Draft ETSI GS QKD 016 V0.6.2 (2021-12)

Quantum Key Distribution (QKD);
Common Criteria Protection Profile Pair of Prepare and Measure Quantum Key Distribution Modules

Disclaimer

This DRAFT is a working document of ETSI ISG QKD. It is provided for information only and is still under development within ETSI ISG QKD. DRAFTS may be updated, deleted, replaced, or obsoleted by other documents at any time.

ETSI and its Members accept no liability for any further use / implementation of the present DRAFT.

Do not use as reference material.

Do not cite this document other than as "work in progress".

Approved and PUBLISHED deliverables shall be obtained via the ETSI Standards search page at: http://www.etsi.org/standards-search

This draft is being developed by the Quantum Key Distribution (QKD) ETSI Industry Specification Group (ISG) but it does not yet represent the considered views of all members who participated in this ISG.

It does not necessarily represent the views of the entire ETSI membership.

0

1

Reference

DGS/QKD-016-PP

Keywords

quantum cryptography; Quantum Key Distribution

2

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group Quantum Key Distribution (QKD).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

Editorial note: Contributions are invited for any Executive summary (optional).

Introduction

Editorial note: Additional contributions are invited for the Introduction.

The current version of the Protection Profile in Annex A has not been certified. ISG QKD intends in the future to develop a certified revision to this Protection Profile.

1 Scope

Editorial note: The text below is based upon the original scope of the work item. Contributions are requested to improve the scope.

The present document specifies a Protection Profile for QKD systems, which describes complete systems involving point-to-point devices from the physical implementation up to the output of final secret keys. The PP specifies the high-level requirements, while technical details will be delegated to documents that either exist or need to be written.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] Common Criteria for Information Technology Security Evaluation: "Part 1: Introduction and General Model", Version 3.1, Revision 5, CCMB-2017-04-001, April 2017.

[2] Common Criteria for Information Technology Security Evaluation: "Part 2: Security Functional Components", Version 3.1, Revision 5, CCMB-2017-04-002, April 2017.

[3] Common Criteria for Information Technology Security Evaluation: "Part 3: Security assurance components", Version 3.1, Revision 5, CCMB-2017-04-003, April 2017.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GS QKD 004: "Quantum Key Distribution (QKD); Application Interface", V1.1.1, 2010-12.

[i.2] ETSI GS QKD 008: "Quantum Key Distribution (QKD); QKD Module Security Specification", V1.1.1, 2010-12.

[i.3] ETSI GS QKD 005: "Quantum Key Distribution (QKD); Security Proofs", V1.1.1, 2010-12.

[i.4] Joint Interpretation Library: "Minimum Site Security Requirements", Version 2.2, April 2019.

[i.5] Bundesamt für Sicherheit in der Informationstechnik AIS31 — Wolfgang Killmann, Werner Schindler: "A proposal for: Functionality classes for random number generators", Version 2.0, 18 September 2011.

[i.6] Bundesamt für Sicherheit in der Informationstechnik: "Evaluation of random number generators", Version 0.8.

[i.7] NIST Special Publication 800-90B: "Recommendation for the Entropy Sources Used for Random Bit Generation", January 2018.

[i.8] Jörn Müller-Quade and Renato Renner: "Composability in quantum cryptography", New J. of Phys. 11, 085006 (2009).

NOTE: Available at http://doi.org/10.1088/1367-2630/11/8/085006

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in CCMB-2017-04-001 [1] and the following apply:

Editorial note: Some definitions remain under review.

active probing: physical probing with additional active physical interaction with the probed device

NOTE: Active physical interactions may force the TOE to produce leakage that would otherwise not be emitted.

ADR Signing Key (ASK): private key to sign ADR for export

Audit Data Records (ADR): organized data generated for auditable events

Authentication Reference Data (ARD): data used by the TOE to verify the AVD sent by a user and in turn authenticate the user

Authentication Verification Data (AVD): data used by the user to authenticate themselves to the TOE

authenticity: property that ensures that the identity of an entity or the source of unmodified information is the one claimed (cf. ISO/IEC 7498-2:1989)

calibration: operation performed on calibration data by a user, including the comparison of measurement values delivered by the TOE with those of a calibration standard of known accuracy

calibration data: physical parameters of the underlying platform, that are adjustable and verifiable by a user, and that are required to be properly adjusted for the TOE to perform the QKD protocol securely

NOTE: Calibration data is considered TSF data. Calibration data may also refer to physical properties requiring physical tools for modification.

certification body: body issuing Common Criteria certificates that is accredited by a nationally recognized accrediting body

coherent attack: most general type of eavesdropping attack on the quantum channel, where an adversary interacts multiple ancillas coherently with QKD signals and then performs a joint measurement on all the ancillas and / or QKD signals to extract information

cryptographic key: a variable parameter that is used in and determines the functional output of a cryptographic algorithm or protocol

data integrity: property that data has not been altered or destroyed in an unauthorized manner (cf. ISO/IEC 7498-2:1989)

Maintainer: user authorized to perform calibrations

operational state: states of the operational life-cycle as defined in clause A.1.3

private key: confidential key used for asymmetric cryptographic mechanisms like decryption of cipher text, signature-creation for authentication proof, where it is infeasible for the adversary to derive the confidential private key from the known public key

public key: public known key used for asymmetric cryptographic mechanisms like encryption of cipher text, signature-verification for authentication verification, where it is infeasible for an adversary to derive the confidential private key from the known public key

prepare and measure protocol: protocol for a QKD system to establish QKD keys in which one QKD module prepares quantum states and the other measures quantum states

QKD Authentication Key (QAK): shared secret used for authentication mechanisms between both QKD modules

NOTE: The authentication is required to ensure the proper functionality of the prepare and measure protocol. The QKD authentication keys have to be available to the QKD modules before any communication using the QKD link can be established.

QKD key: pair of secret random bit strings established by a QKD system jointly in both QKD modules after successfully running a QKD protocol and considered to be identical

NOTE: QKD keys are exportable to authorized users for further use.

