n-doc
An Open Source Platform for CC-Documentation

Alexander Krumeich
alexander.krumeich@n-design.de

Cologne, Germany – https://n-design.de – Website in German only

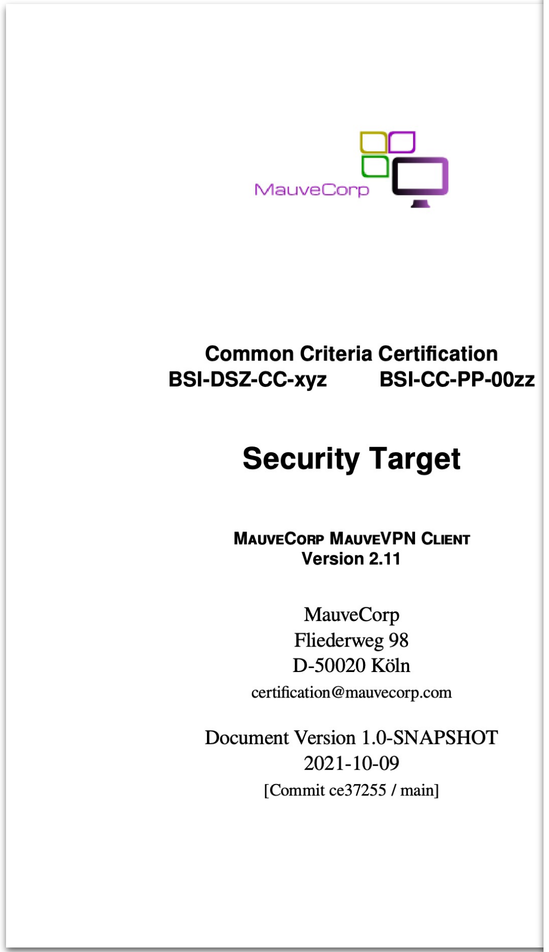
High-Quality, hyperlinked PDF Documents generated with LaTeX.

In development since 2017

Successful adaptation for several in-house projects

Applicable to different certification schemes

Published in 2020 as Open Source under MIT license.



MauveCorp

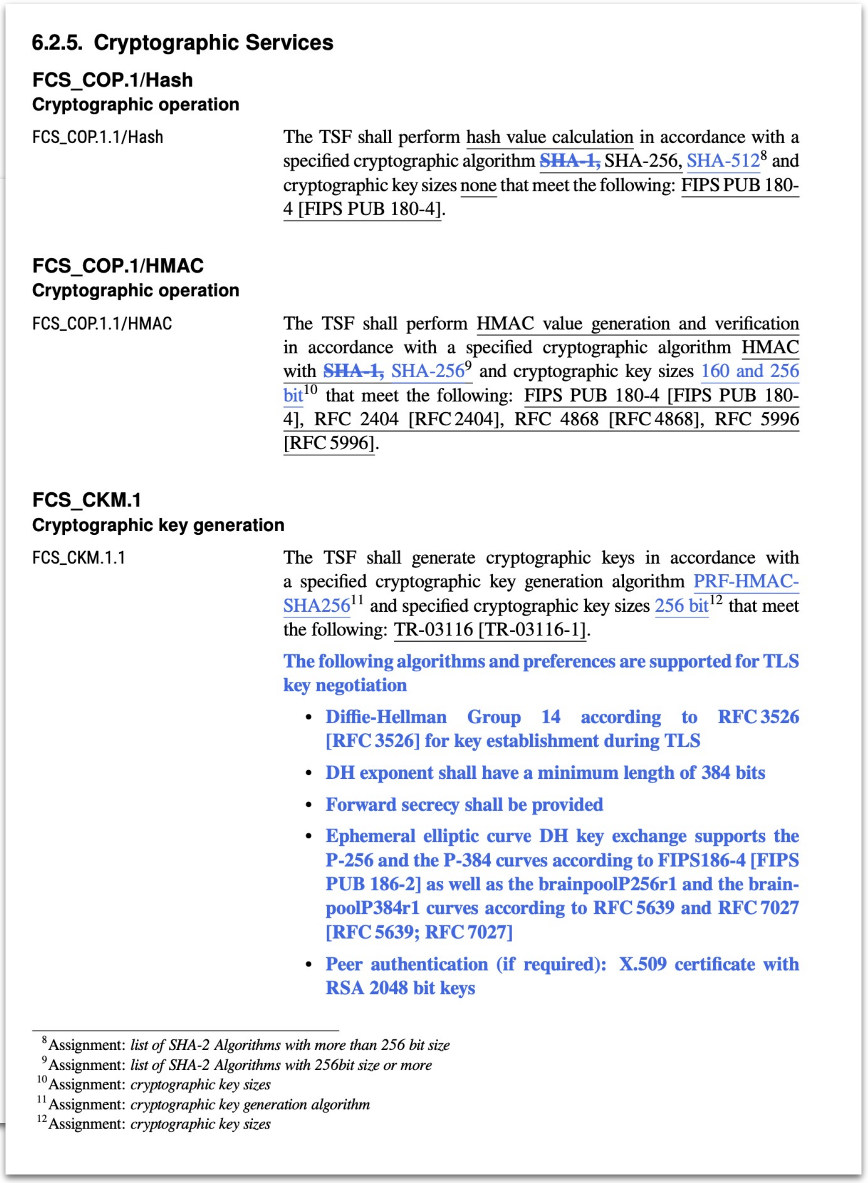
Common Criteria Certification
BSI-DSZ-CC-xyz BSI-CC-PP-00zz

Security Target

MauveCorp MauveVPN Client
Version 2.11

MauveCorp
Fliederweg 98
D-50020 Köln
certification@mauvecorp.com

Document Version 1.0-SNAPSHOT
2021-10-09
[Commit ce37255 / main]



6.2.5. Cryptographic Services

FCS_COP.1/Hash
Cryptographic operation

FCS_COP.1.1/Hash	The TSF shall perform hash value calculation in accordance with a specified cryptographic algorithm ~~SHA-1,~~ SHA-256, SHA-512[8] and cryptographic key sizes none that meet the following: FIPS PUB 180-4 [FIPS PUB 180-4].

FCS_COP.1/HMAC
Cryptographic operation

FCS_COP.1.1/HMAC	The TSF shall perform HMAC value generation and verification in accordance with a specified cryptographic algorithm HMAC with ~~SHA-1,~~ SHA-256[9] and cryptographic key sizes 160 and 256 bit[10] that meet the following: FIPS PUB 180-4 [FIPS PUB 180-4], RFC 2404 [RFC 2404], RFC 4868 [RFC 4868], RFC 5996 [RFC 5996].

FCS_CKM.1
Cryptographic key generation

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm PRF-HMAC-SHA256[11] and specified cryptographic key sizes 256 bit[12] that meet the following: TR-03116 [TR-03116-1].

The following algorithms and preferences are supported for TLS key negotiation

- **Diffie-Hellman Group 14 according to RFC 3526 [RFC 3526] for key establishment during TLS**
- **DH exponent shall have a minimum length of 384 bits**
- **Forward secrecy shall be provided**
- **Ephemeral elliptic curve DH key exchange supports the P-256 and the P-384 curves according to FIPS186-4 [FIPS PUB 186-2] as well as the brainpoolP256r1 and the brainpoolP384r1 curves according to RFC 5639 and RFC 7027 [RFC 5639; RFC 7027]**
- **Peer authentication (if required): X.509 certificate with RSA 2048 bit keys**

[8] Assignment: *list of SHA-2 Algorithms with more than 256 bit size*
[9] Assignment: *list of SHA-2 Algorithms with 256bit size or more*
[10] Assignment: *cryptographic key sizes*
[11] Assignment: *cryptographic key generation algorithm*
[12] Assignment: *cryptographic key sizes*

Challenges of Evaluating our Product

Demanding Protection Profile >130 SFR

Complex TOE 160 modules, 23 subsystems, >60 TSFI

Documentation 15 documents, approx. 4,500 pages

Technical / Organizational Requirements

Collaboration of >6 team members, 3 orgs.

