

Australian Standard™

**Information technology—Security
techniques—Evaluation criteria for IT
security**

Part 2: Security functional requirements



This Australian Standard was prepared by Committee IT-012, Information systems—Security and identification technology. It was approved on behalf of the Council of Standards Australia on 29 January 2004 and published on 17 March 2004.

The following are represented on Committee IT-012:

Attorney General's Department
Australian Association of Permanent Building Societies
Australian Bankers Association
Australian Chamber of Commerce and Industry
Australian Electrical and Electronic Manufacturers Association
Australian Information Industry Association
Certification Forum of Australia
Department of Defence (Australia)
Department of Social Welfare New Zealand
Government Communications Security Bureau, New Zealand
Internet Industry Association
NSW Police Service
New Zealand Defence Force
Reserve Bank of Australia

Keeping Standards up-to-date

Standards are living documents which reflect progress in science, technology and systems. To maintain their currency, all Standards are periodically reviewed, and new editions are published. Between editions, amendments may be issued. Standards may also be withdrawn. It is important that readers assure themselves they are using a current Standard, which should include any amendments which may have been published since the Standard was purchased.

Detailed information about Standards can be found by visiting the Standards Web Shop at www.standards.com.au and looking up the relevant Standard in the on-line catalogue.

Alternatively, the printed Catalogue provides information current at 1 January each year, and the monthly magazine, *The Global Standard*, has a full listing of revisions and amendments published each month.

Australian Standards™ and other products and services developed by Standards Australia are published and distributed under contract by SAI Global, which operates the Standards Web Shop.

We also welcome suggestions for improvement in our Standards, and especially encourage readers to notify us immediately of any apparent inaccuracies or ambiguities. Contact us via email at mail@standards.org.au, or write to the Chief Executive, Standards Australia International Ltd, GPO Box 5420, Sydney, NSW 2001.

This Standard was issued in draft form for comment as DR 03550.

Australian Standard™

Information technology—Security techniques—Evaluation criteria for IT security

Part 2: Security functional requirements

First published as AS ISO/IEC 15408.2—2004.

COPYRIGHT

© Standards Australia International

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of the publisher.

Published by Standards Australia International Ltd
GPO Box 5420, Sydney, NSW 2001, Australia

ISBN 0 7337 5767 7

 **Documents**
Document Sharing Hub

PREFACE

This Standard was prepared by the Australian members of the Joint Standards Australia/Standards New Zealand Committee IT-012, Information systems—Security and identification technology. After consultation with stakeholders in both countries, Standards Australia and Standards New Zealand decided to develop this Standard as an Australian, rather than an Australian/New Zealand Standard.

This Standard is identical with, and has been reproduced from ISO/IEC 15408-2:1999, *Information technology—Security techniques—Evaluation criteria for IT security—Part 2: Security functional requirements*.

The objective of this Standard is to define security functional components - the basis for the Target of Evaluation (TOE), the requirements expressed in a Protection Profile (PP) or a Security Target (ST). These requirements describe the desired security behavior expected of a TOE and are intended to meet the security objectives as stated in a PP or an ST; security properties that users can detect by direct interaction with the TOE (i.e., inputs, outputs) or by the TOE's response to stimulus.

This Standard is Part 2 of AS ISO/IEC 15408, *Information technology—Security techniques—Evaluation criteria for IT security*, which is published in parts as follows:

Part 1: Introduction and general model

Part 2: Security functional requirements (this Standard)

Part 3: Security assurance requirements

The term ‘informative’ has been used in this Standard to define the application of the annex to which it applies. An ‘informative’ annex is only for information and guidance.

As this Standard is reproduced from an international standard, the following applies:

- (a) Its number appears on the cover and title page while the international standard number appears only on the cover.
- (b) In the source text ‘this part of ISO/IEC 15408’ should read ‘this part of AS ISO/IEC 15408’.
- (c) A full point substitutes for a comma when referring to a decimal marker.

CONTENTS

Page

1	Scope	1
1.1	Extending and maintaining functional requirements	1
1.2	Organisation of ISO/IEC 15408-2	2
1.3	Functional requirements paradigm	2
2	Security functional components	9
2.1	Overview	9
2.1.1	Class structure	9
2.1.2	Family structure	10
2.1.3	Component structure	11
2.1.4	Permitted functional component operations	13
2.2	Component catalogue	14
2.2.1	Component changes highlighting	15
3	Class FAU: Security audit	17
3.1	Security audit automatic response (FAU_ARP)	18
3.2	Security audit data generation (FAU_GEN)	19
3.3	Security audit analysis (FAU_SAA)	21
3.4	Security audit review (FAU_SAR)	24
3.5	Security audit event selection (FAU_SEL)	26
3.6	Security audit event storage (FAU_STG)	27
4	Class FCO: Communication	31
4.1	Non-repudiation of origin (FCO_NRO)	32
4.2	Non-repudiation of receipt (FCO_NRR)	34
5	Class FCS: Cryptographic support	37
5.1	Cryptographic key management (FCS_CKM)	38
5.2	Cryptographic operation (FCS_COP)	41
6	Class FDP: User data protection	43
6.1	Access control policy (FDP_ACC)	46
6.2	Access control functions (FDP_ACF)	48
6.3	Data authentication (FDP_DAU)	50
6.4	Export to outside TSF control (FDP_ETC)	52
6.5	Information flow control policy (FDP_IFC)	54
6.6	Information flow control functions (FDP_IFF)	56
6.7	Import from outside TSF control (FDP_ITC)	61
6.8	Internal TOE transfer (FDP_ITT)	63
6.9	Residual information protection (FDP_RIP)	66
6.10	Rollback (FDP_ROL)	68
6.11	Stored data integrity (FDP_SDI)	70

6.12	Inter-TSF user data confidentiality transfer protection (FDP_UCT)	72
6.13	Inter-TSF user data integrity transfer protection (FDP_UIT)	73
7	Class FIA: Identification and authentication	77
7.1	Authentication failures (FIA_AFL)	79
7.2	User attribute definition (FIA_ATD)	80
7.3	Specification of secrets (FIA_SOS)	81
7.4	User authentication (FIA_UAU)	83
7.5	User identification (FIA_UID)	88
7.6	User-subject binding (FIA_USB)	90
8	Class FMT: Security management	91
8.1	Management of functions in TSF (FMT_MOF)	93
8.2	Management of security attributes (FMT_MSA)	94
8.3	Management of TSF data (FMT_MTD)	97
8.4	Revocation (FMT_REV)	99
8.5	Security attribute expiration (FMT_SAE)	100
8.6	Security management roles (FMT_SMR)	101
9	Class FPR: Privacy	105
9.1	Anonymity (FPR_ANO)	106
9.2	Pseudonymity (FPR_PSE)	108
9.3	Unlinkability (FPR_UNL)	110
9.4	Unobservability (FPR_UNO)	111
10	Class FPT: Protection of the TSF	115
10.1	Underlying abstract machine test (FPT_AMT)	118
10.2	Fail secure (FPT_FLS)	119
10.3	Availability of exported TSF data (FPT_ITA)	120
10.4	Confidentiality of exported TSF data (FPT_ITC)	121
10.5	Integrity of exported TSF data (FPT_ITI)	122
10.6	Internal TOE TSF data transfer (FPT_ITT)	124
10.7	TSF physical protection (FPT_PHP)	127
10.8	Trusted recovery (FPT_RCV)	130
10.9	Replay detection (FPT_RPL)	133
10.10	Reference mediation (FPT_RVM)	134
10.11	Domain separation (FPT_SEP)	135
10.12	State synchrony protocol (FPT_SSP)	137
10.13	Time stamps (FPT_STM)	139
10.14	Inter-TSF TSF data consistency (FPT_TDC)	140
10.15	Internal TOE TSF data replication consistency (FPT_TRC)	141
10.16	TSF self test (FPT_TST)	142
11	Class FRU: Resource utilisation	145
11.1	Fault tolerance (FRU_FLT)	146
11.2	Priority of service (FRU_PRS)	148
11.3	Resource allocation (FRU_RSA)	150
12	Class FTA: TOE access	153
12.1	Limitation on scope of selectable attributes (FTA_LSA)	154

12.2	Limitation on multiple concurrent sessions (FTA_MCS)	155
12.3	Session locking (FTA_SSL)	157
12.4	TOE access banners (FTA_TAB)	160
12.5	TOE access history (FTA_TAH)	161
12.6	TOE session establishment (FTA_TSE)	162
13	Class FTP: Trusted path/channels	163
13.1	Inter-TSF trusted channel (FTP_ITC)	164
13.2	Trusted path (FTP_TRP)	166
Annex A	Security functional requirements application notes	169
A.1	Structure of the notes	169
A.1.1	Class structure	169
A.1.2	Family structure	170
A.1.3	Component structure	171
A.2	Dependency table	172
Annex B	Functional classes, families, and components	179
Annex C	Security audit (FAU)	181
C.1	Security audit automatic response (FAU_ARP)	183
C.2	Security audit data generation (FAU_GEN)	184
C.3	Security audit analysis (FAU_SAA)	187
C.4	Security audit review (FAU_SAR)	192
C.5	Security audit event selection (FAU_SEL)	194
C.6	Security audit event storage (FAU_STG)	195
Annex D	Communication (FCO)	199
D.1	Non-repudiation of origin (FCO_NRO)	200
D.2	Non-repudiation of receipt (FCO_NRR)	203
Annex E	Cryptographic support (FCS)	207
E.1	Cryptographic key management (FCS_CKM)	209
E.2	Cryptographic operation (FCS_COP)	212
Annex F	User data protection (FDP)	215
F.1	Access control policy (FDP_ACC)	220
F.2	Access control functions (FDP_ACF)	222
F.3	Data authentication (FDP_DAU)	225
F.4	Export to outside TSF control (FDP_ETC)	227
F.5	Information flow control policy (FDP_IFC)	229
F.6	Information flow control functions (FDP_IFF)	232
F.7	Import from outside TSF control (FDP_ITC)	238
F.8	Internal TOE transfer (FDP_ITT)	241
F.9	Residual information protection (FDP_RIP)	245
F.10	Rollback (FDP_ROL)	247
F.11	Stored data integrity (FDP_SDI)	249
F.12	Inter-TSF user data confidentiality transfer protection (FDP_UCT)	251
F.13	Inter-TSF user data integrity transfer protection (FDP_UIT)	252