QKD link: set of active and/or passive components that connect a pair of QKD modules to enable them to perform QKD

QKD module: set of hardware, software, and/or firmware components that implements a part of a QKD protocol as well as cryptographic functions to be capable of securely establishing shared, confidential, random bit strings with at least one other QKD module

QKD protocol: algorithm that either aborts at any time or produces a shared, random, confidential bit string in the transmitter and receiver modules

QKD system: pair of QKD modules, interconnected by a quantum channel and a classical channel, i.e. a QKD link

QKD transaction: set of information defined by the ST author that is exchanged over the classical channel in a QKD link using QAK(s) that are not used by any other QKD transaction and that is limited by time, data exchanged and other limitations

quantum key distribution: procedure involving the transport of quantum states to agree shared secret bit strings between remote parties using a protocol with security based on quantum entanglement or the impossibility of perfectly cloning or measuring the unknown transported quantum states

remote entities: human users or IT devices consuming QKD keys, which eventually operate on behalf of human users, and communicate through a trusted path with the TOE

NOTE: The term is used solely in clause A.7.1 to point out a potentially indirect communication between human users and the TOE.

transaction: set of information defined by the ST author that is exchanged over a trusted path and limited by time, amount of data exchanged and additional limitations

trusted path: communication channel between QKD modules and remote entities that is logically distinct from other communication paths and that provides assured identification of its end points and protection of the communicated data from modification and disclosure

user: an entity using the TOE

NOTE: A user can either be a machine (on behalf of a human or other machines) or a human interacting with the TOE.

User Definition Records (UDR): information about known users and their associated roles

User Transaction Key (UTK): set of distinct cryptographic keys, where each key is used exclusively to protect data on the trusted path either against modification or disclosure

3.2 Symbols

For the purposes of the present document, the [following] symbols [given in ... and the following] apply:

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

A.xxx	Assumption
ADR	Audit Data Records
ARD	Authentication Reference Data
ASK	ADR Signing Key
AVD	Authentication Verification Data
CB	Certification Body
CC	Common Criteria
IT	Information Technology
ITS	Information Technology Security
n.a.	not applicable
O.xxx	Security Objective for the TOE
OE.xxx	Security Objective for the TOE Environment
OSP.xxx	Organisational Security Policy
P&M protocol	Prepare and Measure QKD protocol
PP	Protection Profile
QAK	QKD Authentication Key
QKD	Quantum Key Distribution
SAR	Security Assurance Requirements
SFP	Security Functional Policy
SFR	Security Functional Requirement
ST	Security Target
T.xxx	Threat
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSP	TOE Security Policy
UDR	User Definition Records
UTK	User Transaction Key

4 User defined clause(s) from here onwards

4.1 User defined subdivisions of clause(s) from here onwards

Editorial note: Contributions are invited for any additional clauses for the main body.

4.2 Application Notes in the Protection Profile

Specific requirements apply to the use of Application Notes in different locations within a Protection Profile and its packages but it is important to note that in general they can have normative impact on the evaluation of a product.

Notes marked "NOTE: (Informative)" within the Protection Profile (Annex A) do not have immediate impact on the evaluation. Such notes would sometimes be referred to as Editorial Notes in a Protection Profile and are intended to be retained in published Protection Profiles.

Editorial note: In this draft, Editorial notes (such as this one) that are written in colour between horizontal lines are temporary notes for use during the preparation of this deliverable and will be removed before publication.

Annex A (normative): Common Criteria Protection Profile Pair of Prepare and Measure Quantum Key Distribution Modules

Editorial note: Copyright clause on reproducing content in ST documents to be inserted in the deliverable.

A.1 PP introduction

A.1.1 PP reference

Editorial note: The entries in this clause need to be reviewed as the document is finalised.

Title:	Common Criteria Protection Profile Pair of Prepare and Measure Quantum Key Distribution Modules
Sponsor:	Federal Office for Information Security (BSI)
CC Version:	3.1 Revision 5
Editor:	Deutsche Telekom Security GmbH, Evaluation Facility
Assurance Level:	EAL4 augmented with AVA_VAN.5 and ALC_DVS.2
General Status:	Draft
Version Number:	V0.6.2
Registration:	
Keywords:	Cryptographic Module, Cryptography, Quantum key distribution

A.1.2 PP Overview

This Protection Profile describes the security requirements for Quantum Key Distribution modules (QKD modules), which use a Prepare and Measure QKD protocol (P&M protocol). This Protection Profile considers the case, where both modules are located in environments with identical security requirements.

This Protection Profile deliberately offers degrees of freedom to ST authors in order to allow them to adapt to upcoming QKD standards and to foster innovative solutions in an upcoming technology. The developers and ST authors are advised to contact their certification body (CB) before and during development to establish a common interpretation. In particular, the CB may discourage certain cryptographic algorithms or protocols for this field of use, which would formally be valid choices in this PP. The PP is written with several incompatible use cases, environments, and business models in mind, and offers options, choices, and even blanks to fill for the ST author to accommodate most of these. Some combinations may appear formally correct, but would be unacceptable to the CB. Developers are advised to agree on the ST with the CB before finalizing the architecture of the product.

A.1.3 TOE overview

TOE type

The Target of Evaluation **(TOE)** is a QKD system (see [i.1]) as laid out by the ETSI Industry Specification Group **(ISG)** on Quantum Key Distribution for users **(QKD)**. The TOE Security Functionality **(TSF)** provides a consistent subset of the functionality defined for these systems in [i.2].

TOE definition

The TOE comprises a QKD system consisting of two QKD modules, but without the QKD link in between. It furthermore includes the associated guidance documentation. The QKD link may pass through uncontrolled environment without physical protection, and does not provide any security services. The QKD link is used for at least two communication channels, a classical and a quantum channel. The communication using the QKD link is considered Inter-TSF communication.