Versioning to track changes in documents

Consistency of content and appearance

Navigation with generated hyperlinks

Acceptance by editors

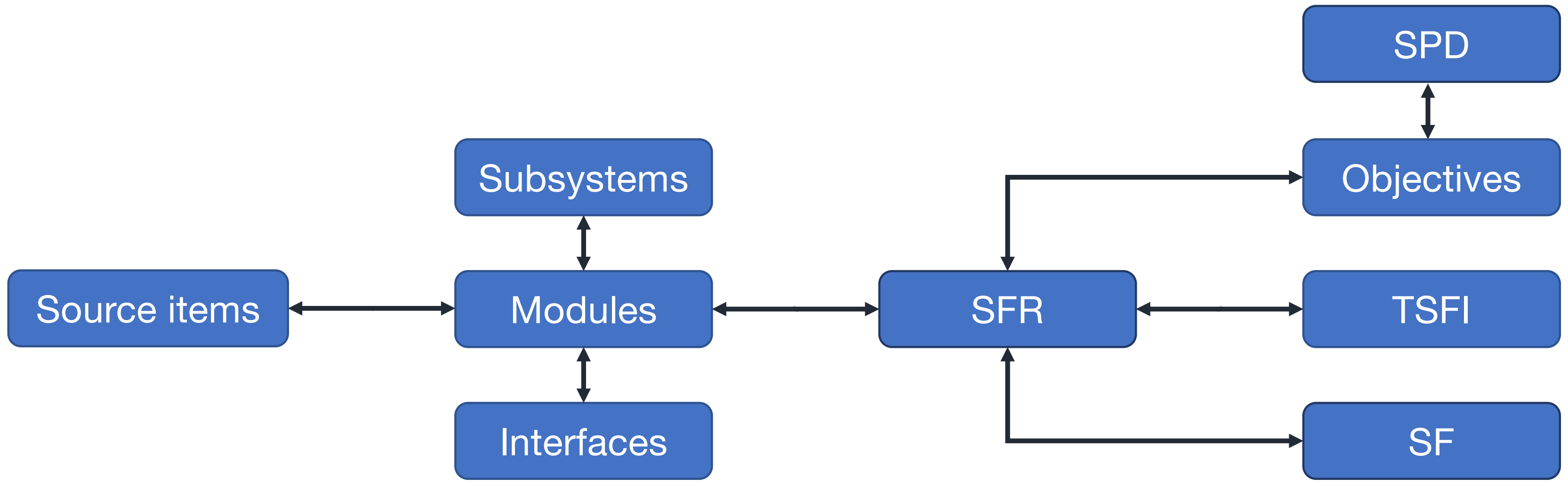
n-doc Key features

TOE model in a relational database

$\LaTeX$ as typesetting tool

Best practices of software engineering

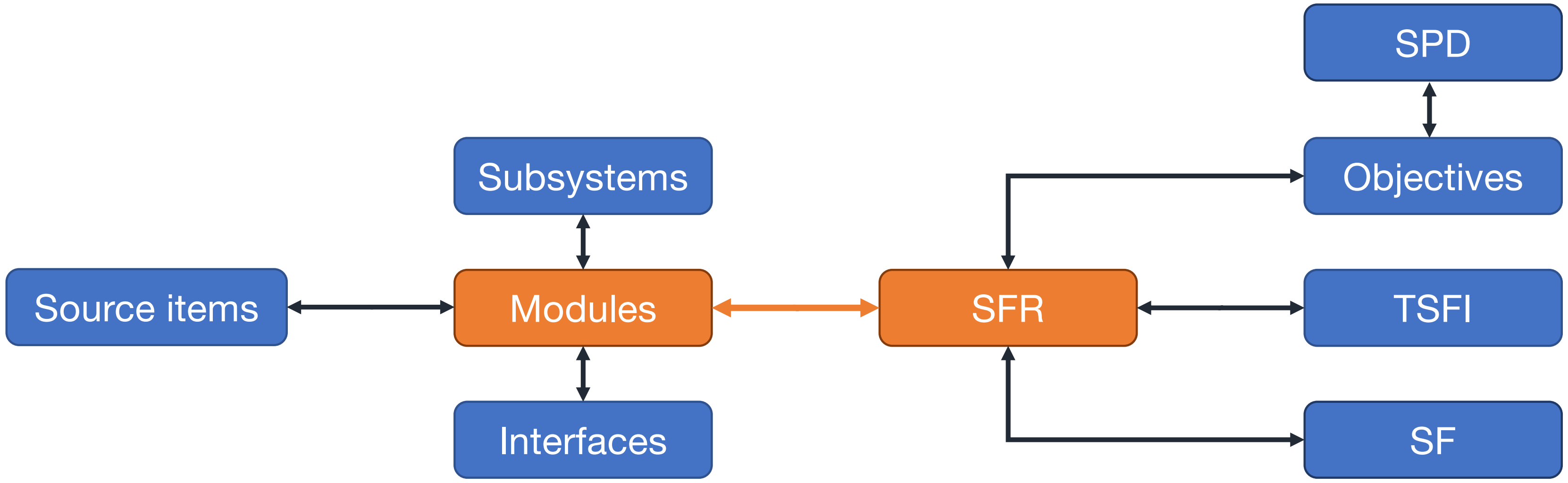
TOE model in a relational database



Gaining assurance about relations in the TOE.

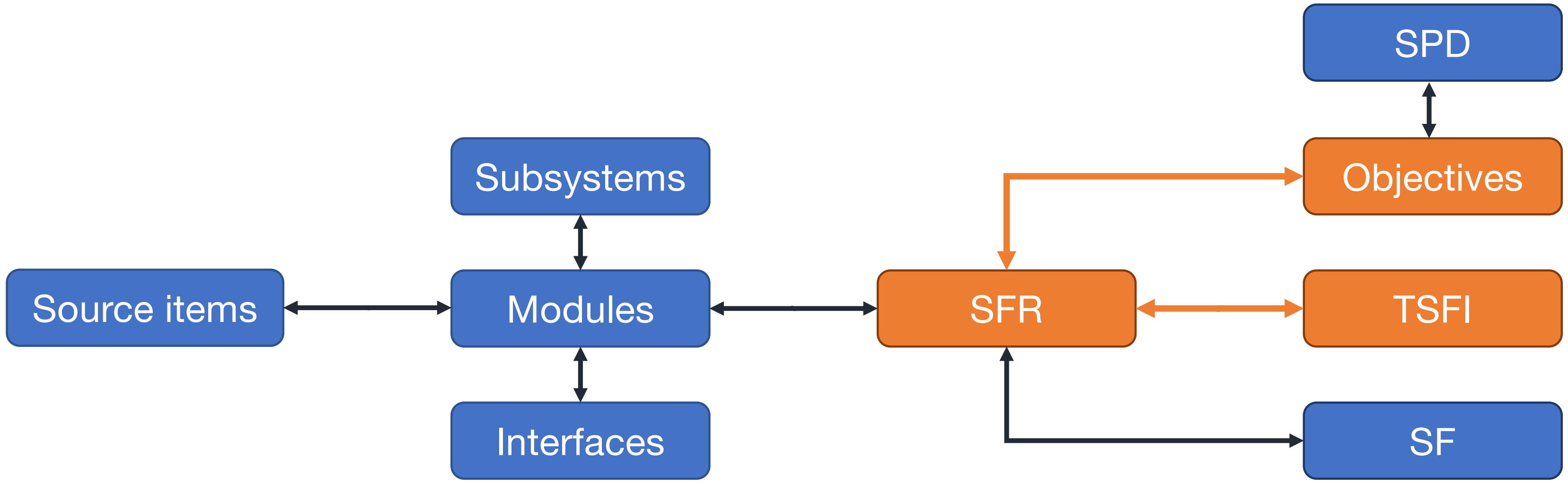
Recognizing and using *undiscovered* relations

TOE model in a relational database



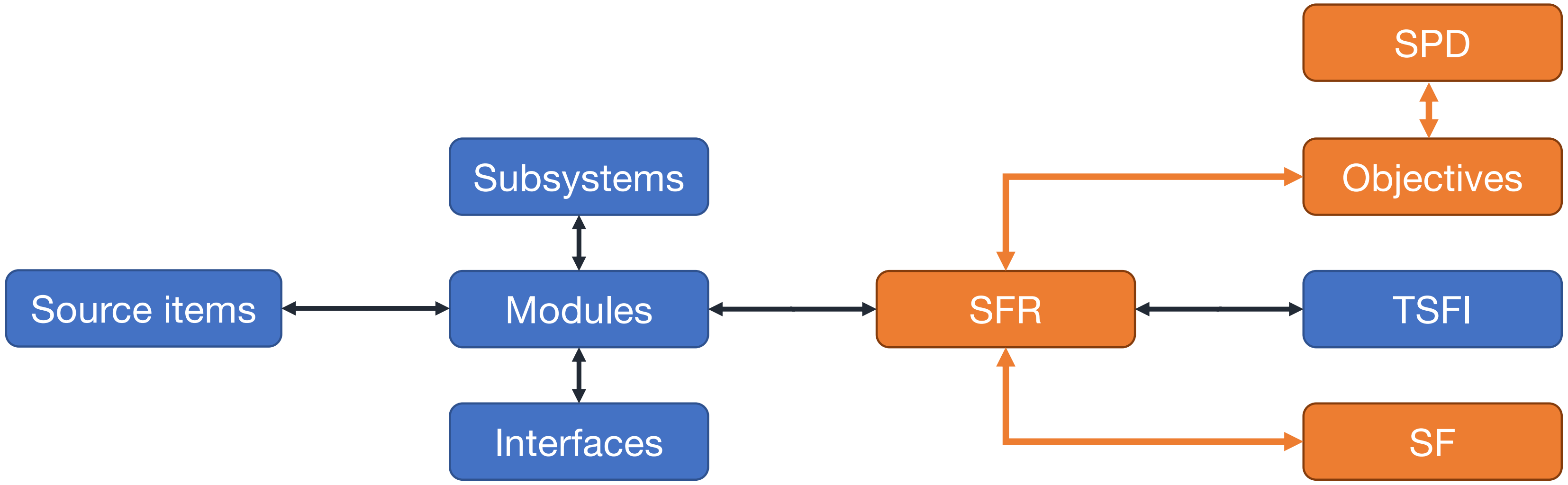
Which SFR is enforced by which module?

TOE model in a relational database



Which objective is fulfilled by which TSFI?

TOE model in a relational database



What is the relation between the Security Problem Definition and the Security Functionalities?

Using the Database

Within the documents

Enforcing consistent terminology

Generating tables, text and references

As an additional deliverable

Evaluator receives the DB file
to formulate their own queries

LaTeX

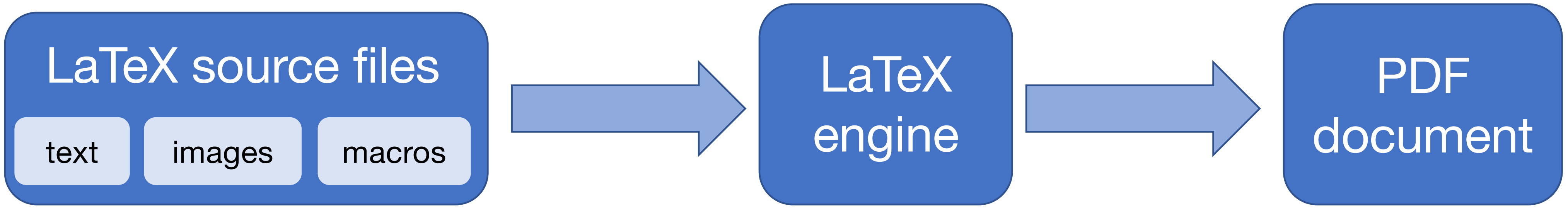
Typesetting system mostly used in academia

40 years old – and still going strong!