Annex G	Identification and authentication (FIA)	255
G.1	Authentication failures (FIA_AFL)	257
G.2	User attribute definition (FIA_ATD)	259
G.3	Specification of secrets (FIA_SOS)	260
G.4	User authentication (FIA_UAU)	262
G.5	User identification (FIA_UID)	266
G.6	User-subject binding (FIA_USB)	267
Annex H	Security management (FMT)	269
H.1	Management of functions in TSF (FMT_MOF)	271
H.2	Management of security attributes (FMT_MSA)	273
H.3	Management of TSF data (FMT_MTD)	276
H.4	Revocation (FMT_REV)	278
H.5	Security attribute expiration (FMT_SAE)	279
H.6	Security management roles (FMT_SMR)	280
Annex I	Privacy (FPR)	283
I.1	Anonymity (FPR_ANO)	285
I.2	Pseudonymity (FPR_PSE)	287
I.3	Unlinkability (FPR_UNL)	292
I.4	Unobservability (FPR_UNO)	294
Annex J	Protection of the TSF (FPT)	299
J.1	Underlying abstract machine test (FPT_AMT)	303
J.2	Fail secure (FPT_FLS)	305
J.3	Availability of exported TSF data (FPT_ITA)	306
J.4	Confidentiality of exported TSF data (FPT_ITC)	307
J.5	Integrity of exported TSF data (FPT_ITI)	308
J.6	Internal TOE TSF data transfer (FPT_ITT)	310
J.7	TSF physical protection (FPT_PHP)	312
J.8	Trusted recovery (FPT_RCV)	314
J.9	Replay detection (FPT_RPL)	317
J.10	Reference mediation (FPT_RVM)	318
J.11	Domain separation (FPT_SEP)	319
J.12	State synchrony protocol (FPT_SSP)	321
J.13	Time stamps (FPT_STM)	322
J.14	Inter-TSF TSF data consistency (FPT_TDC)	323
J.15	Internal TOE TSF data replication consistency (FPT_TRC)	324
J.16	TSF self test (FPT_TST)	325
Annex K	Resource utilisation (FRU)	327
K.1	Fault tolerance (FRU_FLT)	328
K.2	Priority of service (FRU_PRS)	330
K.3	Resource allocation (FRU_RSA)	331
Annex L	TOE access (FTA)	333
L.1	Limitation on scope of selectable attributes (FTA_LSA)	334
L.2	Limitation on multiple concurrent sessions (FTA_MCS)	335
L.3	Session locking (FTA_SSL)	336
L.4	TOE access banners (FTA_TAB)	338

L.5	TOE access history (FTA_TAH)	339
L.6	TOE session establishment (FTA_TSE)	340
Annex M	Trusted path/channels (FTP)	341
M.1	Inter-TSF trusted channel (FTP_ITC)	342
M.2	Trusted path (FTP_TRP)	343

List of Figures

Figure 1.1 - Security functional requirements paradigm (Monolithic TOE)	3
Figure 1.2 - Diagram of security functions in a distributed TOE	4
Figure 1.3 - Relationship between user data and TSF data	7
Figure 1.4 - Relationship between “authentication data” and “secrets”	8
Figure 2.1 - Functional class structure	9
Figure 2.2 - Functional family structure	10
Figure 2.3 - Functional component structure	12
Figure 2.4 - Sample class decomposition diagram	15
Figure 3.1 - Security audit class decomposition	17
Figure 4.1 - Communication class decomposition	31
Figure 5.1 - Cryptographic support class decomposition	37
Figure 6.1 - User data protection class decomposition	44
Figure 6.2 - User data protection class decomposition (cont.)	45
Figure 7.1 - Identification and authentication class decomposition	78
Figure 8.1 - Security management class decomposition	92
Figure 9.1 - Privacy class decomposition	105
Figure 10.1 - Protection of the TSF class decomposition	116
Figure 10.2 - Protection of the TSF class decomposition (Cont.)	117
Figure 11.1 - Resource utilisation class decomposition	145
Figure 12.1 - TOE access class decomposition	153
Figure 13.1 - Trusted path/channels class decomposition	163
Figure A.1 - Functional class structure	169
Figure A.2 - Functional family structure for application notes	170
Figure A.3 - Functional component structure	171
Figure C.1 - Security audit class decomposition	182
Figure D.1 - Communication class decomposition	199
Figure E.1 - Cryptographic support class decomposition	207
Figure F.1 - User data protection class decomposition	217
Figure F.2 - User data protection class decomposition (cont.)	218
Figure G.1 - Identification and authentication class decomposition	256
Figure H.1 - Security management class decomposition	270
Figure I.1 - Privacy class decomposition	283
Figure J.1 - Protection of the TSF class decomposition	300
Figure J.2 - Protection of the TSF class decomposition (Cont.)	301
Figure K.1 - Resource utilisation class decomposition	327
Figure L.1 - TOE access class decomposition	333
Figure M.1 - Trusted path/channels class decomposition	341

AUSTRALIAN STANDARD

Information technology — Security techniques — Evaluation criteria for IT security —**Part 2:
Security functional requirements****1 Scope**

Security functional components, as defined in this part of ISO/IEC 15408, are the basis for the TOE IT security functional requirements expressed in a Protection Profile (PP) or a Security Target (ST). These requirements describe the desired security behaviour expected of a Target of Evaluation (TOE) and are intended to meet the security objectives as stated in a PP or an ST. These requirements describe security properties that users can detect by direct interaction with the TOE (i.e. inputs, outputs) or by the TOE's response to stimulus.

Security functional components express security requirements intended to counter threats in the assumed operating environment of the TOE and/or cover any identified organisational security policies and assumptions.

The audience for this part of ISO/IEC 15408 includes consumers, developers, and evaluators of secure IT systems and products. ISO/IEC 15408-1 clause 3 provides additional information on the target audience of ISO/IEC 15408, and on the use of the standard by the groups that comprise the target audience. These groups may use this part of ISO/IEC 15408 as follows:

- Consumers who use ISO/IEC 15408-2 when selecting components to express functional requirements to satisfy the security objectives expressed in a PP or ST. ISO/IEC 15408-1 subclause 4.3 provides more detailed information on the relationship between security objectives and security requirements.
- Developers, who respond to actual or perceived consumer security requirements in constructing a TOE, may find a standardised method to understand those requirements in this part of ISO/IEC 15408. They can also use the contents of this part of ISO/IEC 15408 as a basis for further defining the TOE security functions and mechanisms that comply with those requirements.
- Evaluators, who use the functional requirements defined in this part of ISO/IEC 15408 in verifying that the TOE functional requirements expressed in the PP or ST satisfy the IT security objectives and that all dependencies are accounted for and shown to be satisfied. Evaluators also should use this part of ISO/IEC 15408 to assist in determining whether a given TOE satisfies stated requirements.

1.1 Extending and maintaining functional requirements

ISO/IEC 15408 and the associated security functional requirements described herein are not meant to be a definitive answer to all the problems of IT security. Rather, the standard offers a set of well understood security functional requirements that can be used to create trusted products or systems

reflecting the needs of the market. These security functional requirements are presented as the current state of the art in requirements specification and evaluation.

This part of ISO/IEC 15408 does not presume to include all possible security functional requirements but rather contains those that are known and agreed to be of value by the ISO/IEC 15408-2 authors at the time of release.

Since the understanding and needs of consumers may change, the functional requirements in this part of ISO/IEC 15408 will need to be maintained. It is envisioned that some PP/ST authors may have security needs not (yet) covered by the functional requirement components in ISO/IEC 15408-2. In those cases the PP/ST author may choose to consider using functional requirements not taken from the standard (referred to as extensibility), as explained in Annexes B and C of ISO/IEC 15408-1.

1.2 Organisation of ISO/IEC 15408-2

Clause 1 is the introductory material for ISO/IEC 15408-2.

Clause 2 introduces the catalogue of ISO/IEC 15408-2 functional components while clauses 3 through 13 describe the functional classes.

Annex A provides additional information of interest to potential users of the functional components including a complete cross reference table of the functional component dependencies.

Annexes B through M provide the application notes for the functional classes. They are a repository for informative supporting material for the users of this part of ISO/IEC 15408, which may help them to apply relevant operations and select appropriate audit or documentation information.

Those who author PPs or STs should refer to Clause 2 of ISO/IEC 15408-1 for relevant structures, rules, and guidance:

- ISO/IEC 15408-1, clause 2 defines the terms used in ISO/IEC 15408.
- ISO/IEC 15408-1, Annex B defines the structure for PPs.
- ISO/IEC 15408-1, Annex C defines the structure for STs.

1.3 Functional requirements paradigm

This subclause describes the paradigm used in the security functional requirements of this part of ISO/IEC 15408. Figures 1.1 and 1.2 depict some of the key concepts of the paradigm. This subclause provides descriptive text for those figures and for other key concepts not depicted. Key concepts discussed are highlighted in **bold/italics**. This subclause is not intended to replace or supersede any of the terms found in the ISO/IEC 15408 glossary in ISO/IEC 15408-1, clause 2.

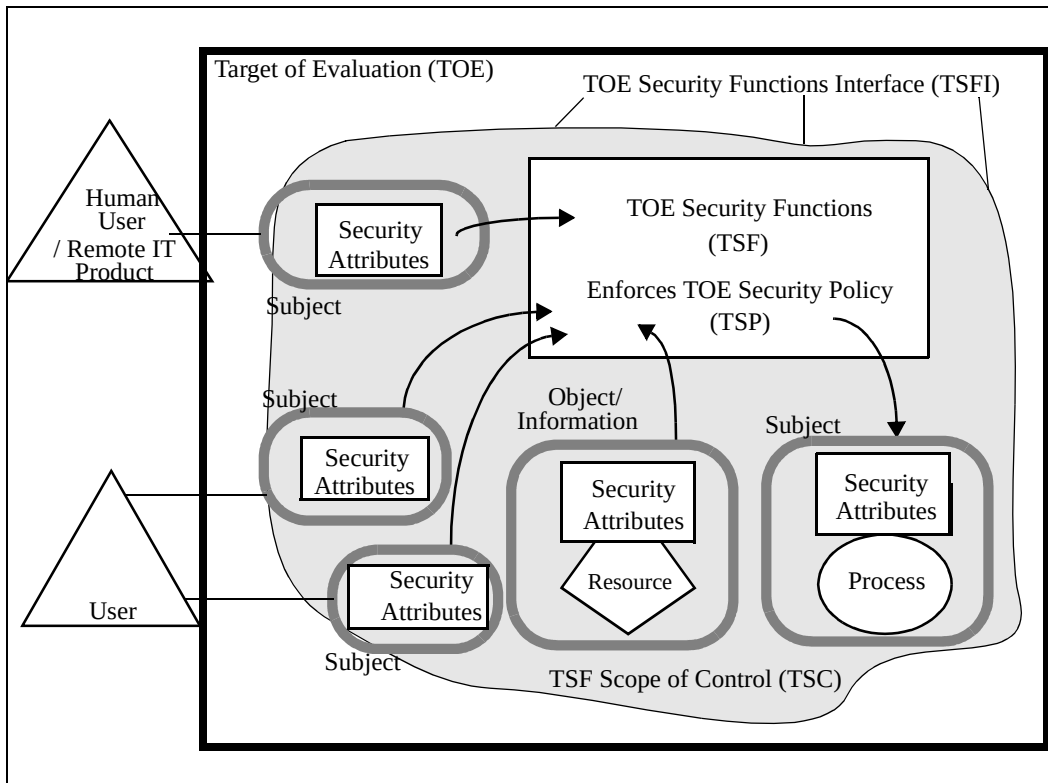


Figure 1.1 - Security functional requirements paradigm (Monolithic TOE)

This part of ISO/IEC 15408 is a catalogue of security functional requirements that can be specified for a **Target of Evaluation (TOE)**. A TOE is an IT product or system (along with user and administrator guidance documentation) containing resources such as electronic storage media (e.g. disks), peripheral devices (e.g. printers), and computing capacity (e.g. CPU time) that can be used for processing and storing information and is the subject of an evaluation.