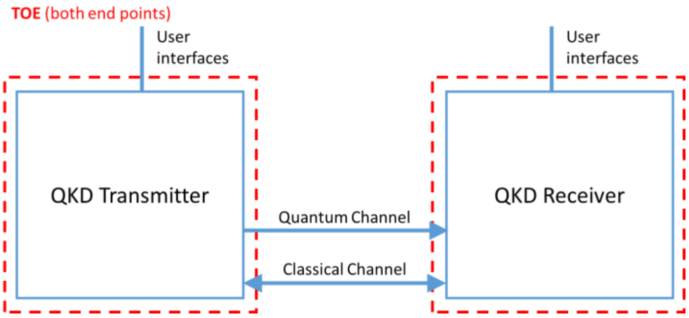


Figure 1: The TOE-boundary, i.e., the two QKD modules

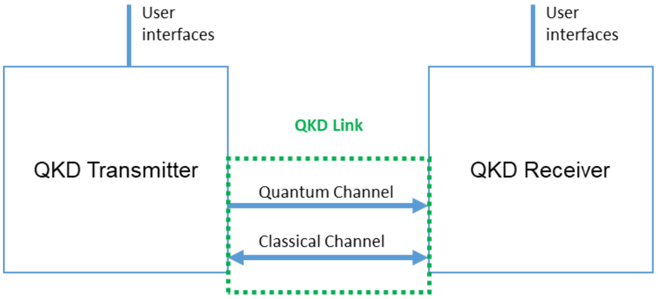


Figure 2: The QKD link

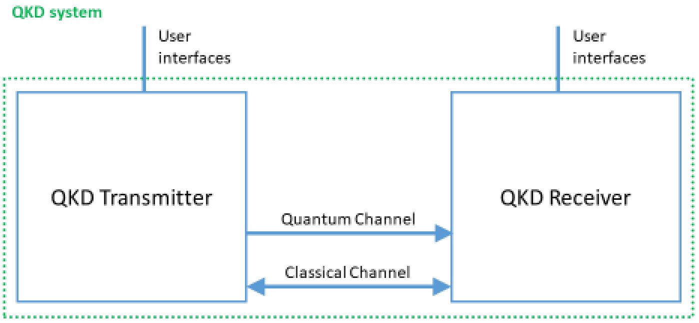


Figure 3: The QKD system

The purpose of the QKD system is to establish QKD keys in a paired QKD system of one QKD transmitter and one QKD receiver. QKD keys are shared, confidential, random bit strings in both QKD modules, which can be consumed by authorized users in well-defined chunks. The property "random" is used in the sense that the strings are unpredictable, uniformly distributed, and independent from each other, i.e., the QKD system shall implement a source with forward and backward secrecy. Each of these properties may be subject to imperfections. The TOE guarantees an upper limit for such imperfections. The ST introduction shall detail this upper limit[1].

If these bit strings are successfully established for export, they are called QKD keys regardless of their appropriateness for or actual use as cryptographic keys. The TOE exports these QKD keys to authorized users from each QKD module[2].

[1] cf. application note *3*

[2] The TOE may use generated shared bit strings for internal purposes. Bit strings used internally shall not be exported as QKD keys.

QKD systems may be modelled in a notion of information-theoretical security and this PP requires a security proof for the QKD protocol. The actual establishment of these QKD keys shall be resistant to attackers with high attack potential[3].

In order to establish QKD keys, the QKD system uses a Prepare and Measure protocol (P&M protocol) as defined in [i.3]. Although these protocols may vary greatly, there is always a distinct sequence of phases:

1) The initialization phase is used to prepare both QKD modules for the establishment of a QKD key. It is not part of the core P&M protocol, but required to initiate the protocol. It may include self-tests, synchronizing the QKD modules, preparation of storage, etc. This phase is initiated upon a user's request for QKD key establishment.

2) During the quantum communication phase the QKD modules prepare and measure quantum states depending on the chosen P&M protocol and their respective role in it.

3) The post-processing phase is used to create the confidential, shared, random bit string from the results of the quantum communication phase. This phase may comprise steps as described in [i.3] like data partitioning, sifting, parameter estimation, error correction (reconciliation), confirmation, privacy amplification, or authentication. The bit string may be partitioned into a QKD key for export and TSF data for internal use. Authentication key derivation and an update of authentication keys for both QKD modules may be part of this phase. Depending on the implementation some steps may not apply while other steps may be added. It comprises whatever is required to establish the confidential QKD key in both QKD modules or to determine that the requested quality of the QKD key cannot be established.

4) During the output phase the QKD key is transferred to the authorized user at each QKD module, or the users are notified that no QKD key could be established.

The TOE may support interleaving transactions for establishing different QKD keys, e.g. it could support performing the quantum communication phase for one key while still performing the post-processing phase for the previously requested key. If running multiple transactions in parallel, the ST author shall extend the ST to support multiple transaction keys. Architectures where QKD keys are not established on explicit user request, but e.g. taken from a pool of continuously generated data, may be based on this PP. In this case, the ST author shall clearly define in the ST introduction what constitutes a QKD transaction, i.e., the scope of a single transaction key, and how it is limited. The data pool by itself would be considered TSF data from which QKD keys are taken eventually.

The TOE manages users with permission to produce and extract QKD keys and provides functions to manage those users, adjust and administrate TSF, and audit specific events.

The security services provided by the TOE are summarized as follows:

1) support of a calibration and pairing mode for the QKD system for designated Maintainers,

2) establishment of the QKD key**,** specified by the authorized user of the TOE using a P&M protocol via the QKD link,

3) plain-text export of the QKD key on behalf of designated users at either QKD module,

4) enforcement of a role-based access control defined by a designated Administrator,

5) generation and export of audit data[4] as defined by a designated Auditor,

6) protection[5] of the configuration and initialization data related to the behaviour of the security functionality.

The key distribution service provided by the TOE is defined as the establishment of the QKD key using a P&M protocol via the QKD link.

[3] Resistance against attackers with high attack potential is required by the SAR AVA_VAN.5.

[4] The required auditable events generating audit data are listed in the SFR FAU_GEN.1, sec. A.6.1.3.

[5] The type of protection (i.e. confidentiality, integrity, authenticity, availability) provided by the TSF depends on the respective data and their protection requirements for a secure operation of the TOE.

While the security services include the export of QKD keys, neither the management of QKD keys necessary for their usage nor the protection of the QKD key after their export to authorized users is provided by the TOE as modelled in this PP.

There are various viable approaches, which ensure the required security provided for user identification via the user interfaces and authentication of the classical channel of the QKD link. Viable approaches for both communication channels may cover algorithms providing either information-theoretical or computational security. User identification may not involve any technical security at all. Symmetric, asymmetric and hybrid algorithms may be considered suitable for establishing a trusted path, for the subsequent security functionalities provided by it and for the authenticity of exchanged data through the classical channel. The cryptographic keys required for their security services may or may not be derived from previously established QKD keys.