Workflow similar to software development

General LaTeX Workflow

LaTeX reads source files and creates PDF



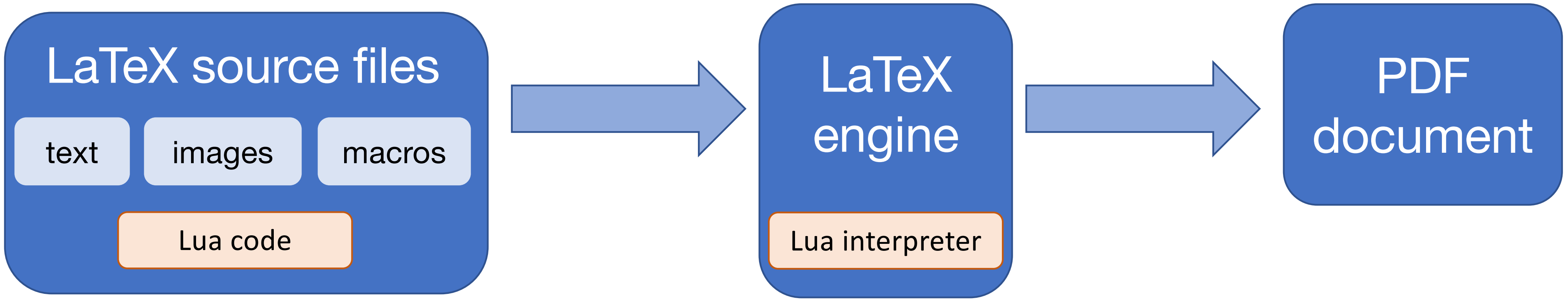
Formatting and structure by using macros

This text is in \textit{italics}.

\section{Headline on Level 1}

General LaTeX Workflow

LaTeX reads source files and creates PDF



Formatting and structure by using macros

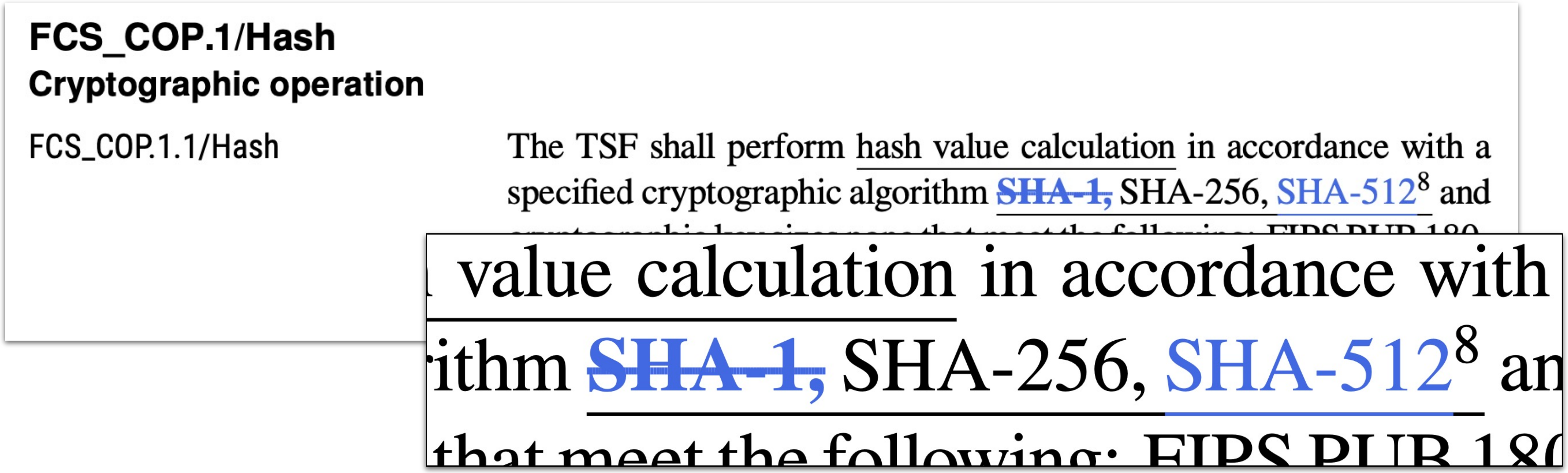
```
This text is in \textit{italics}.

\section{Headline on Level 1}
```

Domain Specific Macros separating content from layout

\keyword{CACHED}	for printing keywords
\kocobox{}	name of the TOE “Kocobox MED+”
\tds{mod.aas.core}	Resolve the name of a subsystem, module or interface

Semantic Markup



FCS_COP.1/Hash
Cryptographic operation

FCS_COP.1.1/Hash The TSF shall perform hash value calculation in accordance with a specified cryptographic algorithm ~~SHA-1,~~ SHA-256, SHA-512[8] and

value calculation in accordance with

ithm ~~SHA-1,~~ SHA-256, SHA-512[8] an

that meet the following: FIPS PUB 180

…in accordance with a specified cryptographic algorithm \ppassigned{\stdeleted{SHA-1,} SHA-256, \stassigned{SHA-512}} and key sizes \ppassigned{none} that meet…

Enforcing Consistent Terminology

"The module \tds{mod.aas.core} enforces access control of the TOE."

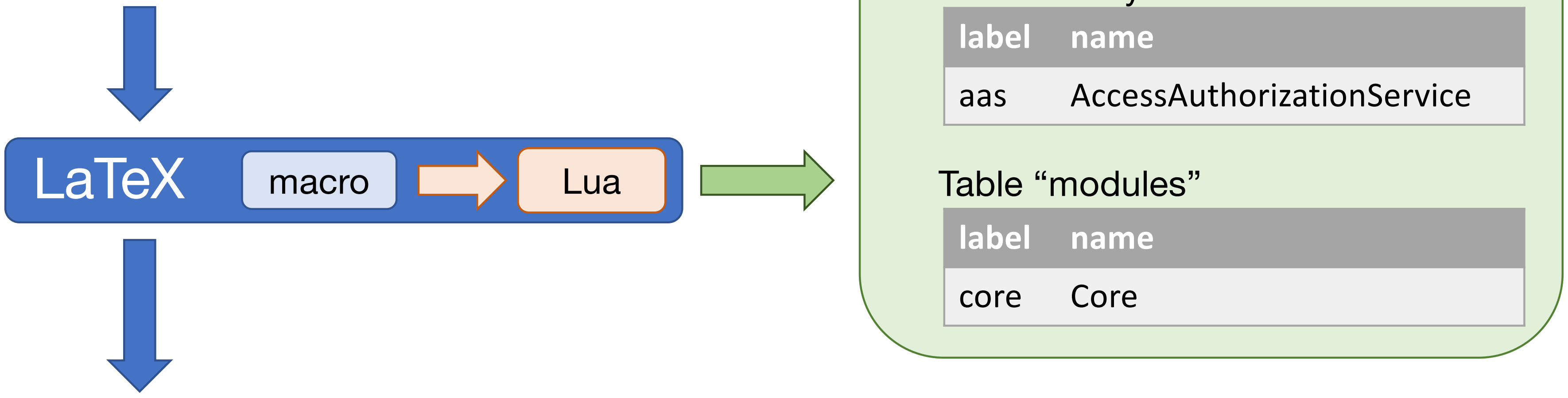


Table "subsystems"

label	name
aas	AccessAuthorizationService

Table "modules"

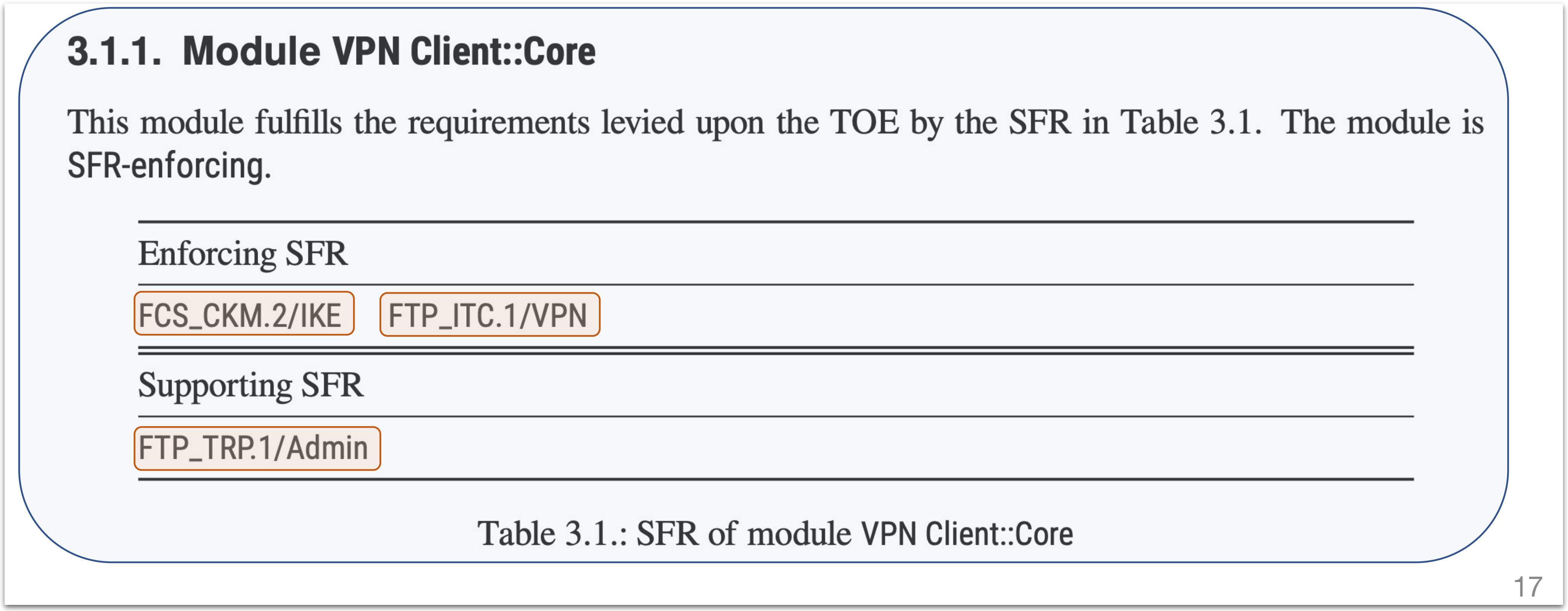
label	name
core	Core

"The module AccessAuthorizationService::Core enforces access control of the TOE."