TOE evaluation is concerned primarily with ensuring that a defined **TOE Security Policy (TSP)** is enforced over the TOE resources. The TSP defines the rules by which the TOE governs access to its resources, and thus all information and services controlled by the TOE.

The TSP is, in turn, made up of multiple **Security Function Policies (SFPs)**. Each SFP has a scope of control, that defines the subjects, objects, and operations controlled under the SFP. The SFP is implemented by a **Security Function (SF)**, whose mechanisms enforce the policy and provide necessary capabilities.

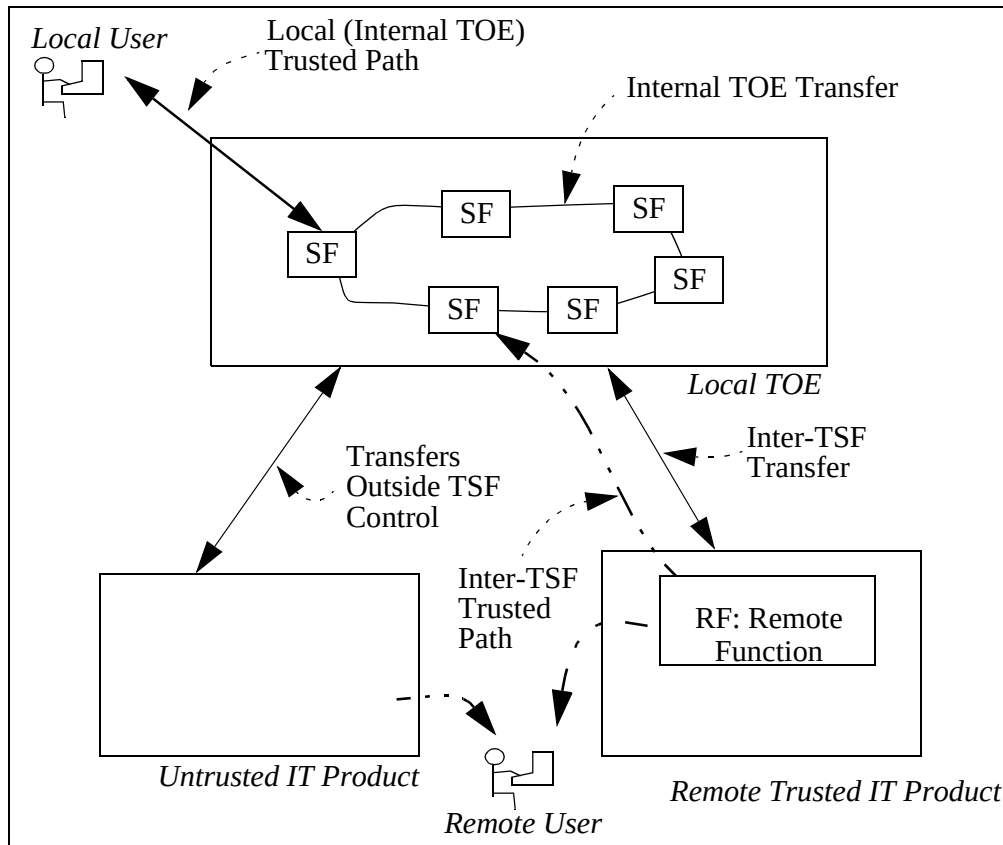


Figure 1.2 - Diagram of security functions in a distributed TOE

Those portions of a TOE that must be relied on for the correct enforcement of the TSP are collectively referred to as the **TOE Security Functions (TSF)**. The TSF consists of all hardware, software, and firmware of a TOE that is either directly or indirectly relied upon for security enforcement.

A **reference monitor** is an abstract machine that enforces the access control policies of a TOE. A **reference validation mechanism** is an implementation of the reference monitor concept that possesses the following properties: tamperproof, always invoked, and simple enough to be subjected to thorough analysis and testing. The **TSF** may consist of a reference validation mechanism and/or other security functions necessary for the operation of the TOE.

The TOE may be a monolithic product containing hardware, firmware, and software.

Alternatively a TOE may be a distributed product that consists internally of multiple separated parts. Each of these parts of the TOE provides a particular service for the TOE, and is connected to the other parts of the TOE through an **internal communication channel**. This channel can be as small as a processor bus, or may encompass a network internal to the TOE.

When the TOE consists of multiple parts, each part of the TOE may have its own part of the TSF which exchanges user and TSF data over internal communication channels with other parts of the TSF. This interaction is called **internal TOE transfer**. In this case the separate parts of the TSF abstractly form the composite TSF, which enforces the TSP.

TOE interfaces may be localised to the particular TOE, or they may allow interaction with other IT products over **external communication channels**. These external interactions with other IT products may take two forms:

- a) The security policy of the ‘remote trusted IT product’ and the TSP of the local TOEs have been administratively coordinated and evaluated. Exchanges of information in this situation are called **inter-TSF transfers**, as they are between the TSFs of distinct trusted products.
- b) The remote IT product may not be evaluated, indicated in Figure 1.2 as ‘untrusted IT product’, therefore its security policy is unknown. Exchanges of information in this situation are called **transfers outside TSF control**, as there is no TSF (or its policy characteristics are unknown) on the remote IT product.

The set of interactions that can occur with or within a TOE and are subject to the rules of the TSP is called the **TSF Scope of Control (TSC)**. The TSC encompasses a defined set of interactions based on subjects, objects, and operations within the TOE, but it need not encompass all resources of a TOE.

The set of interfaces, whether interactive (man-machine interface) or programmatic (application programming interface), through which resources are accessed that are mediated by the TSF, or information is obtained from the TSF, is referred to as the **TSF Interface (TSFI)**. The TSFI defines the boundaries of the TOE functions that provide for the enforcement of the TSP.

Users are outside of the TOE, and therefore outside of the TSC. However, in order to request that services be performed by the TOE, users interact with the TOE through the TSFI. There are two types of users of interest to the ISO/IEC 15408-2 security functional requirements: **human users** and **external IT entities**. Human users are further differentiated as **local human users**, meaning they interact directly with the TOE via TOE devices (e.g. workstations), or **remote human users**, meaning they interact indirectly with the TOE through another IT product.

A period of interaction between users and the TSF is referred to as a user **session**. Establishment of user sessions can be controlled based on a variety of considerations, for example: user authentication, time of day, method of accessing the TOE, and number of allowed concurrent sessions per user.

This part of ISO/IEC 15408 uses the term **authorised** to signify a user who possesses the rights and/or privileges necessary to perform an operation. The term **authorised user**, therefore, indicates that it is allowable for a user to perform an operation as defined by the TSP.

To express requirements that call for the separation of administrator duties, the relevant ISO/IEC 15408-2 security functional components (from family FMT_SMR) explicitly state that administrative **roles** are required. A role is a pre-defined set of rules establishing the allowed interactions between a user and the TOE. A TOE may support the definition of any number of roles. For example, roles related to the secure operation of a TOE may include “Audit Administrator” and “User Accounts Administrator”.

TOEs contain resources that may be used for the processing and storing of information. The primary goal of the TSF is the complete and correct enforcement of the TSP over the resources and information that the TOE controls.

TOE resources can be structured and utilised in many different ways. However, ISO/IEC 15408-2 makes a specific distinction that allows for the specification of desired security properties. All entities that can be created from resources can be characterised in one of two ways. The entities may be active, meaning that they are the cause of actions that occur internal to the TOE and cause operations to be performed on information. Alternatively, the entities may be passive, meaning that they are either the container from which information originates or to which information is stored.

Active entities are referred to as **subjects**. Several types of subjects may exist within a TOE:

- a) those acting on behalf of an authorised user and which are subject to all the rules of the TSP (e.g. UNIX processes);
- b) those acting as a specific functional process that may in turn act on behalf of multiple users (e.g. functions as might be found in client/server architectures); or
- c) those acting as part of the TOE itself (e.g. trusted processes).

ISO/IEC 15408-2 addresses the enforcement of the TSP over types of subjects as those listed above.

Passive entities (i.e. information containers) are referred to in the ISO/IEC 15408-2 security functional requirements as **objects**. Objects are the targets of operations that may be performed by subjects. In the case where a subject (an active entity) is the target of an operation (e.g. interprocess communication), a subject may also be acted on as an object.

Objects can contain **information**. This concept is required to specify information flow control policies as addressed in the FDP class.

Users, subjects, information and objects possess certain **attributes** that contain information that allows the TOE to behave correctly. Some attributes, such as file names, may be intended to be informational (i.e. to increase the user-friendliness of the TOE) while others, such as access control information, may exist specifically for the enforcement of the TSP. These latter attributes are generally referred to as '**security attributes**'. The word attribute will be used as a shorthand in this part of ISO/IEC 15408 for the word 'security attribute', unless otherwise indicated. However, no matter what the intended purpose of the attribute information, it may be necessary to have controls on attributes as dictated by the TSP.

Data in a TOE is categorised as either user data or TSF data. Figure 1.3 depicts this relationship. **User Data** is information stored in TOE resources that can be operated upon by users in accordance with the TSP and upon which the TSF places no special meaning. For example, the contents of an electronic mail message is user data. **TSF Data** is information used by the TSF in making TSP decisions. TSF Data may be influenced by users if allowed by the TSP. Security attributes, authentication data and access control list entries are examples of TSF data.