To assure that the chosen cryptographic implementations meet the security requirements of the intended application(s), users are advised to consult with the certification body before finalizing the architecture of the product.

The TOE is intended for operation in an access-controlled environment and features only local user access. User identification may be as simple as connecting to the appropriate interface, while the access control policy of the environment ensures user authorization.

However, the PP does define packages for more common use cases. Users may connect to the TOE via a trusted path, which requires some external IT device. In this scenario users may be located remotely. In this case, the ST author is advised to select the package defined in clause A.7.1, disregarding whether the users are actually remote. In case the TOE itself features the interface for human users, the package in clause A.7.4 may apply.

Another package deals with self-protection of the security services of the TOE, if it shall be deployed in an environment, which cannot impede attackers with high attack potential (e.g., organized crime or foreign intelligence services). The ST author is advised to pick the package defined in clause A.7.2, if the TOE shall be operated in a commercial grade environment.

Finally, clause A.7.3 defines a package to personalize and re-personalize the TOE after delivery.

TOE users

The TOE supports local user interfaces, which may be integrated into the TOE or require some IT product to be connected as a user interface. The ST author shall detail the required non-TOE hard- and software if required. The basic configuration for an access-controlled environment does not authenticate users, because only authorized users will have access to the TOE. The ST author is advised to select one of the packages defined in clauses A.7.1 or A.7.4, if user authentication is desired. Otherwise, the ST author shall detail how users are authenticated.

The TOE associates roles to identified users. At least the following roles are supported by the TOE:

- Administrator
- Maintainer
- Auditor
- Key Requester

An identified user in the role Administrator is allowed to associate user identities with roles. Likewise, the Maintainer is allowed to query, modify and change the default values for calibration data, the Auditor is allowed to define auditable events. The Auditor may also export audit records and delete them from the TOE after export. The Key Requester is allowed to request establishment and export of QKD keys.

ST authors may subdivide roles in order to match their application requirements[6]. The access permissions of roles shall not be merged.

[6] The ST may define additional roles or split current roles into sub-roles, e.g. the Administrator role may be split in a User Administrator role and a Crypto Officer role or the Maintainer role may be split in a Hardware Maintainer and a Calibrator role.

Method of use

On request, the TOE delivers a shared QKD key with a well-defined quality or notifies the users at both QKD modules of a failure. The original Key Requester will define which users are allowed to receive the QKD key from each QKD module. It is the users' responsibility to properly handle the established QKD key, especially to ensure the security requirements as required for further use. This PP is limited to QKD key establishment. Any further use of the QKD key and its suitability for any specific purpose is beyond the scope of this PP.

The TOE may produce the QKD key in background and deliver portions of requested length to the user, or produce a dedicated QKD key in response to a request. A continuous QKD key bit stream may be considered as a background establishment with 1-bit deliveries. This PP does not limit the user interfaces in this respect, but it requires to protect any pre-generated bits of the QKD key, while they are stored in the TOE, and requires deletion of bits after consumption.

Life-cycle

This PP defines a generic life-cycle for the TOE. It is acknowledged that production processes are not yet standardized along the industry. It is neither the intent of this PP to define such standards nor to interfere with the competition of manufacturers concerning the most usable concepts. The ST author shall detail and where appropriate subdivide the phases given here.

The generic life-cycle model consists at least of the following high-level phases:

- Development phase,
- Manufacturing phase,
- Pre-operational phase,
- Operational phase, and
- End of Life,

which may be detailed to accommodate the actual processes for provisioning and deployment. Figure 4 puts some conceptual detail to this scheme. In particular, delivery may be chosen to occur in between steps, which are considered the pre-operational phase in this PP.

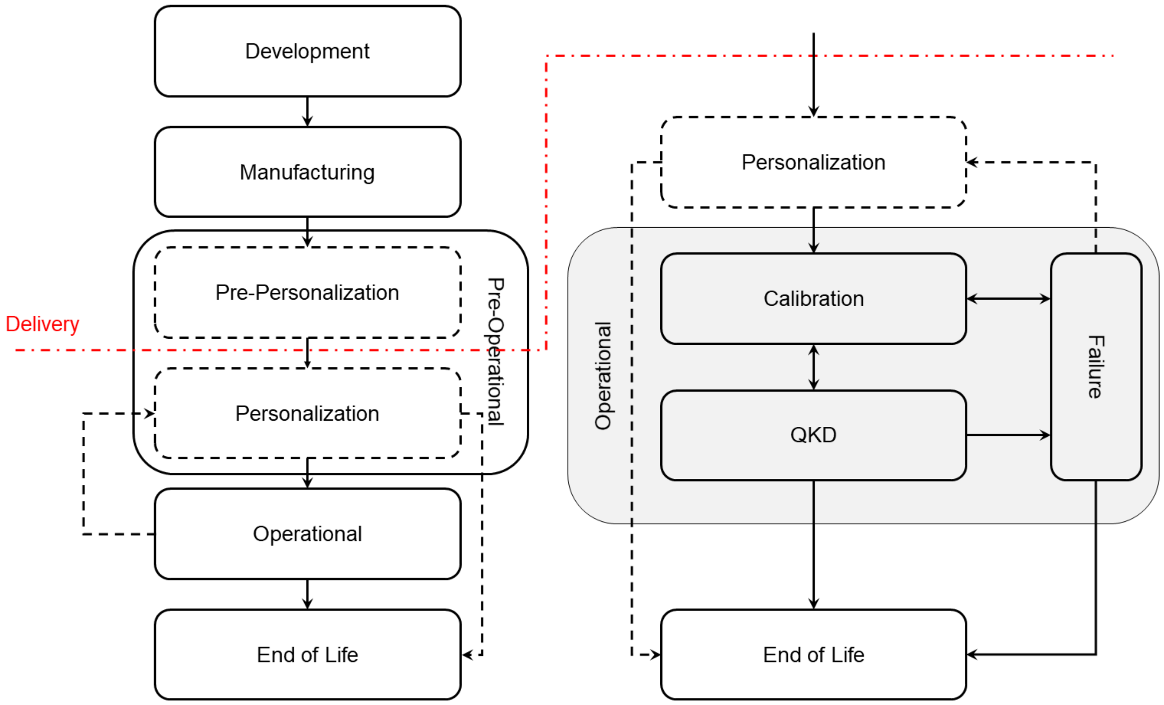


Figure 4: Life-Cycle model overview
Left: Complete life-cycle. Right: Close-up of post-delivery phases including operational states of the TOE. Dashed elements may be empty and are not defined in this PP.