Generating Text and Tables

Programatically generated text

Automatically generated hyperlinks



3.1.1. Module VPN Client::Core

This module fulfills the requirements levied upon the TOE by the SFR in Table 3.1. The module is SFR-enforcing.

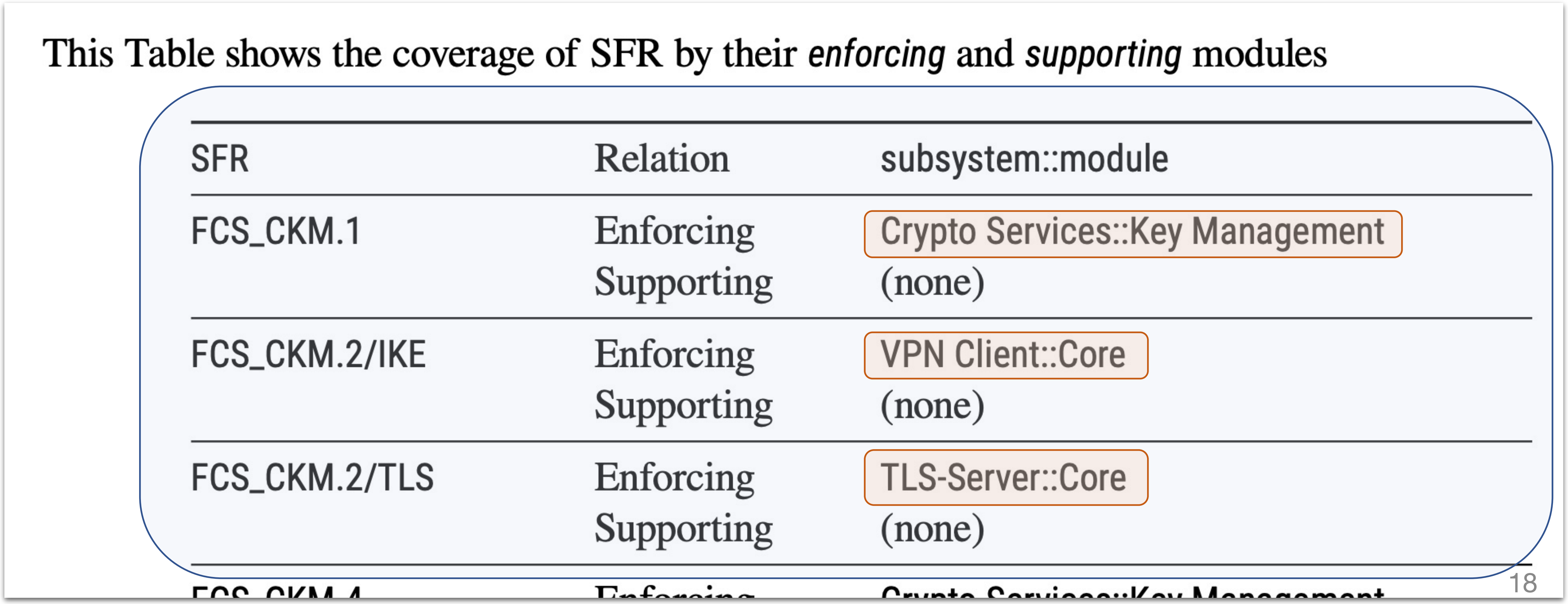
Enforcing SFR	
FCS_CKM.2/IKE	FTP_ITC.1/VPN
Supporting SFR	
FTP_TRP.1/Admin	

Table 3.1.: SFR of module VPN Client::Core

Generating Text and Tables

Programatically generated text

Automatically generated hyperlinks

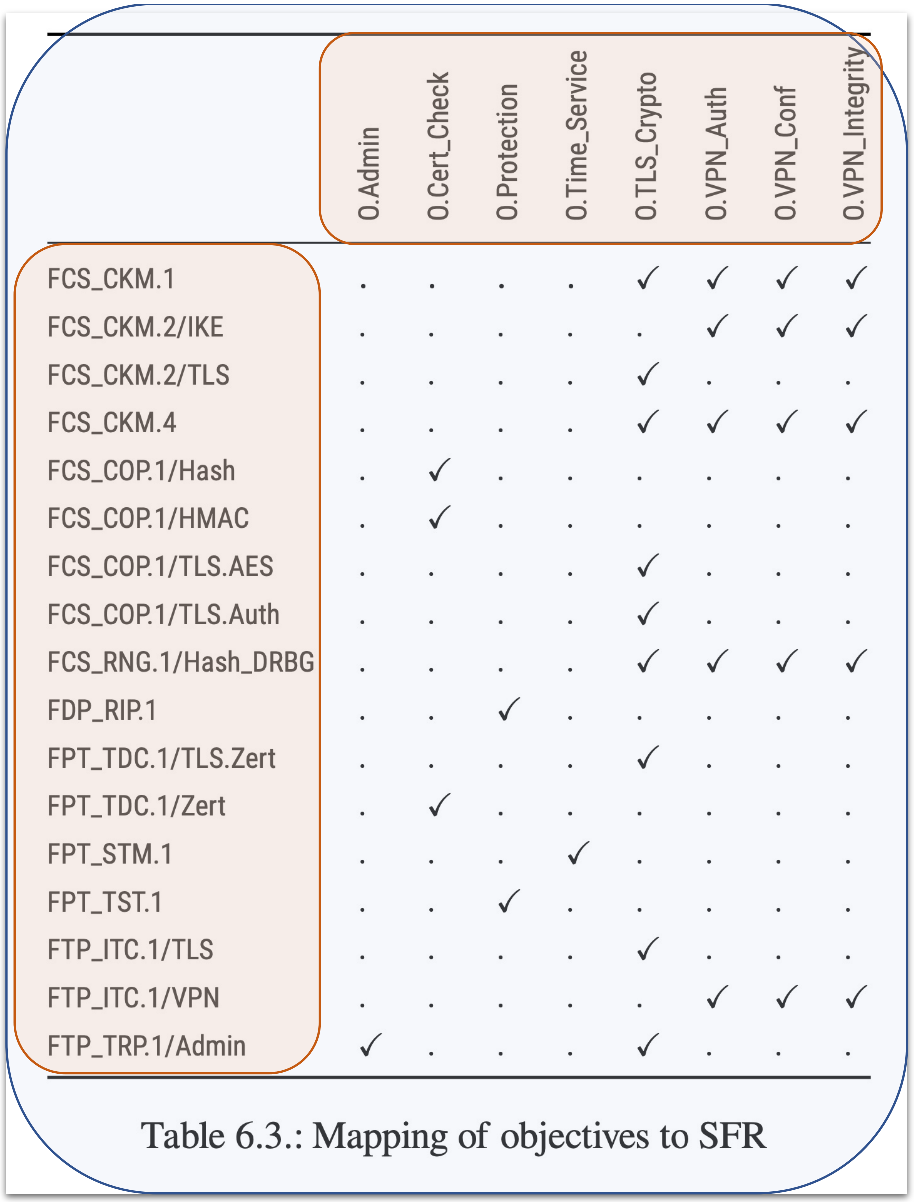


This Table shows the coverage of SFR by their *enforcing* and *supporting* modules

SFR	Relation	subsystem::module
FCS_CKM.1	Enforcing Supporting	Crypto Services::Key Management (none)
FCS_CKM.2/IKE	Enforcing Supporting	VPN Client::Core (none)
FCS_CKM.2/TLS	Enforcing Supporting	TLS-Server::Core (none)
FCS_CKM.4	Enforcing	Crypto Services::Key Management