There are several SFPs that apply to data protection such as **access control SFPs** and **information flow control SFPs**. The mechanisms that implement access control SFPs base their policy decisions on attributes of the subjects, objects and operations within the scope of control. These attributes are used in the set of rules that govern operations that subjects may perform on objects.

The mechanisms that implement information flow control SFPs base their policy decisions on the attributes of the subjects and information within the scope of control and the set of rules that govern the operations by subjects on information. The attributes of the information, which may be associated with the attributes of the container (or may not, as in the case of a multi-level database) stay with the information as it moves.

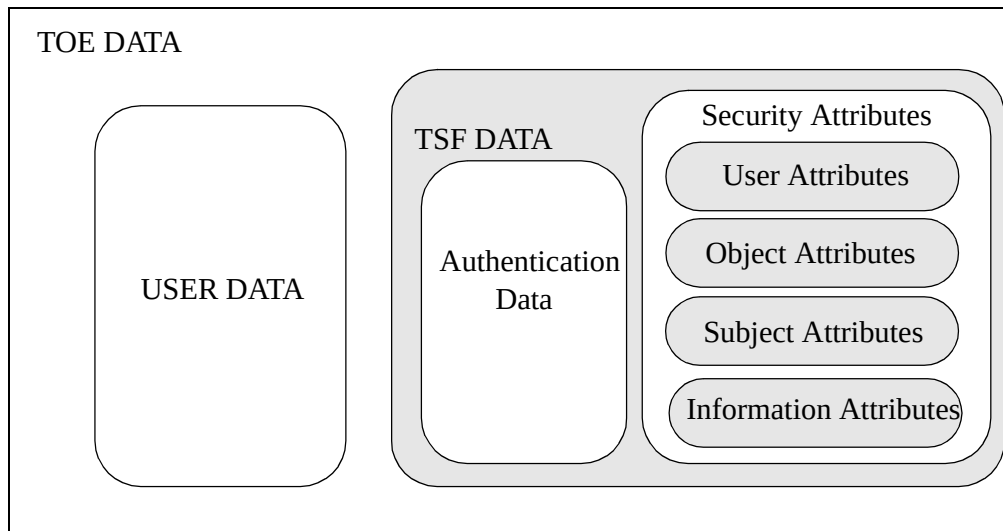


Figure 1.3 - Relationship between user data and TSF data

Two specific types of TSF data addressed by ISO/IEC 15408-2 can be, but are not necessarily, the same. These are **authentication data** and **secrets**.

Authentication data is used to verify the claimed identity of a user requesting services from a TOE. The most common form of authentication data is the password, which depends on being kept secret in order to be an effective security mechanism. However, not all forms of authentication data need to be kept secret. Biometric authentication devices (e.g. fingerprint readers, retinal scanners) do not rely on the fact that the data is kept secret, but rather that the data is something that only one user possesses and that cannot be forged.

The term secrets, as used in ISO/IEC 15408-2 functional requirements, while applicable to authentication data, is intended to also be applicable to other types of data that must be kept secret in order to enforce a specific SFP. For example, a trusted channel mechanism that relies on cryptography to preserve the confidentiality of information being transmitted via the channel can only be as strong as the method used to keep the cryptographic keys secret from unauthorised disclosure.

Therefore, some, but not all, authentication data needs to be kept secret and some, but not all, secrets are used as authentication data. Figure 1.4 shows this relationship between secrets and authentication data. In the Figure the types of data typically encountered in the authentication data and the secrets sections are indicated.

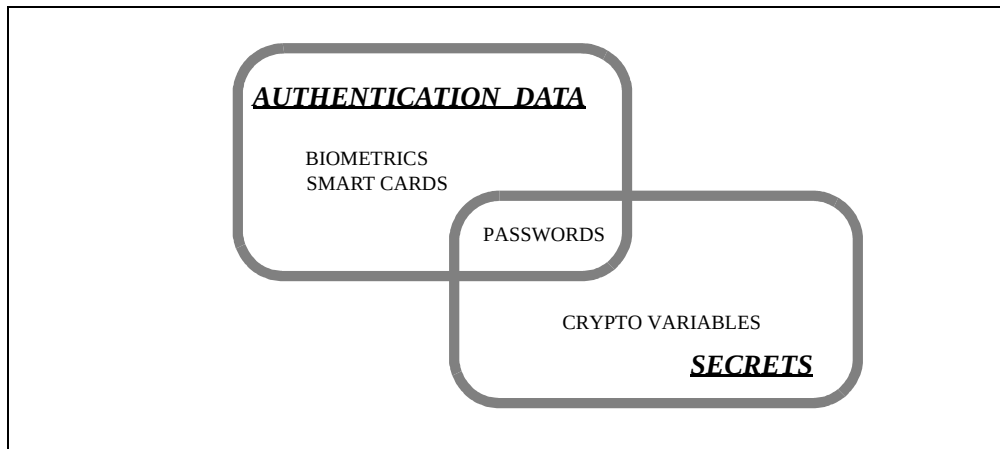


Figure 1.4 - Relationship between “authentication data” and “secrets”

2 Security functional components

2.1 Overview

This clause defines the content and presentation of the functional requirements of ISO/IEC 15408, and provides guidance on the organisation of the requirements for new components to be included in an ST. The functional requirements are expressed in classes, families, and components.

2.1.1 Class structure

Figure 2.1 illustrates the functional class structure in diagrammatic form. Each functional class includes a class name, class introduction, and one or more functional families.

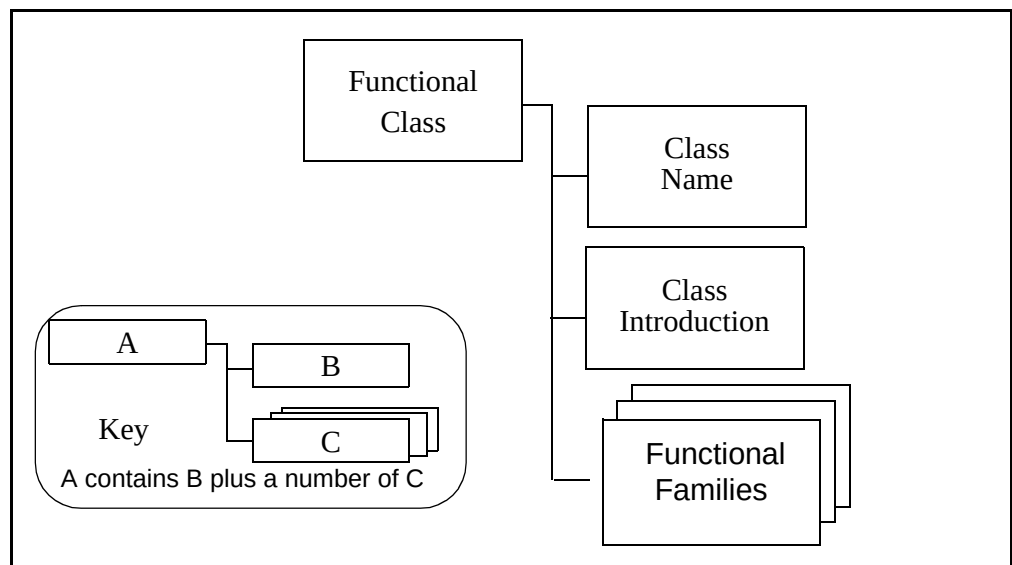


Figure 2.1 - Functional class structure

2.1.1.1 Class name

The class name subclause provides information necessary to identify and categorise a functional class. Every functional class has a unique name. The categorical information consists of a short name of three characters. The short name of the class is used in the specification of the short names of the families of that class.

2.1.1.2 Class introduction

The class introduction expresses the common intent or approach of those families to satisfy security objectives. The definition of functional classes does not reflect any formal taxonomy in the specification of the requirements.

The class introduction provides a figure describing the families in this class and the hierarchy of the components in each family, as explained in 2.2.

2.1.2 Family structure

Figure 2.2 illustrates the functional family structure in diagrammatic form.

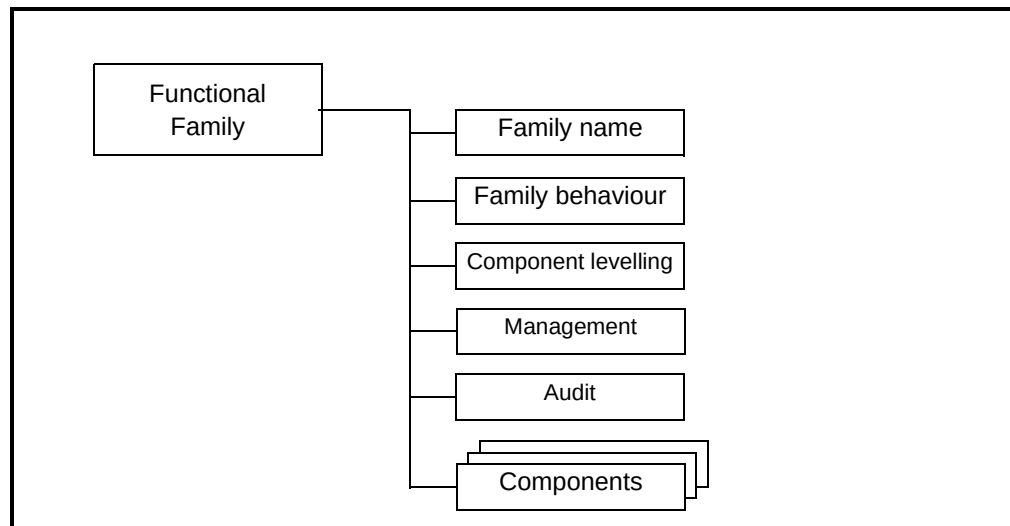


Figure 2.2 - Functional family structure

2.1.2.1 Family name

The family name subclause provides categorical and descriptive information necessary to identify and categorise a functional family. Every functional family has a unique name. The categorical information consists of a short name of seven characters, with the first three identical to the short name of the class followed by an underscore and the short name of the family as follows XXX_YYY. The unique short form of the family name provides the principal reference name for the components.

2.1.2.2 Family behaviour

The family behaviour is the narrative description of the functional family stating its security objective and a general description of the functional requirements. These are described in greater detail below:

- a) The *security objectives* of the family address a security problem that may be solved with the help of a TOE that incorporates a component of this family;
- b) The description of the *functional requirements* summarises all the requirements that are included in the component(s). The description is aimed at authors of PPs, STs and functional packages who wish to assess whether the family is relevant to their specific requirements.

