired certificate results in the function failing.*
- ***Test 3:*** *The evaluator shall test that the TOE can properly handle revoked certificates-"conditional on whether CRL, OCSP, OCSP Stapling or OCSP Multi-stapling is selected; if multiple methods are selected, then the following tests shall be performed for each method:*
  - *The evaluator shall test revocation of the node certificate.*
  - *The evaluator shall also test revocation of an intermediate CA certificate (i.e. the intermediate CA certificate should be revoked by the root CA), if intermediate CA certificates are supported. If OCSP stapling per RFC 6066 is the only supported revocation method, this test is omitted.*
  - *The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails.*
- ***Test 4:*** *If any OCSP option is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set, and verify that validation of the CRL fails.*

- ***Test 5:*** *The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)*
- ***Test 6:*** *The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)*
- ***Test 7:*** *The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)*
- ***Test 8:*** *(Conditional on support for EC certificates as indicated in FCS_COP.1/Sig). The evaluator shall establish a valid, trusted certificate chain consisting of an EC leaf certificate, an EC Intermediate CA certificate not designated as a trust anchor, and an EC certificate designated as a trusted anchor, where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.*
- ***Test 9:*** *(Conditional on support for EC certificates as indicated in FCS_COP.1/Sig). The evaluator shall replace the intermediate certificate in the certificate chain for Test 9 with a modified certificate, where the modified intermediate CA has a public key information field where the EC parameters uses an explicit format version of the Elliptic Curve parameters in the public key information field of the intermediate CA certificate from Test 9, and the modified Intermediate CA certificate is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.*

*FIA_X509_EXT.1.2*
***TSS***
*None.*

***Guidance***
*None.*

***Tests***
*The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created.*

- ***Test 1:*** *The evaluator shall ensure that the certificate of at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator shall confirm that validation of the certificate path fails (i) as part of the validation of the peer certificate belonging to this chain; and/or (ii) when attempting to add the CA certificate without the basicConstraints extension to the TOE's trust store.*
- ***Test 2:*** *The evaluator shall ensure that the certificate of at least one of the CAs in the chain has the CA flag in the basicConstraints extension not set (or set to FALSE). The evaluator shall confirm that validation of the certificate path fails (i) as part of the validation of the peer certificate belonging to this chain; and/or (ii) when attempting to add the CA certificate with the CA flag not set (or set to FALSE) in the basicConstraints extension to the TOE's trust store.*

# Appendix A - Implementation-Dependent Requirements

Implementation-Dependent Requirements are dependent on the TOE implementing a particular function. If the TOE fulfills any of these requirements, the vendor must either add the related SFR or disable the functionality for the evaluated configuration.

# Appendix B - Use Case Templates

## B.1 EAP

The configuration for *[USE CASE 1] EAP* modifies the base requirements as follows:

## B.2 Pre-Shared Keys

The configuration for *[USE CASE 2] Pre-Shared Keys* modifies the base requirements as follows:

- Include FIA_PSK_EXT.1 in the ST

## B.3 X.509 Certificates

The configuration for *[USE CASE 3] X.509 Certificates* modifies the base requirements as follows:

- Include FIA_PSK_EXT.1 in the ST

# Appendix C - Acronyms

| Acronym | Meaning |
|---|---|
| Base-PP | Base Protection Profile |
| CA | Certificate Authority |
| CC | Common Criteria |
| CEM | Common Evaluation Methodology |
| CSR | Certificate Signing Request |
| DN | Distinguished Name |
| EAP | Extensible Authentication Protocol |
| ECP | Eliptic Curve group modulo a Prime |
| EP | Extended Package |
| ESN | Extended Sequence Number |
| ESP | Encapsulating Security Payload |
| FP | Functional Package |
| FQDN | Fully Qualified Domain Name |
| ICMP | Internet Control Message Protocol |
| IKE | Internet Key Exchange |
| IPsec | Internet Protocol Security |
| ISAKMP | Internet Security Association and Key Management Protocol |
| OCSP | Online Certificate Status Protocol |
| OE | Operational Environment |
| PP | Protection Profile |
| PP-Configuration | Protection Profile Configuration |
| PP-Module | Protection Profile Module |
| PSK | Pre-Shared Key |
| SA | Security Association |
| SAR | Security Assurance Requirement |
| SFR | Security Functional Requirement |
| SPD | Security Policy Database |
| ST | Security Target |
| TOE | Target of Evaluation |
| TSF | TOE Security Functionality |
| TSFI | TSF Interface |
| TSS | TOE Summary Specification |
| UDP | User Datagram Protocol |
| VPN | Virtual Private Network |
| XAUTH | Extended Authentication |
| cPP | Collaborative Protection Profile |

# Appendix D - Bibliography


| Identifier | Title |
| --- | --- |
| [Functional Package for TLS] | Functional Package for Transport Layer Security (TLS) |
| [NIST SP 800-132 Part 1] | Recommendation for Password-Based Key Derivation Part 1: Storage Applications |
| [NIST SP 800-63B] | Digital Identity Guidelines: Authentication and Lifecycle Management |
| [RFC 2407] | The Internet IP Security Domain of Interpretation for ISAKMP |
| [RFC 4226] | HOTP: An HMAC-Based One-Time Password Algorithm |
| [RFC 5216] | The EAP-TLS Authentication Protocol |
| [RFC 5281] | Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0) |
| [RFC 6238] | TOTP: Time-Based One-Time Password Algorithm |
| [RFC 8996] | Deprecating TLS 1.0 and TLS 1.1 |
| [CC] | Common Criteria for Information Technology Security Evaluation -<br>• Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1 Revision 5, April 2017.<br>• Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1 Revision 5, April 2017.<br>• Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1 Revision 5, April 2017. |