Programatically generated text

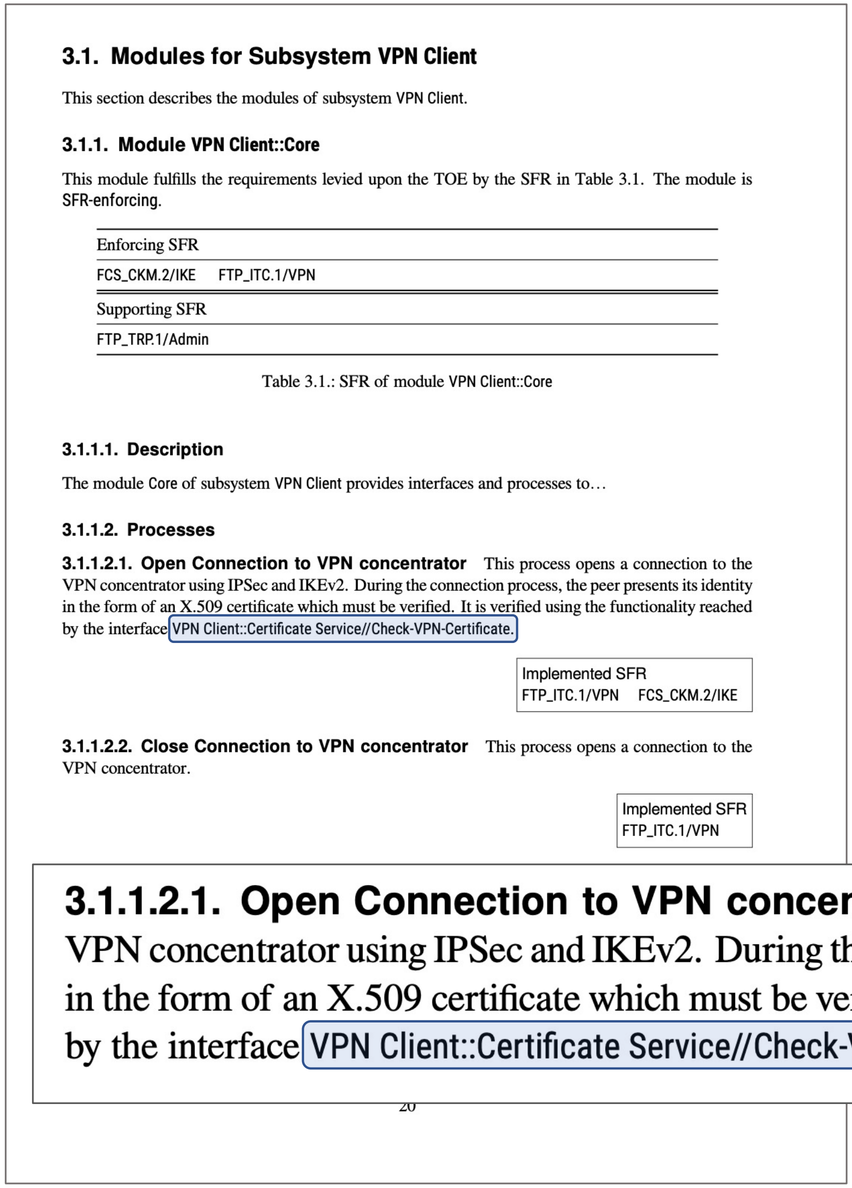
Automatically generated hyperlinks



	O.Admin	O.Cert_Check	O.Protection	O.Time_Service	O.TLS_Crypto	O.VPN_Auth	O.VPN_Conf	O.VPN_Integrity
FCS_CKM.1	.	.	.	.	✓	✓	✓	✓
FCS_CKM.2/IKE	.	.	.	.	.	✓	✓	✓
FCS_CKM.2/TLS	.	.	.	.	✓	.	.	.
FCS_CKM.4	.	.	.	.	✓	✓	✓	✓
FCS_COP.1/Hash	.	✓	.	.	.	.	.	.
FCS_COP.1/HMAC	.	✓	.	.	.	.	.	.
FCS_COP.1/TLS.AES	.	.	.	.	✓	.	.	.
FCS_COP.1/TLS.Auth	.	.	.	.	✓	.	.	.
FCS_RNG.1/Hash_DRBG	.	.	.	.	✓	✓	✓	✓
FDP_RIP.1	.	.	✓	.	.	.	.	.
FPT_TDC.1/TLS.Zert	.	.	.	.	✓	.	.	.
FPT_TDC.1/Zert	.	✓	.	.	.	.	.	.
FPT_STM.1	.	.	.	✓	.	.	.	.
FPT_TST.1	.	.	✓	.	.	.	.	.
FTP_ITC.1/TLS	.	.	.	.	✓	.	.	.
FTP_ITC.1/VPN	.	.	.	.	.	✓	✓	✓
FTP_TRP.1/Admin	✓	.	.	.	✓	.	.	.

Table 6.3.: Mapping of objectives to SFR

Using non-linear Navigation



3.1. Modules for Subsystem VPN Client

This section describes the modules of subsystem VPN Client.

3.1.1. Module VPN Client::Core

This module fulfills the requirements levied upon the TOE by the SFR in Table 3.1. The module is SFR-enforcing.

Enforcing SFR	
FCS_CKM.2/IKE	FTP_ITC.1/VPN
Supporting SFR	
FTP_TRP.1/Admin	

Table 3.1.: SFR of module VPN Client::Core

3.1.1.1. Description

The module Core of subsystem VPN Client provides interfaces and processes to…

3.1.1.2. Processes

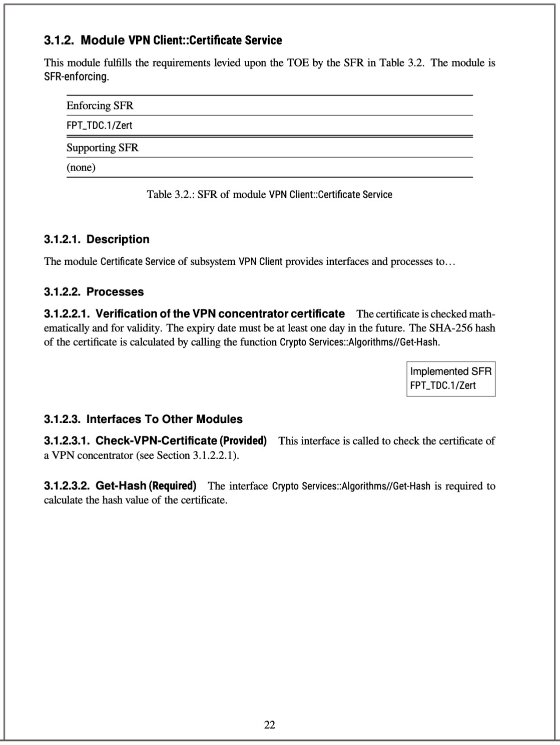
3.1.1.2.1. Open Connection to VPN concentrator This process opens a connection to the VPN concentrator using IPSec and IKEv2. During the connection process, the peer presents its identity in the form of an X.509 certificate which must be verified. It is verified using the functionality reached by the interface VPN Client::Certificate Service//Check-VPN-Certificate.

Implemented SFR	
FTP_ITC.1/VPN	FCS_CKM.2/IKE

3.1.1.2.2. Close Connection to VPN concentrator This process opens a connection to the VPN concentrator.

Implemented SFR
FTP_ITC.1/VPN

20



3.1.2. Module VPN Client::Certificate Service

This module fulfills the requirements levied upon the TOE by the SFR in Table 3.2. The module is SFR-enforcing.

Enforcing SFR
FPT_TDC.1/Zert
Supporting SFR
(none)

Table 3.2.: SFR of module VPN Client::Certificate Service

3.1.2.1. Description

The module Certificate Service of subsystem VPN Client provides interfaces and processes to…

3.1.2.2. Processes

3.1.2.2.1. Verification of the VPN concentrator certificate The certificate is checked mathematically and for validity. The expiry date must be at least one day in the future. The SHA-256 hash of the certificate is calculated by calling the function Crypto Services::Algorithms//Get-Hash.

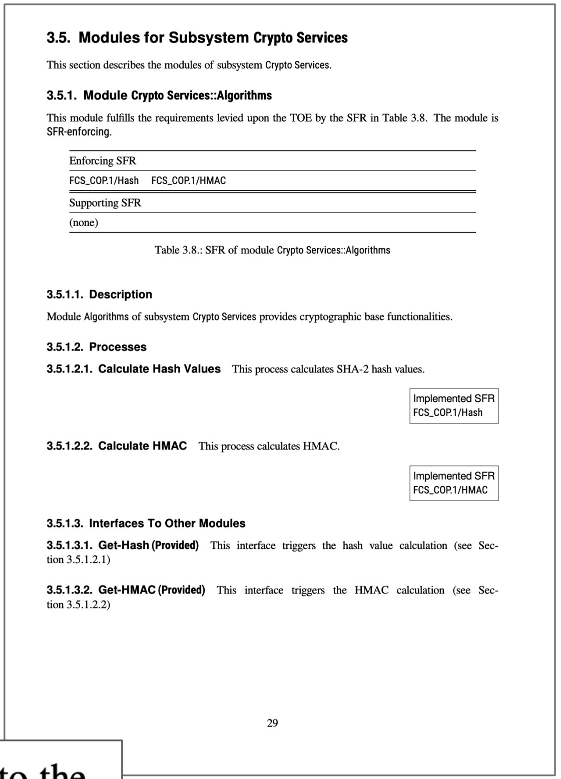
Implemented SFR
FPT_TDC.1/Zert

3.1.2.3. Interfaces To Other Modules

3.1.2.3.1. Check-VPN-Certificate (Provided) This interface is called to check the certificate of a VPN concentrator (see Section 3.1.2.2.1).

3.1.2.3.2. Get-Hash (Required) The interface Crypto Services::Algorithms//Get-Hash is required to calculate the hash value of the certificate.

22



3.5. Modules for Subsystem Crypto Services

This section describes the modules of subsystem Crypto Services.

3.5.1. Module Crypto Services::Algorithms

This module fulfills the requirements levied upon the TOE by the SFR in Table 3.8. The module is SFR-enforcing.

Enforcing SFR	
FCS_COP.1/Hash	FCS_COP.1/HMAC
Supporting SFR	
(none)	

Table 3.8.: SFR of module Crypto Services::Algorithms

3.5.1.1. Description

Module Algorithms of subsystem Crypto Services provides cryptographic base functionalities.