2.1.2.3 Component levelling

Functional families contain one or more components, any one of which can be selected for inclusion in PPs, STs and functional packages. The goal of this section is to provide information to users in selecting an appropriate functional component once the family has been identified as being a necessary or useful part of their security requirements.

This section of the functional family description describes the components available, and their rationale. The exact details of the components are contained within each component.

The relationships between components within a functional family may or may not be hierarchical. A component is hierarchical to another if it offers more security.

As explained in 2.2 the descriptions of the families provide a graphical overview of the hierarchy of the components in a family.

2.1.2.4 Management

The *management* requirements contain information for the PP/ST authors to consider as management activities for a given component. The management requirements are detailed in components of the management class (FMT).

A PP/ST author may select the indicated management requirements or may include other management requirements not listed. As such the information should be considered informative.

2.1.2.5 Audit

The *audit* requirements contain auditable events for the PP/ST authors to select, if requirements from the class FAU, Security audit, are included in the PP/ST. These requirements include security relevant events in terms of the various levels of detail supported by the components of the FAU_GEN Security audit data generation family. For example, an audit note might include actions that are in terms of: Minimal - successful use of the security mechanism; Basic - any use of the security mechanism as well as relevant information regarding the security attributes involved; Detailed - any configuration changes made to the mechanism, including the actual configuration values before and after the change.

It should be observed that the categorisation of auditable events is hierarchical. For example, when Basic Audit Generation is desired, all auditable events identified as being both Minimal and Basic should be included in the PP/ST through the use of the appropriate assignment operation, except when the higher level event simply provides more detail than the lower level event. When Detailed Audit Generation is desired, all identified auditable events (Minimal, Basic and Detailed) should be included in the PP/ST.

In the class FAU the rules governing the audit are explained in more detail.

2.1.3 Component structure

Figure 2.3 illustrates the functional component structure.

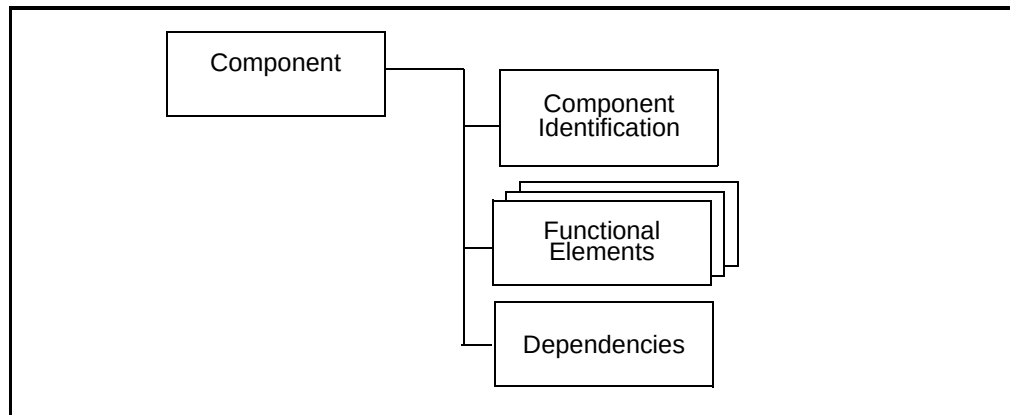


Figure 2.3 - Functional component structure

2.1.3.1 Component identification

The component identification subclause provides descriptive information necessary to identify, categorise, register and cross-reference a component. The following is provided as part of every functional component:

A unique name. The name reflects the purpose of the component.

A short name. A unique short form of the functional component name. This short name serves as the principal reference name for the categorisation, registration and cross-referencing of the component. This short name reflects the class and family to which the component belongs and the component number within the family.

A hierarchical-to list. A list of other components that this component is hierarchical to and for which this component can be used to satisfy dependencies to the listed components.

2.1.3.2 Functional elements

A set of elements is provided for each component. Each element is individually defined and is self-contained.

A functional element is a security functional requirement that if further divided would not yield a meaningful evaluation result. It is the smallest security functional requirement identified and recognised in ISO/IEC 15408.

When building packages, PPs and/or STs, it is not permitted to select only one or more elements from a component. The complete set of elements of a component must be selected for inclusion in a PP, ST or package.

A unique short form of the functional element name is provided. For example the requirement name FDP_IFF.4.2 reads as follows: F - functional requirement, DP - class “User data protection”, _IFF - family “Information flow control functions”, .4 - 4th component named “Partial elimination of illicit information flows”, .2 - 2nd element of the component.

2.1.3.3 Dependencies

Dependencies among functional components arise when a component is not self sufficient and relies upon the functionality of, or interaction with, another component for its own proper functioning.

Each functional component provides a complete list of dependencies to other functional and assurance components. Some components may list “No dependencies”. The components depended upon may in turn have dependencies on other components. The list provided in the components will be the direct dependencies. That is only references to the functional requirements that are required for this requirement to perform its job properly. The indirect dependencies, that is the dependencies that result from the depended upon components can be found in Annex A of this part of ISO/IEC 15408. It is noted that in some cases the dependency is optional in that a number of functional requirements are provided, where each one of them would be sufficient to satisfy the dependency (see for example FDP_UIT.1).

The dependency list identifies the minimum functional or assurance components needed to satisfy the security requirements associated with an identified component. Components that are hierarchical to the identified component may also be used to satisfy the dependency.

The dependencies indicated in ISO/IEC 15408-2 are normative. They must be satisfied within a PP/ST. In specific situations the indicated dependencies might not be applicable. The PP/ST author, by providing the rationale why it is not applicable, may leave the depended upon component out of the package, PP or ST.

2.1.4 Permitted functional component operations

The functional components used in the definition of the requirements in a PP, an ST or a functional package may be exactly as specified in clauses 3 to 13 of this part of ISO/IEC 15408, or they may be tailored to meet a specific security objective. However, selecting and tailoring these functional components is complicated by the fact that identified component dependencies must be considered. Thus, this tailoring is restricted to an approved set of operations.

A list of permitted operations is included with each functional component. Not all operations are permitted on all functional components.

The permitted operations are selected from the following set:

- iteration: allows a component to be used more than once with varying operations,
- assignment: allows the specification of an identified parameter,
- selection: allows the specification of one or more elements from a list,
- refinement: allows the addition of details.

2.1.4.1 Iteration

Where necessary to cover different aspects of the same requirement (e.g. identification of more than one type of user), repetitive use of the same component from this part of ISO/IEC 15408 to cover each aspect is permitted.

2.1.4.2 Assignment

Some functional component elements contain parameters or variables that enable the PP/ST author to specify a policy or a set of values for incorporation into the PP or ST to meet a specific security objective. These elements clearly identify each parameter and constraint on values that may be assigned to that parameter.

Any aspect of an element whose acceptable values can be unambiguously described or enumerated can be represented by a parameter. The parameter may be an attribute or rule that narrows the requirement to a specific value or range of values. For instance, based on a specified security objective, the functional component element may state that a given operation should be performed a number of times. In this case, the assignment would provide the number, or range of numbers, to be used in the parameter.

2.1.4.3 Selection

This is the operation of picking one or more items from a list in order to narrow the scope of a component element.

2.1.4.4 Refinement

For all functional component elements the PP/ST author is permitted to limit the set of acceptable implementations by specifying additional detail in order to meet a security objective. Refinement of an element consists of adding these technical details.

Within a ST, the meanings of the terms subject and object might need to be explained for the TOE to be meaningful, and are therefore subject to refinement.

Like the other operations, refinement does not levy any completely new requirements. It applies an elaboration, interpretation, or a special meaning to a requirement, rule, constant or condition based on security objectives. Refinement shall only further restrict the set of possible acceptable functions or mechanisms to implement the requirements, but never increase it. Refinement does not allow new requirements to be created, and therefore does not increase the list of dependencies associated with a component. The PP/ST author must be careful that the dependency needs of other requirements that depend on this requirement, are satisfied.

2.2 Component catalogue

The grouping of the components in this part of ISO/IEC 15408 does not reflect any formal taxonomy.

This part of ISO/IEC 15408 contains classes of families and components, which are rough groupings on the basis of related function or purpose, presented in alphabetic order. At the start of each class is an informative diagram that indicates the taxonomy of each class, indicating the families in each class and the components in each family. The diagram is a useful indicator of the hierarchical relationship that may exist between components.

In the description of the functional components, a section identifies the dependencies between the component and any other components.

In each class a figure describing the family hierarchy similar to Figure 2.4, is provided. In Figure 2.4. the first family, Family 1, contains three hierarchical components, where component 2 and component 3 can both be used to satisfy dependencies on component 1. Component 3 is hierarchical to component 2 and can also be used to satisfy dependencies on component 2.

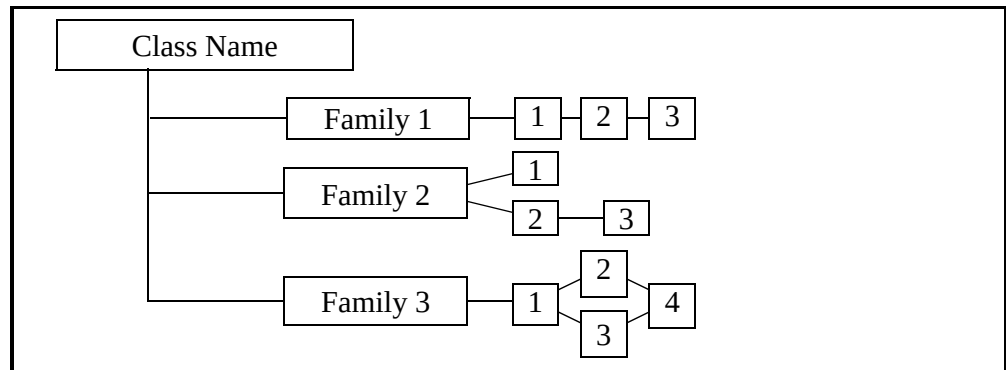


Figure 2.4 - Sample class decomposition diagram

In Family 2 there are three components not all of which are hierarchical. Components 1 and 2 are hierarchical to no other components. Component 3 is hierarchical to component 2, and can be used to satisfy dependencies on component 2, but not to satisfy dependencies on component 1.

In Family 3, components 2, 3, and 4 are hierarchical to component 1. Components 2 and 3 are both hierarchical to component 1, but non-comparable. Component 4 is hierarchical to both component 2 and component 3.

These diagrams are meant to complement the text of the families and make identification of the relationships easier. They do not replace the “Hierarchical to:” note in each component that is the mandatory claim of hierarchy for each component.