During the development and manufacturing phase, the TOE, its components, and associated documentation about the development and production is under control of the manufacturer or his sub-contractors. Sensitive information shall be restricted by a documented need to know policy.

During the development phase, i.e., before the TOE for delivery is actually built, the full production documentation is generated. Furthermore, it is expected that analyses with respect to feasibility or optimal parametrisation of mechanisms will be performed. These documents shall be protected from illicit modification both in scope and content. While corrupted production documents may lead to compromised TOE instances, the analyses may provide valuable input for test strategies and vulnerability analyses.

The manufacturing phase, i.e., when the TOE for delivery is actually built, shall strictly adhere to the production documentation generated during the development phase. It shall be ensured that each instance is built exactly as developed in order to guarantee the security services offered by the TOE. Furthermore, the production shall track each instance until delivery.

The pre-operational phase comprises everything required to customize and configure the TOE to achieve that all TSF are enforced. This necessarily includes provisioning of initial secrets / credentials required for pairing the QKD modules to form a QKD system, i.e. the QKD authentication key (QAK). The PP anticipates that there will be many different approaches for this phase. Note, that prior consultation with the certification body is advised, since not all instantiations may be acceptable. The base PP assumes that the TOE is delivered as a pair of QKD modules already paired as a QKD system, i.e., the pre-operational phase takes place before delivery. In clauses A.7.3 a package with additional security functionality is presented, if the pre-operational phase shall be left with the user after delivery.

Actual commercial and scalable processes may involve third parties, e.g. retailers, solution integrators, or network operators, to perform (parts of) the (pre-)personalization during pre-operational phase. ST authors shall sub-divide this phase appropriately and define the actual delivery to the user[7]. The sub-divisions shall clearly describe

1) who is responsible and accountable for the security of the TOE during that phase[8],

2) whether the phase is before or after delivery[9], and

3) 3) which secrets / credentials are processed and imported to or generated by the TOE. If secrets are generated by the TOE, this will require appropriate TSF to be defined in the ST. If secrets are generated externally, this will require appropriate sources. If secrets / credentials are processed, adequate site security is required to protect against high attack potential.

The ST author shall furthermore define appropriate TSF for pre-operational tasks performed after delivery.

During the operational phase the TOE is under control of the user and set-up to establish QKD keys. This phase is after delivery, i.e., the TSF are enforced and the assumptions of this PP apply. This PP defines several recoverable error conditions, where the TOE stops establishing QKD keys.

This Protection Profile assumes the following operational life-cycle states, which may be more detailed by the ST author to match the particular implementation:

- Calibration state
- QKD state
- Failure state
- End of Life

The PP assumes that the TOE is delivered as a ready to use QKD system, i.e. there is no Personalization state. Clause A.7.3 defines a package which puts the pre-operational phase after delivery, i.e. into the Personalization state.

[7] Note that each site / party involved before delivery will be subject to evaluation according to class ALC, and that any pre-personalization after delivery has to be under control of the TSF.

[8] There shall be no phase, where the accountability is not uniquely defined.

[9] There shall not be a phase, which contains delivery, and following delivery there shall be no more pre-delivery phase.

Calibration state:

The TOE requires a diligent calibration of physical parameters in order to properly enforce the key distribution services of the P&M protocol. This calibration requires trusted and skilled personnel, who access the TOE in the role of a Maintainer. The TOE does not offer any other service while in Calibration state.

The Calibration state is required for initial set-up of the QKD system and thus necessarily precedes the QKD state. However, scheduled maintenance and repair operations may require the TOE to return to the Calibration state[10]. The Maintainer role has the permission to perform this life-cycle shift and may perform the maintenance and repair operations that are possible in the field. Such shifts to and from the Calibration state and operations performed therein shall generate audit data.

Leaving the Calibration state shifts to QKD state, unless the TOE self-test requires a shift to Failure state.

QKD state:

In QKD state, the TOE is used to establish the QKD key at both QKD modules. This process is initiated by a user in Key Requester role. The TOE exports the established QKD key to Key Requesters designated as receivers by the requesting user and deletes it from internal storage at both modules.

It furthermore allows user data management by the Administrators and audit data management by the Auditors. The TOE may monitor and tune its TSF to maintain secure operation, e.g. adapting calibrations to environmental influences.

Failure state:

The TOE is able to detect a certain set of malfunctions of itself. In this case it may shift to Failure state or, depending on the type of failure, immediately to End of Life. If it shifts to Failure state, either an Administrator can shift it to End of Life manually, or if applicable shift it to the Personalization state for re-personalization, or a Maintainer may shift to the Calibration state for repair.

The TOE may also shift to End of Life from Failure state if additional conditions potentially compromising its security are detected.

End of Life

In End of Life state the TOE erases all confidential user data and TSF data or ensures that confidential data cannot be retrieved, for data that cannot be erased[11]. The TOE prohibits any further operation or state transition.

The Guidance documentation shall specify a procedure to securely destroy the QKD modules.

Non-TOE hardware/software/firmware available to the TOE

The TOE needs a classical and a quantum channel connecting the two QKD modules. The links need to be able to exchange the TSF data as required by the TOE.

If the TOE does not feature inbuilt user interfaces, it requires some terminal device as user interface. The ST author shall detail the specific requirements for the TOE.

A.2 Conformance claims

A.2.1 CC conformance claims

The PP claims conformance to CC version 3.1 revision 5 [1].

Conformance of this PP with respect to CC Part 2 [2] (security functional components) is CC Part 2 extended.

Conformance of this PP with respect to CC Part 3 [3] (security assurance components) is CC Part 3 conformant.

[10] Although this Protection Profile models only calibration procedures performed by a Maintainer, the actual implementation may require or enable additional automated calibrations, both for initial and maintenance calibrations during the Calibration state, and for regular calibrations during the QKD state. The ST author shall model those calibration and self-test procedures and their requirements.

[11] To guarantee that data cannot be retrieved, the TOE may ensure that the memory for confidential data cannot be read.