3.5.1.2. Processes

3.5.1.2.1. Calculate Hash Values This process calculates SHA-2 hash values.

Implemented SFR
FCS_COP.1/Hash

3.5.1.2.2. Calculate HMAC This process calculates HMAC.

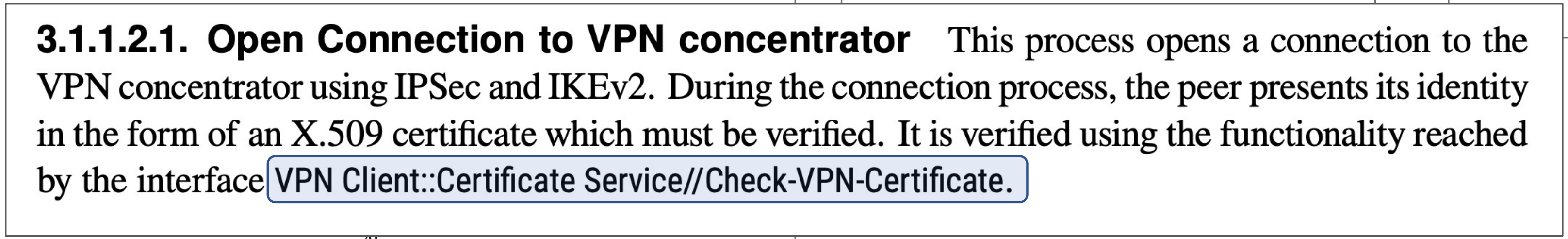
Implemented SFR
FCS_COP.1/HMAC

3.5.1.3. Interfaces To Other Modules

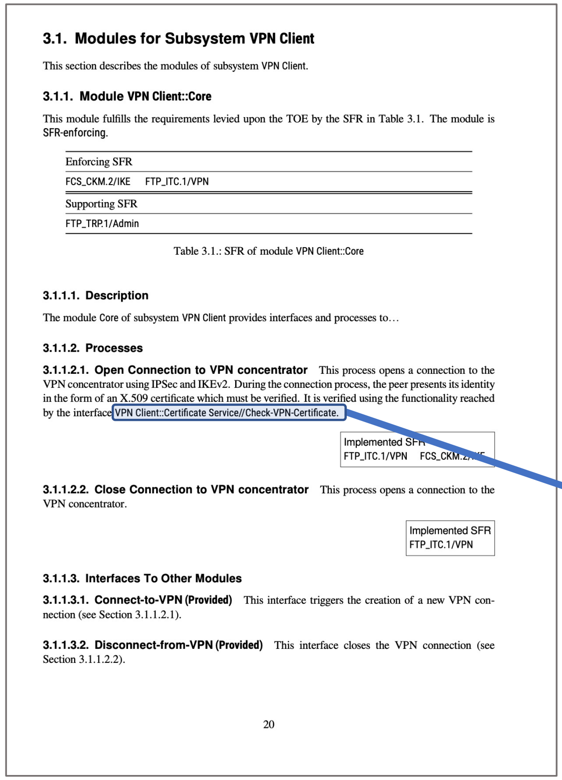
3.5.1.3.1. Get-Hash (Provided) This interface triggers the hash value calculation (see Section 3.5.1.2.1)

3.5.1.3.2. Get-HMAC (Provided) This interface triggers the HMAC calculation (see Section 3.5.1.2.2)

29



3.1.1.2.1. Open Connection to VPN concentrator This process opens a connection to the VPN concentrator using IPSec and IKEv2. During the connection process, the peer presents its identity in the form of an X.509 certificate which must be verified. It is verified using the functionality reached by the interface VPN Client::Certificate Service//Check-VPN-Certificate.



3.1. Modules for Subsystem VPN Client

This section describes the modules of subsystem VPN Client.

3.1.1. Module VPN Client::Core

This module fulfills the requirements levied upon the TOE by the SFR in Table 3.1. The module is SFR-enforcing.

Enforcing SFR	
FCS_CKM.2/IKE	FTP_ITC.1/VPN
Supporting SFR	
FTP_TRP.1/Admin	

Table 3.1.: SFR of module VPN Client::Core

3.1.1.1. Description

The module Core of subsystem VPN Client provides interfaces and processes to…

3.1.1.2. Processes

3.1.1.2.1. Open Connection to VPN concentrator This process opens a connection to the VPN concentrator using IPSec and IKEv2. During the connection process, the peer presents its identity in the form of an X.509 certificate which must be verified. It is verified using the functionality reached by the interface VPN Client::Certificate Service//Check-VPN-Certificate.

Implemented SFR
FTP_ITC.1/VPN FCS_CKM.2/IKE

3.1.1.2.2. Close Connection to VPN concentrator This process opens a connection to the VPN concentrator.

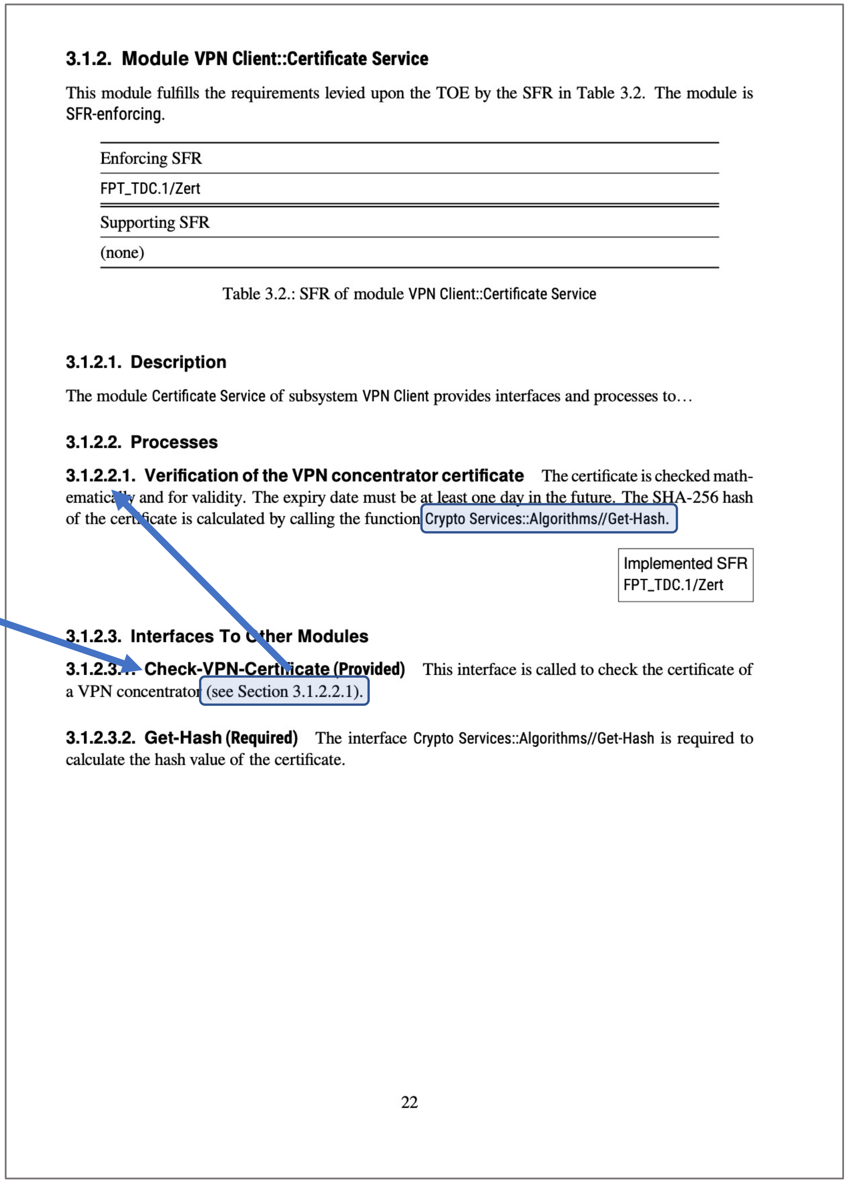
Implemented SFR
FTP_ITC.1/VPN

3.1.1.3. Interfaces To Other Modules

3.1.1.3.1. Connect-to-VPN (Provided) This interface triggers the creation of a new VPN connection (see Section 3.1.1.2.1).

3.1.1.3.2. Disconnect-from-VPN (Provided) This interface closes the VPN connection (see Section 3.1.1.2.2).

20



3.1.2. Module VPN Client::Certificate Service

This module fulfills the requirements levied upon the TOE by the SFR in Table 3.2. The module is SFR-enforcing.

Enforcing SFR
FPT_TDC.1/Zert
Supporting SFR
(none)

Table 3.2.: SFR of module VPN Client::Certificate Service

3.1.2.1. Description

The module Certificate Service of subsystem VPN Client provides interfaces and processes to…

3.1.2.2. Processes

3.1.2.2.1. Verification of the VPN concentrator certificate The certificate is checked mathematically and for validity. The expiry date must be at least one day in the future. The SHA-256 hash of the certificate is calculated by calling the function Crypto Services::Algorithms//Get-Hash.