2.2.1 Component changes highlighting

The relationship between components within a family is highlighted using a **bolding** convention. This bolding convention calls for the bolding of all new requirements. For hierarchical components, requirements and/or dependencies are bolded when they are enhanced or modified beyond the requirements of the previous component. In addition, any new or enhanced permitted operations beyond the previous component are also highlighted using **bold** type.

3 Class FAU: Security audit

Security auditing involves recognising, recording, storing, and analysing information related to security relevant activities (i.e. activities controlled by the TSP). The resulting audit records can be examined to determine which security relevant activities took place and whom (which user) is responsible for them.

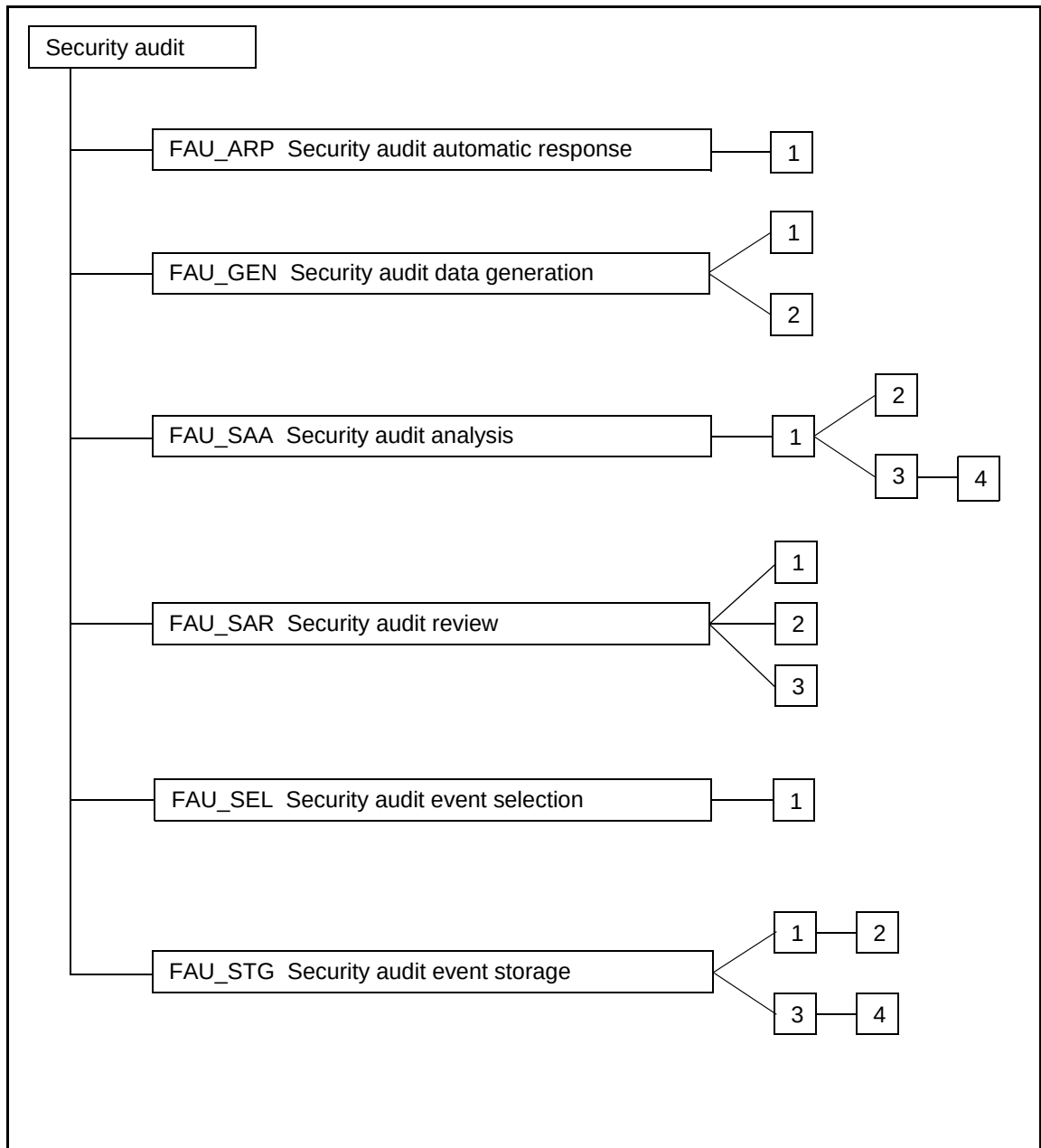


Figure 3.1 - Security audit class decomposition

3.1 Security audit automatic response (FAU_ARP)

Family behaviour

This family defines the response to be taken in case of detected events indicative of a potential security violation.

Component levelling

FAU_ARP Security audit automatic response

1

At FAU_ARP.1 Security alarms, the TSF shall take actions in case a potential security violation is detected.

Management: FAU_ARP.1

The following actions could be considered for the management functions in FMT:

- a) the management (addition, removal, or modification) of actions.

Audit: FAU_ARP.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- a) Minimal: Actions taken due to imminent security violations.

FAU_ARP.1 Security alarms

Hierarchical to: No other components.

FAU_ARP.1.1 The TSF shall take [assignment: *list of the least disruptive actions*] upon detection of a potential security violation.

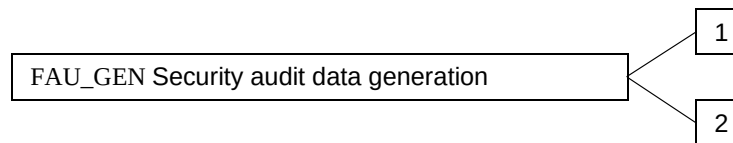
Dependencies: FAU_SAA.1 Potential violation analysis

3.2 Security audit data generation (FAU_GEN)

Family behaviour

This family defines requirements for recording the occurrence of security relevant events that take place under TSF control. This family identifies the level of auditing, enumerates the types of events that shall be auditable by the TSF, and identifies the minimum set of audit-related information that should be provided within various audit record types.

Component levelling



FAU_GEN.1 Audit data generation defines the level of auditable events, and specifies the list of data that shall be recorded in each record.

At FAU_GEN.2 User identity association, the TSF shall associate auditable events to individual user identities.

Management: FAU_GEN.1, FAU_GEN.2

There are no management activities foreseen.

Audit: FAU_GEN.1, FAU_GEN.2

There are no actions identified that should be auditable if FAU_GEN Security audit data generation is included in the PP/ST.

FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

- a) **Start-up and shutdown of the audit functions;**
- b) **All auditable events for the [selection: *minimum, basic, detailed, not specified*] level of audit; and**
- c) **[assignment: *other specifically defined auditable events*].**

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

- a) **Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and**
- b) **For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [assignment: *other audit relevant information*]**

Dependencies: **FPT_STM.1 Reliable time stamps**

FAU_GEN.2 User identity association

Hierarchical to: No other components.

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

Dependencies: **FAU_GEN.1 Audit data generation**
FIA_UID.1 Timing of identification

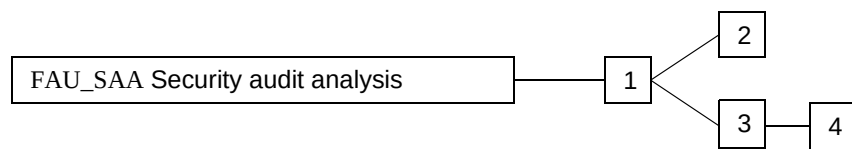
3.3 Security audit analysis (FAU_SAA)

Family behaviour

This family defines requirements for automated means that analyse system activity and audit data looking for possible or real security violations. This analysis may work in support of intrusion detection, or automatic response to an imminent security violation.

The actions to be taken based on the detection can be specified using the FAU_ARP family as desired.

Component levelling



In FAU_SAA.1 Potential violation analysis, basic threshold detection on the basis of a fixed rule set is required.

In FAU_SAA.2 Profile based anomaly detection, the TSF maintains individual *profiles* of system usage, where a profile represents the historical patterns of usage performed by members of the *profile target group*. A profile target group refers to a group of one or more individuals (e.g. a single user, users who share a group ID or group account, users who operate under an assigned role, users of an entire system or network node) who interact with the TSF. Each member of a profile target group is assigned an individual *suspicion rating* that represents how well that member's current activity corresponds to the established patterns of usage represented in the profile. This analysis can be performed at runtime or during a post-collection batch-mode analysis.

In FAU_SAA.3 Simple attack heuristics, the TSF shall be able to detect the occurrence of signature events that represent a significant threat to TSP enforcement. This search for signature events may occur in real-time or during a post-collection batch-mode analysis.

In FAU_SAA.4 Complex attack heuristics, the TSF shall be able to represent and detect multi-step intrusion scenarios. The TSF is able to compare system events (possibly performed by multiple individuals) against event sequences known to represent entire intrusion scenarios. The TSF shall be able to indicate when a signature event or event sequence is found that indicates a potential violation of the TSP.

Management: FAU_SAA.1

The following actions could be considered for the management functions in FMT:

- a) maintenance of the rules by (adding, modifying, deletion) of rules from the set of rules.

Management: FAU_SAA.2

The following actions could be considered for the management functions in FMT:

- a) maintenance (deletion, modification, addition) of the group of users in the profile target group.

Management: FAU_SAA.3

The following actions could be considered for the management functions in FMT:

- a) maintenance (deletion, modification, addition) of the subset of system events.

Management: FAU_SAA.4

The following actions could be considered for the management functions in FMT:

- a) maintenance (deletion, modification, addition) of the subset of system events;
- b) maintenance (deletion, modification, addition) of the set of sequence of system events.

Audit: FAU_SAA.1, FAU_SAA.2, FAU_SAA.3, FAU_SAA.4

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- a) Minimal: Enabling and disabling of any of the analysis mechanisms;
- b) Minimal: Automated responses performed by the tool.

FAU_SAA.1 Potential violation analysis

Hierarchical to: No other components.

FAU_SAA.1.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:

- a) **Accumulation or combination of [assignment: *subset of defined auditable events*] known to indicate a potential security violation;**
- b) **[assignment: *any other rules*].**

Dependencies: FAU_GEN.1 Audit data generation

FAU_SAA.2 Profile based anomaly detection

Hierarchical to: FAU_SAA.1

FAU_SAA.2.1 The TSF shall be able to maintain profiles of system usage, where an individual profile represents the historical patterns of usage performed by the member(s) of [assignment: *the profile target group*].

FAU_SAA.2.2 The TSF shall be able to maintain a suspicion rating associated with each user whose activity is recorded in a profile, where the suspicion rating represents the degree to which the user's current activity is found inconsistent with the established patterns of usage represented in the profile.