A.2.2 Package claim

This PP claims package-augmented conformance to EAL4. The minimum assurance level for this Protection Profile is EAL4 augmented with AVA_VAN.5 and ALC_DVS.2.

A.2.3 PP claim

This PP does not claim conformance to any PP.

A.2.4 Conformance rationale

This chapter is not applicable because the PP does not claim conformance to any PP.

A.2.5 Conformance statement

Security targets and protection profiles claiming conformance to this PP at hand shall conform with strict conformance to this PP.

A.2.6 PP Application Notes

Operations that are not completed in this Protection Profile shall be completed by the ST author.

In chapter A.7 the Protection Profile defines several packages to support extended functionality of the TOE. ST authors may choose any of these considering that A.7.1 and A.7.4 are mutually exclusive. If these packages do not reflect the actual extended security functionality, ST authors may extend the Protection Profile by their own modelling. In this case, the packages in chapter A.7 may serve as examples for orientation.

The ST/PP author shall adopt all formal items from a package, if conformance to this PP with that package is claimed. This Protection Profile contains other application notes distributed through the paper. The application notes are separated paragraphs which are marked with "Application Note" followed by a number.

This Protection Profile does not mandate storage encryption and storage integrity protection as dedicated SFR. This security functionality is often required for devices used in security applications. ST authors may add respective SFR to meet such requirements.

A.3 Security problem definition

A.3.1 Introduction

Assets and TSF data

The assets of the TOE are those security services and data, for whose protection the TOE primarily exists. These assets are

- *QKD keys*, whose integrity and confidentiality shall be protected,
- *key distribution services* which shall be protected against unauthorized use.

Beyond the assets the TOE maintains additional information, which by itself is not threatened. However, compromising such a secondary asset may be an important step on attacks to the assets above. The secondary assets are:

ADR Audit Data Records

The TOE furthermore maintains TSF data. Compromising this data may compromise the security services of the TOE. These data elements are:

QAK *QKD Authentication Key*, the shared secret required to authenticate the classical communication on the QKD link,

ASK ADR Signing Key, i.e., the key to sign ADR for export,

UDR User Definition Records, the information about known users and their associated roles, CD calibration data, physical parameters of the underlying platform, which are adjustable and verifiable by a user, through any interface or by physical manipulation, and which are required to be properly adjusted for the TOE to perform the QKD protocol securely.

Users and subjects

The TOE communicates with

- users by local user interfaces in an environment secured by organizational means, and
- itself (i.e., the remote peer QKD module), via the QKD link.

The TOE may offer user interfaces, which can be operated by human users immediately, or offer technical interfaces, where such interfaces (terminals) can be connected to, locally. As described in clause A.1.3 the TOE associates identified users with at least the following roles according to the UDR:

Unidentified users are users, which are not associated with any UDR,

Administrator able to define new users and assign roles to users by creating, modifying, and deleting UDR,

Auditor able to export audit data records (ADR) and clear exported audit data from the TOE,

Maintainer able to configure, calibrate, or perform limited repairs of the TSF, i.e., modify the CD, and

Key Requester as authorized user of the key distribution services and recipient of QKD keys.

The TOE protects the assets against operations by adversaries. The adversary is not considered limited in the choice of his means beyond the assumptions stated in this Protection Profile. Hence coherent attackers are implied as long as their attack potential does not surpass high attack potential.

The subjects as active entities in the TOE perform operations on objects. The subjects obtain their associated security attributes either by default or from the authenticated users on whose behalf they act.

Objects

The TOE maintains the following user data objects and manages user access to these objects:

QKD keys are created using the key distribution services on behalf of Key Requesters. They are temporarily stored and exported to Key Requesters, if successfully established. They are destroyed after export, after a defined time or on behalf of authorized users.

ADR, Audit Data Records, are generated for auditable events according to FAU_GEN.1. ADR may be exported by Auditors for external archiving and deleted after export. Audit shall be used for forensic purposes and therefore modifications shall be detectable.

Security attributes

The security attributes of users known to the TOE are stored in User Definition Records (UDR) containing

- *User Identity* (User-ID),
- *Role* determining the access rights.

The TOE supports at least the roles defined above under Users and subjects. The TOE is delivered with initial UDR for Unidentified User and at least one Administrator.

Key Requesters may specify who is allowed to finally receive any requested QKD key from each QKD module. The QKD keys therefore hold the *receiver* and *owner* attributes.

Audit Data Records carry the security attribute *exported*, which is false on creation and true after successful export by an Auditor.

The Security Target (ST) author may define additional security attributes or may subdivide roles to map specific operational policies.

While not a security attribute by itself, the TSF data item *operational state* determines the current rules for access of all subjects to any objects based on the aforementioned security attributes.

A.3.2 Threats

T.ServAcc **Unauthorized access to user data**

An identified user gets unauthorized access to

a) key distribution services of the TOE, or

b) the QKD key.

The identified user may also exploit inconsistent or ambiguous rules concerning the authorized receiver of the QKD key at either QKD module.

T.Session **Session hijacking or piggybacking**

An adversary or a legitimate user may use the open session of a different identified user to get unauthorized access to

a) key distribution services of the TOE, or

b) the QKD key.

T.QKDEave **Eavesdropping on QKD link data**

An adversary may eavesdrop on the communication sent through the QKD link in order to compromise the confidentiality of the QKD key.

T.QKDMani **Manipulation of QKD link data**

An adversary generates or manipulates data on the QKD link in order to compromise the confidentiality of the QKD key. Attacks which aim to regenerate some part of previously established QKD keys are considered as attacks, which compromise the confidentiality of the QKD key.

Application Note 1 Attacks, which may induce a bias, prefer bit patterns or similarly affect the statistics of the QKD key, including correlations to any previously generated QKD keys or correlations to results of other QKD links, shall be considered as compromising the confidentiality.

T.ExplMal **Exploitation of TOE malfunction**

An adversary or unauthorized user gains knowledge of a QKD key by exploiting malfunction of the TOE either induced, spontaneous or due to incorrect calibration.

T.Observe **Observation of TSF characteristics**

An adversary observes emanations, including signals on intended interfaces, or injects probe signals through accessible interfaces of the TOE, or applies other non-destructive inspection methods (e.g. X-ray or radar imaging) in order to obtain intelligence concerning the internal state of the TSF suitable to compromise the confidentiality of the QKD key.