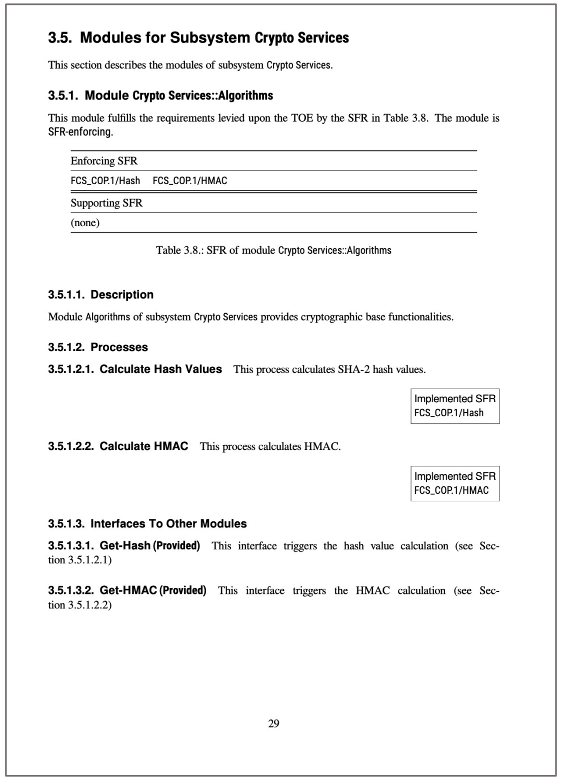
Implemented SFR
FPT_TDC.1/Zert

3.1.2.3. Interfaces To Other Modules

3.1.2.3.1. Check-VPN-Certificate (Provided) This interface is called to check the certificate of a VPN concentrator (see Section 3.1.2.2.1).

3.1.2.3.2. Get-Hash (Required) The interface Crypto Services::Algorithms//Get-Hash is required to calculate the hash value of the certificate.

22



3.5. Modules for Subsystem Crypto Services

This section describes the modules of subsystem Crypto Services.

3.5.1. Module Crypto Services::Algorithms

This module fulfills the requirements levied upon the TOE by the SFR in Table 3.8. The module is SFR-enforcing.

Enforcing SFR	
FCS_COP.1/Hash	FCS_COP.1/HMAC
Supporting SFR	
(none)	

Table 3.8.: SFR of module Crypto Services::Algorithms

3.5.1.1. Description

Module Algorithms of subsystem Crypto Services provides cryptographic base functionalities.

3.5.1.2. Processes

3.5.1.2.1. Calculate Hash Values This process calculates SHA-2 hash values.

Implemented SFR
FCS_COP.1/Hash

3.5.1.2.2. Calculate HMAC This process calculates HMAC.

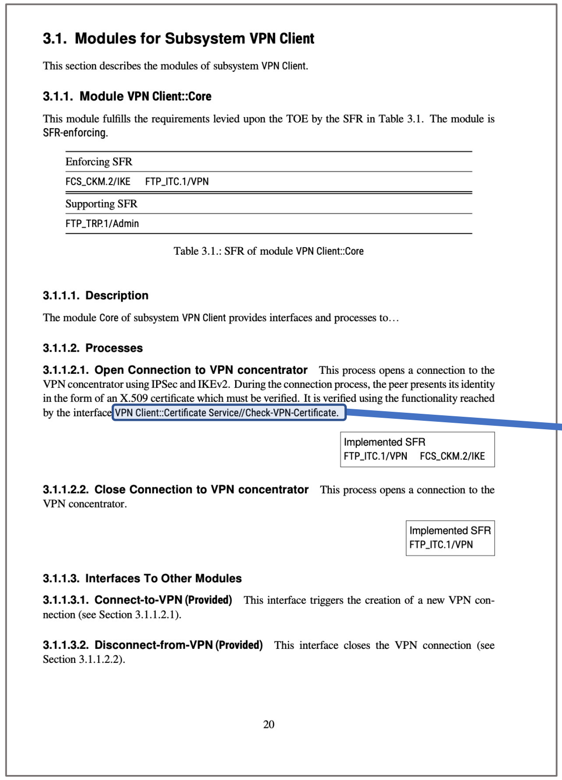
Implemented SFR
FCS_COP.1/HMAC

3.5.1.3. Interfaces To Other Modules

3.5.1.3.1. Get-Hash (Provided) This interface triggers the hash value calculation (see Section 3.5.1.2.1)

3.5.1.3.2. Get-HMAC (Provided) This interface triggers the HMAC calculation (see Section 3.5.1.2.2)

29



3.1. Modules for Subsystem VPN Client

This section describes the modules of subsystem VPN Client.

3.1.1. Module VPN Client::Core

This module fulfills the requirements levied upon the TOE by the SFR in Table 3.1. The module is SFR-enforcing.

Enforcing SFR	
FCS_CKM.2/IKE	FTP_ITC.1/VPN
Supporting SFR	
FTP_TRP.1/Admin	

Table 3.1.: SFR of module VPN Client::Core

3.1.1.1. Description

The module Core of subsystem VPN Client provides interfaces and processes to…

3.1.1.2. Processes

3.1.1.2.1. Open Connection to VPN concentrator This process opens a connection to the VPN concentrator using IPSec and IKEv2. During the connection process, the peer presents its identity in the form of an X.509 certificate which must be verified. It is verified using the functionality reached by the interface VPN Client::Certificate Service//Check-VPN-Certificate.

Implemented SFR	
FTP_ITC.1/VPN	FCS_CKM.2/IKE

3.1.1.2.2. Close Connection to VPN concentrator This process opens a connection to the VPN concentrator.

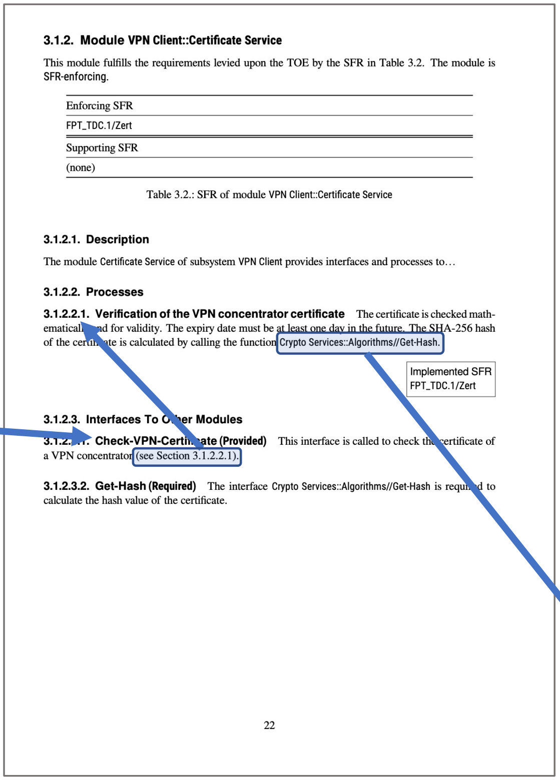
Implemented SFR
FTP_ITC.1/VPN

3.1.1.3. Interfaces To Other Modules

3.1.1.3.1. Connect-to-VPN (Provided) This interface triggers the creation of a new VPN connection (see Section 3.1.1.2.1).

3.1.1.3.2. Disconnect-from-VPN (Provided) This interface closes the VPN connection (see Section 3.1.1.2.2).

20



3.1.2. Module VPN Client::Certificate Service

This module fulfills the requirements levied upon the TOE by the SFR in Table 3.2. The module is SFR-enforcing.

Enforcing SFR
FPT_TDC.1/Zert
Supporting SFR
(none)

Table 3.2.: SFR of module VPN Client::Certificate Service

3.1.2.1. Description

The module Certificate Service of subsystem VPN Client provides interfaces and processes to…

3.1.2.2. Processes

3.1.2.2.1. Verification of the VPN concentrator certificate The certificate is checked mathematically and for validity. The expiry date must be at least one day in the future. The SHA-256 hash of the certificate is calculated by calling the function Crypto Services::Algorithms//Get-Hash.

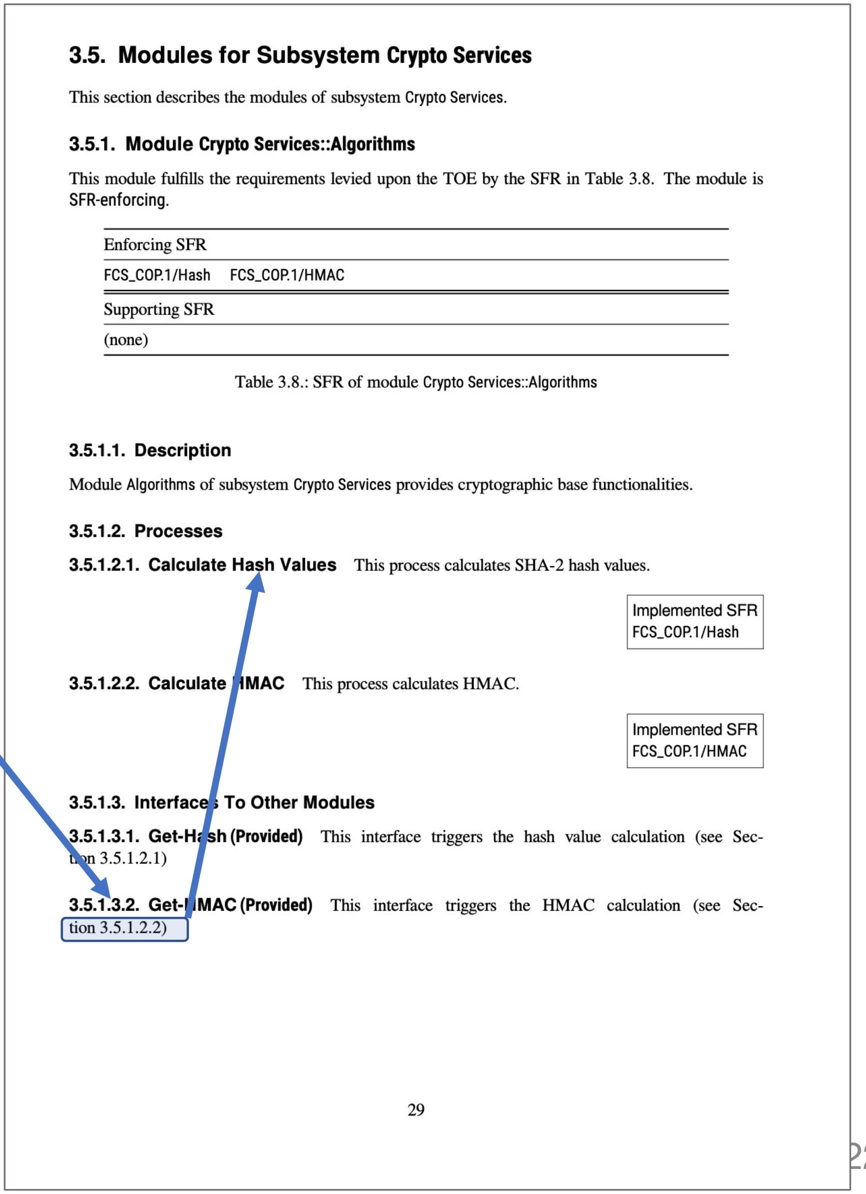
Implemented SFR
FPT_TDC.1/Zert

3.1.2.3. Interfaces To Other Modules

3.1.2.3.1. Check-VPN-Certificate (Provided) This interface is called to check the certificate of a VPN concentrator (see Section 3.1.2.2.1).