FAU_SAA.2.3 The TSF shall be able to indicate an imminent violation of the TSP when a user's suspicion rating exceeds the following threshold conditions [assignment: *conditions under which anomalous activity is reported by the TSF*].

Dependencies: FIA_UID.1 Timing of identification

FAU_SAA.3 Simple attack heuristics

Hierarchical to: FAU_SAA.1

FAU_SAA.3.1 The TSF shall be able to maintain an internal representation of the following signature events [assignment: *a subset of system events*] that may indicate a violation of the TSP.

FAU_SAA.3.2 The TSF shall be able to compare the signature events against the record of system activity discernible from an examination of [assignment: *the information to be used to determine system activity*].

FAU_SAA.3.3 The TSF shall be able to indicate an imminent violation of the TSP when a system event is found to match a signature event that indicates a potential violation of the TSP.

Dependencies: No dependencies.

FAU_SAA.4 Complex attack heuristics

Hierarchical to: FAU_SAA.3

FAU_SAA.4.1 The TSF shall be able to maintain an internal representation of the **following event sequences of known intrusion scenarios** [assignment: *list of sequences of system events whose occurrence are representative of known penetration scenarios*] and the following signature events [assignment: *a subset of system events*] that may indicate a potential violation of the TSP.

FAU_SAA.4.2 The TSF shall be able to compare the signature events **and event sequences** against the record of system activity discernible from an examination of [assignment: *the information to be used to determine system activity*].

FAU_SAA.4.3 The TSF shall be able to indicate an imminent violation of the TSP when **system activity** is found to match a signature event **or event sequence** that indicates a potential violation of the TSP.

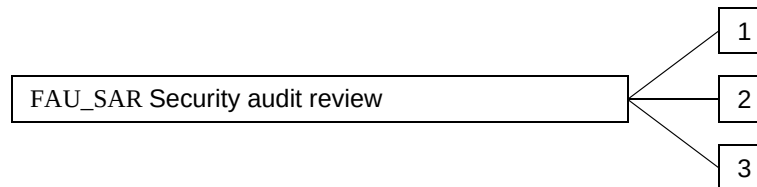
Dependencies: No dependencies.

3.4 Security audit review (FAU_SAR)

Family behaviour

This family defines the requirements for audit tools that should be available to authorised users to assist in the review of audit data.

Component levelling



FAU_SAR.1 Audit review provides the capability to read information from the audit records.

FAU_SAR.2 Restricted audit review requires that there are no other users except those that have been identified in FAU_SAR.1 that can read the information.

FAU_SAR.3 Selectable audit review requires audit review tools to select the audit data to be reviewed based on criteria.

Management: FAU_SAR.1

The following actions could be considered for the management functions in FMT:

- a) maintenance (deletion, modification, addition) of the group of users with read access right to the audit records.

Management: FAU_SAR.2, FAU_SAR.3

There are no management activities foreseen.

Audit: FAU_SAR.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- a) Basic: Reading of information from the audit records.

Audit: FAU_SAR.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- a) Basic: Unsuccessful attempts to read information from the audit records.

Audit: FAU_SAR.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- a) Detailed: the parameters used for the viewing.

FAU_SAR.1 Audit review

This component will provide authorised users the capability to obtain and interpret the information. In case of human users this information needs to be in a human understandable presentation. In case of external IT entities the information needs to be unambiguously represented in an electronic fashion.

Hierarchical to: No other components.

FAU_SAR.1.1 The TSF shall provide [assignment: *authorised users*] with the capability to read [assignment: *list of audit information*] from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

Dependencies: FAU_GEN.1 Audit data generation

FAU_SAR.2 Restricted audit review

Hierarchical to: No other components.

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

Dependencies: FAU_SAR.1 Audit review

FAU_SAR.3 Selectable audit review

Hierarchical to: No other components.

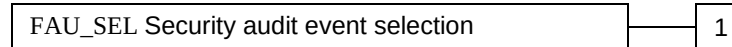
FAU_SAR.3.1 The TSF shall provide the ability to perform [selection: *searches, sorting, ordering*] of audit data based on [assignment: *criteria with logical relations*].

Dependencies: FAU_SAR.1 Audit review

3.5 Security audit event selection (FAU_SEL)

Family behaviour

This family defines requirements to select the events to be audited during TOE operation. It defines requirements to include or exclude events from the set of auditable events.



FAU_SEL.1 Selective audit, requires the ability to include or exclude events from the set of audited events based upon attributes to be specified by the PP/ST author.

Management: FAU_SEL.1

The following actions could be considered for the management functions in FMT:

- a) maintenance of the rights to view/modify the audit events.

Audit: FAU_SEL.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- a) Minimal: All modifications to the audit configuration that occur while the audit collection functions are operating.

FAU_SEL.1 Selective audit

Hierarchical to: No other components.

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

- a) [selection: *object identity, user identity, subject identity, host identity, event type*]
- b) [assignment: *list of additional attributes that audit selectivity is based upon*].

Dependencies: FAU_GEN.1 Audit data generation

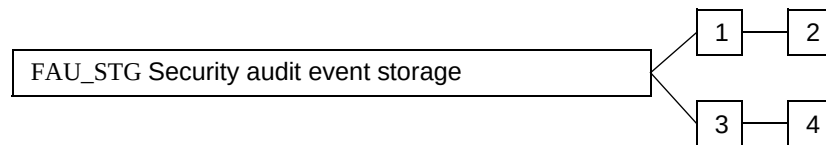
FMT_MTD.1 Management of TSF data

3.6 Security audit event storage (FAU_STG)

Family behaviour

This family defines the requirements for the TSF to be able to create and maintain a secure audit trail.

Component levelling



At FAU_STG.1 Protected audit trail storage, requirements are placed on the audit trail. It will be protected from unauthorised deletion and/or modification.

FAU_STG.2 Guarantees of audit data availability specifies the guarantees that the TSF maintains over the audit data given the occurrence of an undesired condition.

FAU_STG.3 Action in case of possible audit data loss specifies actions to be taken if a threshold on the audit trail is exceeded.

FAU_STG.4 Prevention of audit data loss specifies actions in case the audit trail is full.

Management: FAU_STG.1

There are no management activities foreseen.

Management: FAU_STG.2

The following actions could be considered for the management functions in FMT:

- a) maintenance of the parameters that control the audit storage capability.

Management: FAU_STG.3

The following actions could be considered for the management functions in FMT:

- a) maintenance of the threshold;
- b) maintenance (deletion, modification, addition) of actions to be taken in case of imminent audit storage failure.

Management: FAU_STG.4

The following actions could be considered for the management functions in FMT:

- a) maintenance (deletion, modification, addition) of actions to be taken in case of audit storage failure.

Audit: FAU_STG.1, FAU_STG.2

There are no actions identified that should be auditable if FAU_GEN Security audit data generation is included in the PP/ST.

Audit: FAU_STG.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- a) Basic: Actions taken due to exceeding of a threshold.

Audit: FAU_STG.4

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- a) Basic: Actions taken due to the audit storage failure.

FAU_STG.1 Protected audit trail storage

Hierarchical to: No other components.

FAU_STG.1.1 The TSF shall protect the stored audit records from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to [selection: *prevent*, *detect*] modifications to the audit records.

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.2 Guarantees of audit data availability

Hierarchical to: FAU_STG.1

FAU_STG.2.1 The TSF shall protect the stored audit records from unauthorised deletion.

FAU_STG.2.2 The TSF shall be able to [selection: *prevent*, *detect*] modifications to the audit records.

FAU_STG.2.3 The TSF shall ensure that [assignment: *metric for saving audit records*] audit records will be maintained when the following conditions occur: [selection: *audit storage exhaustion*, *failure*, *attack*].

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.3 Action in case of possible audit data loss

Hierarchical to: No other components.

FAU_STG.3.1 The TSF shall take [assignment: *actions to be taken in case of possible audit storage failure*] if the audit trail exceeds [assignment: *pre-defined limit*].

Dependencies: FAU_STG.1 Protected audit trail storage

FAU_STG.4 Prevention of audit data loss

Hierarchical to: FAU_STG.3

FAU_STG.4.1 The TSF shall [selection: *'ignore auditable events', 'prevent auditable events, except those taken by the authorised user with special rights', 'overwrite the oldest stored audit records'*] and [assignment: *other actions to be taken in case of audit storage failure*] if the audit trail is full.

Dependencies: FAU_STG.1 Protected audit trail storage

4 Class FCO: Communication

This class provides two families specifically concerned with assuring the identity of a party participating in a data exchange. These families are related to assuring the identity of the originator of transmitted information (proof of origin) and assuring the identity of the recipient of transmitted information (proof of receipt). These families ensure that an originator cannot deny having sent the message, nor can the recipient deny having received it.

Figure 4.1 shows the decomposition of this class into its constituent components.

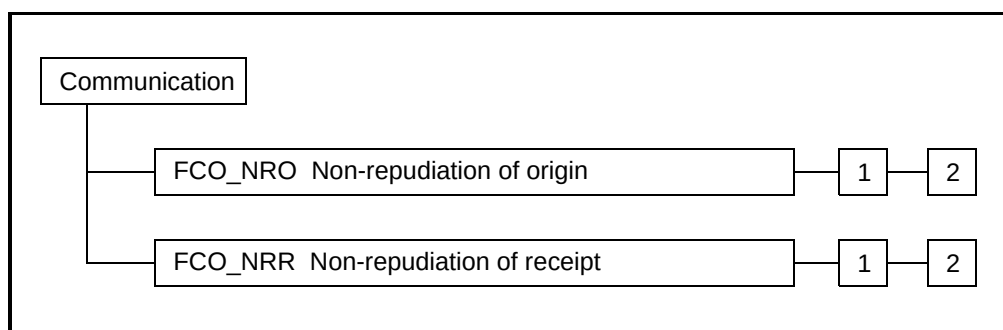


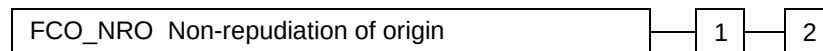
Figure 4.1 - Communication class decomposition

4.1 Non-repudiation of origin (FCO_NRO)

Family behaviour

Non-repudiation of origin ensures that the originator of information cannot successfully deny having sent the information. This family requires that the TSF provide a method to ensure that a subject that receives information during a data exchange is provided with evidence of the origin of the information. This evidence can then be verified by either this subject or other subjects.

Component levelling



FCO_NRO.1 Selective proof of origin requires the TSF to provide subjects with the capability to request evidence of the origin of information.