Application Note 2 Attacks, which may expose a bias, preferred bit patterns or similar effects on the statistics of the QKD key, including correlations to any previously generated QKD keys or correlations to results of other QKD links, shall be considered as compromising the confidentiality.

A.3.3 Organisational security policies

OSP.QKDService **Key distribution services of the TOE**

The TOE provides key distribution services to authorized users. The key distribution services are based on a P&M protocol for quantum key distribution and establish shared, confidential, random bit strings in each QKD module.

OSP.Audit **Audit for security operations**

The TOE supports security auditing of administration, calibration, and key distribution service operations. The configuration of the scope of the data audited and the permission to delete audit data is restricted to the Auditor role. *Users with an* Auditor *role shall neither hold an* Administrator *nor* Maintainer *role.*

Exported audit data is stored securely for forensic purposes.

OSP.SecEoL **Secure End of Life state**

The TOE deletes all confidential data or ensures that confidential data cannot be retrieved, for data that cannot be erased, when it reaches the End of Life state. It shall at least allow the Administrator role to deliberately put the TOE to end of life for decommissioning.

A.3.4 Assumptions

A.Maint **Diligent maintenance**

The Administrator and Maintainer are trustworthy users. Maintainers perform calibrations diligently without deliberately compromising the security of the TOE. Administrators will not add users or assign roles to users who are not authorized. Administrators will assign users as Auditors. Auditors will configure and perform audits of the TOE.

A.SecureOp **Operation in a secure area**

The TOE is installed and operated in a secure area, i.e., only authorized personnel can obtain physical access to the TOE. These authorized personnel will not intentionally misuse the TOE. The environment will detect any unauthorized access and the TOE will be taken out of service upon such detection.

A.4 Security objectives

A.4.1 Security objectives for the TOE

O.Identify **Identification of users**

The TSF shall uniquely identify users before providing access to any controlled resources. Each user shall be associated with at least one role.

O.AccCtrl **Access control**

The TSF provides access to control to

1) key distribution services and QKD keys,

2) ADR, and to

3) management of TSF and TSF data,

based on roles of identified users and the operational state of the TOE (cf. Life-cycle).

The TSF ensures that each role is constrained to its associated permissions and that Administrator and Auditor role cannot be shared by the same identified user.

The TSF shall maintain unambiguous and consistent information about which users at each QKD module are allowed to receive any given established QKD key and deny access to any other users.

O.QKD **Quantum Key Distribution**

The TSF provides key distribution services based on a P&M protocol for quantum key distribution and deletes the QKD key immediately after (acknowledged) export or time-out from the respective QKD module. The key distribution services establish shared, confidential, random bit strings for export as QKD keys even in the presence of an eavesdropper on the communication on the QKD link, given that the communication on the classical channel of the QKD link is authenticated.

Application Note 3: The key distribution services in the sense of the objective O.QKD comprises all processing steps starting from the data exchange on the QKD link up to the final agreement on the shared QKD key. This may include any number of repetitive attempts to establish a QKD key if single protocol runs led to abortion.

O.QKDAuth **Authentication of classical channel**

The TSF provides mutual authentication of both QKD modules, i.e., ensures the authenticity of the data exchanged for O.QKD through the classical channel. Authentication is based on a shared secret, the QKD Authentication Key (QAK).

To avoid compromise of the QAK to an adversary the TSF updates the QAK regularly. Data exchanged using the same QAK or keys derived from it is considered a single QKD transaction. Updating the QAK may consume a part of the shared secret bit string, which in turn cannot enter the QKD key anymore. The update protocol ensures that the confidentiality of the QAK is not compromised by eavesdropping on any part of the communication.

If no new QAK is available at the end of a QKD transaction, the TSF denies any further access to the key distribution services and sets the operational state to Failure state.

Application Note 4: The ST author shall define the limits of the *QKD transaction* to avoid any form of overuse of *QAK* or use of the same *QAK* for distinguishable purposes.

Replacement of parts of the *QAK* e.g., as used for certain Wegman-Carter implementations, shall not be considered key derivation but a new *QAK* for the purpose of transaction definition. The necessity to prevent overuse of information contained in the *QAK* remains.

NOTE: (Informative) The base PP assumes that the TOE is delivered with an initial *QAK* already defined by the manufacturer. See the package in clause A.7.3, if *QAK* shall be defined / replaced after delivery. Note that without this option a used up *QAK* or run out of synchronization *QAK* necessarily leads to *End of Life* phase.

O.Audit Audit for cryptographic TSF

The TSF provides security auditing of administration, calibration, and key distribution services by recognizing, recording, and reliably storing of selected auditable events using audit records related to activities controlled by the TSF. The TSF provides the Auditor exclusively with management functionality to define additional auditable events and to delete audit records after export. The TSF generates evidence for the validity and origin of said audit records and enables the Auditor to verify the said validity.

O.TST Self-test

The TSF self-tests important security functions and monitors its operational parameters, including the parameters of the QKD link. It denies access to the key distribution services and QKD keys unless the TSF are ensured.

The TSF supresses or detects signals on the QKD link, which are suitable to probe internal states of the TSF. It denies access to the key distribution services and QKD keys, if such probing signals are detected.

O.EMSec Emanation Security

The TSF is designed in order to prevent leakage of any intelligible confidential user data or TSF data through the QKD link. This includes leakage induced by any active probing.

Application Note 5: *Information sent intentionally through the* QKD link *is considered to be non-confidential. The TSF shall suppress side-channel information accompanying this intentional traffic, e.g. timing, signal levels, noise, ...*

O.Sanitize Secure End of Life state

The TSF allows to securely delete all confidential information stored in the TOE before entering an End of Life state. The TOE in End of Life state cannot be returned to operational use. Full disclosure of a TOE in end of life does neither compromise any QKD key generated by the TOE, nor does it allow use of key distribution services, nor does it contain information suitable to compromise other instances of the TOE.

While ST authors may require access restrictions as to which role may induce a shift to the End of Life state, the PP requires no particular restriction beyond that the Administrator role shall be allowed to perform this transition. ST authors shall consider emergency reactions, if access restrictions are defined for the End of Life state.