3.1.2.3.2. Get-Hash (Required) The interface Crypto Services::Algorithms//Get-Hash is required to calculate the hash value of the certificate.

22



3.5. Modules for Subsystem Crypto Services

This section describes the modules of subsystem Crypto Services.

3.5.1. Module Crypto Services::Algorithms

This module fulfills the requirements levied upon the TOE by the SFR in Table 3.8. The module is SFR-enforcing.

Enforcing SFR	
FCS_COP.1/Hash	FCS_COP.1/HMAC
Supporting SFR	
(none)	

Table 3.8.: SFR of module Crypto Services::Algorithms

3.5.1.1. Description

Module Algorithms of subsystem Crypto Services provides cryptographic base functionalities.

3.5.1.2. Processes

3.5.1.2.1. Calculate Hash Values This process calculates SHA-2 hash values.

Implemented SFR
FCS_COP.1/Hash

3.5.1.2.2. Calculate HMAC This process calculates HMAC.

Implemented SFR
FCS_COP.1/HMAC

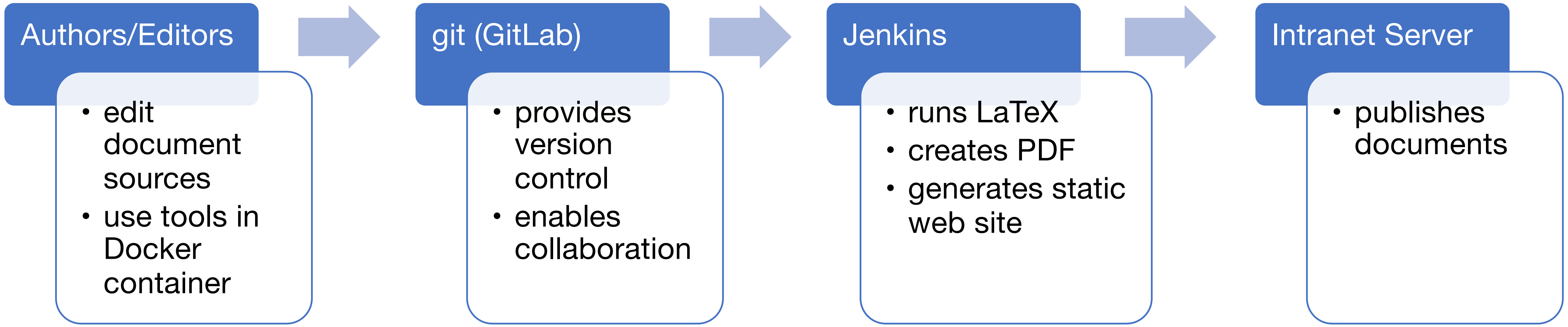
3.5.1.3. Interfaces To Other Modules

3.5.1.3.1. Get-Hash (Provided) This interface triggers the hash value calculation (see Section 3.5.1.2.1)

3.5.1.3.2. Get-HMAC (Provided) This interface triggers the HMAC calculation (see Section 3.5.1.2.2)

29

Continuous Delivery of Documents



"Where is the current version?"

"What did we ship three weeks ago?"

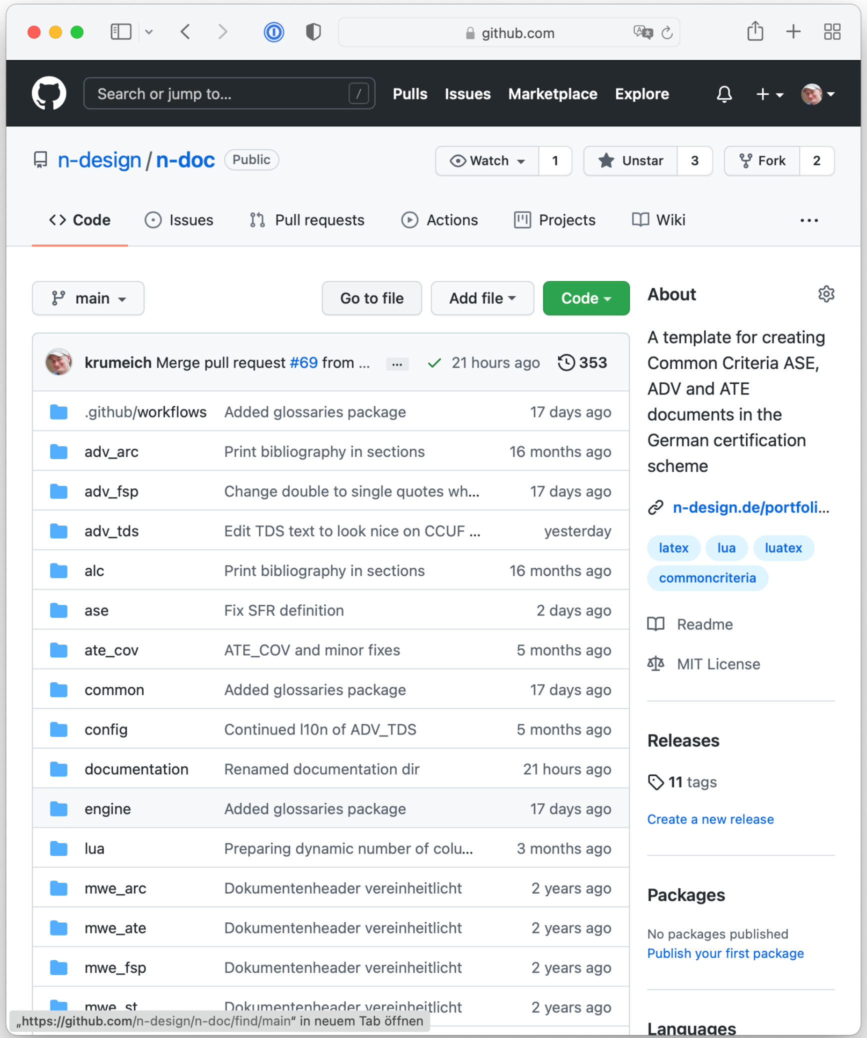
What’s in the Bag?

CC documents for a fictional TOE:
ASE, ADV_FSP, ADV_TDS, ATE_COV

Templates for ADV_ARC, ALC

Lua programs, sample DB, Makefiles, CI pipeline, Source files for Docker image, documentation

MIT License **Free software**



n-design / n-doc Public

krumeich Merge pull request #69 from ... ✓ 21 hours ago 353

.github/workflows	Added glossaries package	17 days ago
adv_arc	Print bibliography in sections	16 months ago
adv_fsp	Change double to single quotes wh...	17 days ago
adv_tds	Edit TDS text to look nice on CCUF ...	yesterday
alc	Print bibliography in sections	16 months ago
ase	Fix SFR definition	2 days ago
ate_cov	ATE_COV and minor fixes	5 months ago
common	Added glossaries package	17 days ago
config	Continued l10n of ADV_TDS	5 months ago
documentation	Renamed documentation dir	21 hours ago
engine	Added glossaries package	17 days ago
lua	Preparing dynamic number of colu...	3 months ago
mwe_arc	Dokumentenheader vereinheitlicht	2 years ago
mwe_ate	Dokumentenheader vereinheitlicht	2 years ago
mwe_fsp	Dokumentenheader vereinheitlicht	2 years ago
mwe_st	Dokumentenheader vereinheitlicht	2 years ago

„https://github.com/n-design/n-doc/find/main“ in neuem Tab öffnen

About

A template for creating Common Criteria ASE, ADV and ATE documents in the German certification scheme

n-design.de/portfoli...

latex lua luatex commoncriteria

Readme

MIT License

Releases

11 tags

Create a new release

Packages

No packages published
Publish your first package

Languages

https://github.com/n-design/n-doc

Sounds good! How can I use n-doc?

Step 1	Install git and Docker
Step 2	Clone https://github.com/n-design/n-doc.git
Step 3	Call ./runmake.sh
Step 4	Enjoy documents in ./deliverables
optional Step 5	**Reach out to us for support, customization and training.**

Do we have time for final thoughts?

Highly scalable solution

15 Documents, 4,500 pages, 10 minutes delivery time

Tremendous gains in efficiency

Document management takes 10% of a single person’s time.

Outstanding evaluator satisfaction

Fewer routine tasks because of TOE database model.
High degree of automation ensures reliability.

Thanks – Merci – Gracias – Danke

Questions?

https://github.com/n-design/n-doc

alexander.krumeich@n-design.de