FCO_NRO.2 Enforced proof of origin requires that the TSF always generate evidence of origin for transmitted information.

Management: FCO_NRO.1, FCO_NRO.2

The following actions could be considered for the management functions in FMT:

- a) The management of changes to information types, fields, originator attributes and recipients of evidence.

Audit: FCO_NRO.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- a) Minimal: The identity of the user who requested that evidence of origin would be generated.
- b) Minimal: The invocation of the non-repudiation service.
- c) Basic: Identification of the information, the destination, and a copy of the evidence provided.
- d) Detailed: The identity of the user who requested a verification of the evidence.

Audit: FCO_NRO.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- a) Minimal: The invocation of the non-repudiation service.

- b) Basic: Identification of the information, the destination, and a copy of the evidence provided.
- c) Detailed: The identity of the user who requested a verification of the evidence.

FCO_NRO.1 Selective proof of origin

Hierarchical to: No other components.

FCO_NRO.1.1 The TSF shall be able to generate evidence of origin for transmitted [assignment: *list of information types*] at the request of the [selection: *originator, recipient, [assignment: list of third parties]*].

FCO_NRO.1.2 The TSF shall be able to relate the [assignment: *list of attributes*] of the originator of the information, and the [assignment: *list of information fields*] of the information to which the evidence applies.

FCO_NRO.1.3 The TSF shall provide a capability to verify the evidence of origin of information to [selection: *originator, recipient, [assignment: list of third parties]*] given [assignment: *limitations on the evidence of origin*].

Dependencies: FIA_UID.1 Timing of identification

FCO_NRO.2 Enforced proof of origin

Hierarchical to: FCO_NRO.1

FCO_NRO.2.1 The TSF shall **enforce the generation of** evidence of origin for transmitted [assignment: *list of information types*] **at all times**.

FCO_NRO.2.2 The TSF shall be able to relate the [assignment: *list of attributes*] of the originator of the information, and the [assignment: *list of information fields*] of the information to which the evidence applies.

FCO_NRO.2.3 The TSF shall provide a capability to verify the evidence of origin of information to [selection: *originator, recipient, [assignment: list of third parties]*] given [assignment: *limitations on the evidence of origin*].

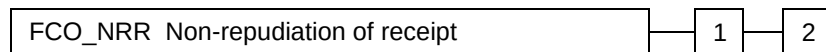
Dependencies: FIA_UID.1 Timing of identification

4.2 Non-repudiation of receipt (FCO_NRR)

Family behaviour

Non-repudiation of receipt ensures that the recipient of information cannot successfully deny receiving the information. This family requires that the TSF provide a method to ensure that a subject that transmits information during a data exchange is provided with evidence of receipt of the information. This evidence can then be verified by either this subject or other subjects.

Component levelling



FCO_NRR.1 Selective proof of receipt requires the TSF to provide subjects with a capability to request evidence of the receipt of information.

FCO_NRR.2 Enforced proof of receipt requires that the TSF always generate evidence of receipt for received information.

Management: FCO_NRR.1, FCO_NRR.2

The following actions could be considered for the management functions in FMT:

- a) The management of changes to information types, fields, originator attributes and third parties recipients of evidence.

Audit: FCO_NRR.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- a) Minimal: The identity of the user who requested that evidence of receipt would be generated.
- b) Minimal: The invocation of the non-repudiation service.
- c) Basic: Identification of the information, the destination, and a copy of the evidence provided.
- d) Detailed: The identity of the user who requested a verification of the evidence.

Audit: FCO_NRR.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- a) Minimal: The invocation of the non-repudiation service.
- b) Basic: Identification of the information, the destination, and a copy of the evidence provided.

- c) Detailed: The identity of the user who requested a verification of the evidence.

FCO_NRR.1 Selective proof of receipt

Hierarchical to: No other components.

FCO_NRR.1.1 The TSF shall be able to generate evidence of receipt for received [assignment: *list of information types*] at the request of the [selection: *originator, recipient, [assignment: *list of third parties*]*].

FCO_NRR.1.2 The TSF shall be able to relate the [assignment: *list of attributes*] of the recipient of the information, and the [assignment: *list of information fields*] of the information to which the evidence applies.

FCO_NRR.1.3 The TSF shall provide a capability to verify the evidence of receipt of information to [selection: *originator, recipient, [assignment: *list of third parties*]*] given [assignment: *limitations on the evidence of receipt*].

Dependencies: FIA_UID.1 Timing of identification

FCO_NRR.2 Enforced proof of receipt

Hierarchical to: FCO_NRR.1

FCO_NRR.2.1 The TSF shall **enforce the generation of** evidence of receipt for received [assignment: *list of information types*].

FCO_NRR.2.2 The TSF shall be able to relate the [assignment: *list of attributes*] of the recipient of the information, and the [assignment: *list of information fields*] of the information to which the evidence applies.

FCO_NRR.2.3 The TSF shall provide a capability to verify the evidence of receipt of information to [selection: *originator, recipient, [assignment: *list of third parties*]*] given [assignment: *limitations on the evidence of receipt*].

Dependencies: FIA_UID.1 Timing of identification

5 Class FCS: Cryptographic support

The TSF may employ cryptographic functionality to help satisfy several high-level security objectives. These include (but are not limited to): identification and authentication, non-repudiation, trusted path, trusted channel and data separation. This class is used when the TOE implements cryptographic functions, the implementation of which could be in hardware, firmware and/or software.

The FCS class is composed of two families: FCS_CKM Cryptographic key management and FCS_COP Cryptographic operation. The FCS_CKM family addresses the management aspects of cryptographic keys, while the FCS_COP family is concerned with the operational use of those cryptographic keys.

Figure 5.1 shows the decomposition of this class into its constituent components.

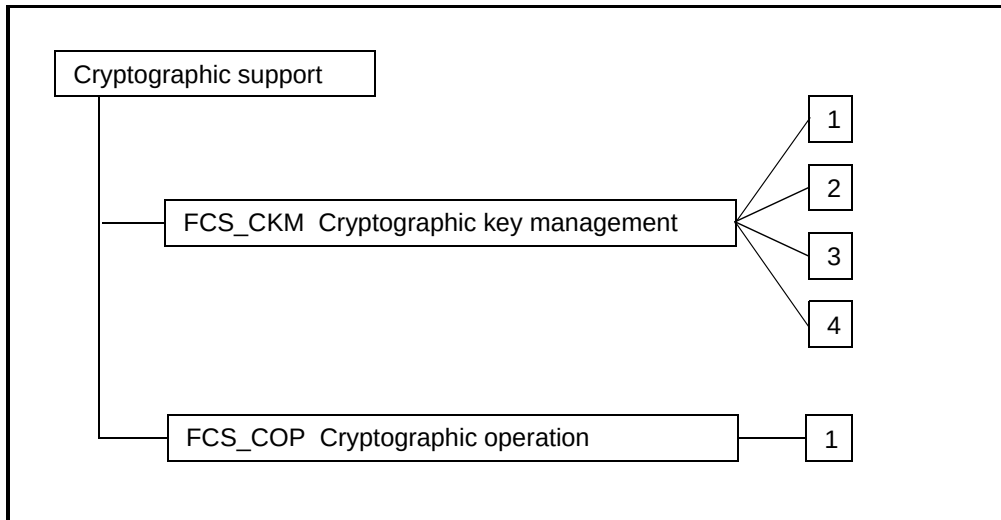


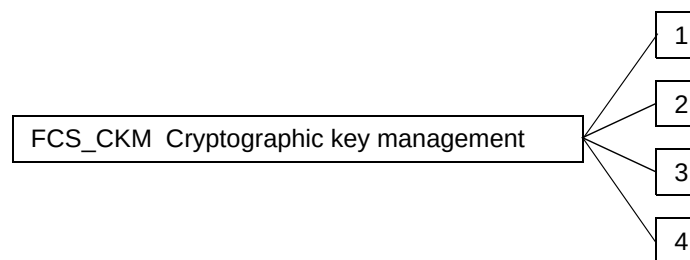
Figure 5.1 - Cryptographic support class decomposition

5.1 Cryptographic key management (FCS_CKM)

Family behaviour

Cryptographic keys must be managed throughout their life cycle. This family is intended to support that lifecycle and consequently defines requirements for the following activities: cryptographic key generation, cryptographic key distribution, cryptographic key access and cryptographic key destruction. This family should be included whenever there are functional requirements for the management of cryptographic keys.

Component levelling



FCS_CKM.1 Cryptographic key generation requires cryptographic keys to be generated in accordance with a specified algorithm and key sizes which can be based on an assigned standard.

FCS_CKM.2 Cryptographic key distribution requires cryptographic keys to be distributed in accordance with a specified distribution method which can be based on an assigned standard.

FCS_CKM.3 Cryptographic key access requires access to cryptographic keys to be performed in accordance with a specified access method which can be based on an assigned standard.

FCS_CKM.4 Cryptographic key destruction requires cryptographic keys to be destroyed in accordance with a specified destruction method which can be based on an assigned standard.

Management: FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.4

The following actions could be considered for the management functions in FMT:

- a) the management of changes to cryptographic key attributes. Examples of key attributes include user, key type (e.g. public, private, secret), validity period, and use (e.g. digital signature, key encryption, key agreement, data encryption).

Audit: FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.4

The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- a) Minimal: Success and failure of the activity.
- b) Basic: The object attribute(s), and object value(s) excluding any sensitive information (e.g. secret or private keys).

FCS_CKM.1 Cryptographic key generation

Hierarchical to: No other components.

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: *cryptographic key generation algorithm*] and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].

Dependencies: [FCS_CKM.2 Cryptographic key distribution
or
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security attributes

FCS_CKM.2 Cryptographic key distribution

Hierarchical to: No other components.

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [assignment: *cryptographic key distribution method*] that meets the following: [assignment: *list of standards*].

Dependencies: [FDP_ITC.1 Import of user data without security attributes
or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security attributes

FCS_CKM.3 Cryptographic key access

Hierarchical to: No other components.

FCS_CKM.3.1 The TSF shall perform [assignment: *type of cryptographic key access*] in accordance with a specified cryptographic key access method [assignment: *cryptographic key access method*] that meets the following: [assignment: *list of standards*].

Dependencies: [FDP_ITC.1 Import of user data without security attributes
or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security attributes

FCS_CKM.4 Cryptographic key destruction

Hierarchical to: No other components.

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [assignment: *cryptographic key destruction method*] that meets the following: [assignment: *list of standards*].

Dependencies: **[FDP_ITC.1 Import of user data without security attributes**

or

FCS_CKM.1 Cryptographic key generation]

FMT_MSA.2 Secure security attributes