The TOE shall enter the End of Life state by itself when it cannot uphold the TSF.

O.SessionLimit Limitation of user sessions

The TSF allows the users to terminate their sessions and automatically terminate unused or stale sessions.

A.4.2 Security objectives for the operational environment

OE.Trust **Trustworthy users**

The operational environment shall ensure that the Administrators and Maintainers are trustworthy and well trained. This means that Maintainers perform their tasks diligently without deliberately compromising the security of the TOE, and that Administrators will not add users or assign roles to users who are not authorized.

OE.Audit ***Review and availability of audit records***

The Administrator shall assign the Auditor role to appropriate user identities. The Auditors shall define auditable events and perform audits. Users with an Auditor role shall neither hold an Administrator nor Maintainer role.

NOTE: (Informative) The TOE supports audit data suitable for forensic investigation. If this is intended by the security policy of the users, exported audit data shall be stored securely for forensic purposes and clearly assigned to a unique QKD module.

OE.SecureOp **Secure Operational environment**

The TOE shall be stored and operated inside an access controlled area, which ensures that only authorized personnel can physically access the TOE **and its user interfaces**. If access to the TOE by unauthorized personnel cannot be excluded, the TOE shall be removed from operation and all QKD keys created since it was last assured to have been continuously inaccessible to unauthorized personnel shall be considered as compromised. When designing the security perimeter it shall be taken into account that the PP claims high attack potential, i.e. the adversary may be backed by organized crime. Standard commercial warehouse protection shall not be considered as adequate protection.

The security perimeter shall ensure that any emanations of the TOE, e.g. electromagnetic, acoustic, power consumption profiles, cannot be detected outside the access controlled area, except signals or emanations conveyed on the QKD link.

OE.Personnel **Trustworthy personnel**

Personnel authorized to use the TOE are trustworthy and well trained. They will not intentionally misuse the TSF. In particular, users won’t identify as other users and will close sessions, while they do not actively interact with the TOE. Organizational means shall be in place to mitigate potential misconduct. Sample measures may comprise:

1) assignment of user IDs, which are not obvious to other users and shall be kept confidential by the users,

2) verification of correspondence of the logs for room access and TOE use, i.e. detection of users, who shouldn’t have been in the room,

3) security screening of personnel.

While none of these proposals is considered mandatory, any single one of these is neither considered sufficient.

A.4.3 Security objective rationale

The following table traces

1) the security objectives for the TOE back to

   a) threats countered by and

   b) OSPs enforced by that security objective, and

2) the security objective for the operational environment back to

   a) threats countered by,

   b) OSPs enforced by and

   c) assumptions upheld by that security objective.

Table 1: Security objective rationale

	T.ServAcc	**T.Session**	**T.QKDEave**	**T.QKDMani**	**T.ExplMal**	**T.Observe**	**OSP.QKDService**	**OSP.Audit**	***OSP.SecEoL***	**A.SecureOp**	**A.Maint**
O.Identify							×	×			
O.AccCtrl	×						×	×			
O.QKD			×	×			×				
O.QKDAuth			×	×			×				
O.Audit								×			
O.TST					×	×					
O.EMSec						×					
O.Sanitize					×				×		
O.SessionLimit		×									
OE.SecureOp					×	×	×	×		×	
OE.Personnel		×					×	×		×	
OE.Trust								×			×
OE.Audit								×			×

The following part of the chapter demonstrates that the security objectives counter all threats and enforce all OSPs, and the security objectives for the operational environment uphold all assumptions.

T.ServAcc

O.AccCtrl prohibits unauthorized access for identified users. It explicitly requires an unambiguous definition of authorized users for fetching any established key from each QKD module.

T.Session

O.SessionLimit allows the users to terminate sessions as required by OE.Personnel, when they leave their terminal. It furthermore eliminates sessions, which are not or cannot be closed. Therefore, session re-use by other users or an adversary is not possible.

T.QKDEave

O.QKD requires that any eavesdropping attempt on the QKD link will not leak any information about the QKD key. O.QKD requires that the classical channel of the QKD link is authenticated, which is provided by O.QKDAuth.

T.QKDMani

O.QKD ensures that modifications on the quantum channel are properly handled such that the final QKD key remains confidential. O.QKDAuth provides the required prerequisites for O.QKD and requires the TSF to provide an authenticated channel, where the integrity of the communication data exchanged on the classical channel of the QKD link is guaranteed.

T.ExplMal

OE.SecureOp excludes than an adversary has access to the TOE to induce any kind of malfunctions locally. O.TST monitors the operational conditions on the QKD link, which may be accessible to the adversary, and denies access to the key distribution services and QKD keys unless the TSFs are ensured.

O.TST furthermore verifies its own functionality by self-tests and also denies access in case the TSF are not assured. Therefore, spurious malfunctions cannot be exploited.

O.Sanitize requires that the TOE shifts to End of Life state, if the TSF cannot be upheld.

T.Observe

OE.SecureOp excludes that an adversary has access to the TOE and thus cannot observe the TOE locally, i.e. the adversary is restrained to monitoring or probing the QKD link. *O.TST* explicitly detects or suppresses active probing signals on the QKD link and stops operation in presence of such signals. O.EMSec requires the TSF to not leak any intelligible information on the QKD link.

OSP.QKDService

O.AccCtrl requires the TSF to restrict access to the key distribution services to authorized users by their identities, which are provided by O.Identify. According to OE.SecureOp only authorized personnel has access to the user interfaces of the TOE and OE.Personnel ensures that no authorized user will impersonate any other.

O.QKD requires the TSF to provide the said key distribution services. O.QKDAuth provides the required prerequisites for O.QKD.

OSP.Audit

O.Audit requires the TSFs to provide the specified audit information. It defines the Auditor role with exclusive permission to manage such information. It provides evidence, which enable the operational environment to verify origin and completeness of stored audit data. This evidence encompasses data stored in the environment for forensic purposes.

O.AccCtrl is used by the TSFs to enforce this exclusive permission of the Auditor role by user identities, which are provided by O.Identify. By requiring that Administrators cannot share an Auditor role, it furthermore ensures that operations of Administrators cannot be excluded from audits by themselves.

According to OE.SecureOp only authorized personnel has access to the user interfaces of the TOE and OE.Personnel ensures that no authorized user will impersonate any other.

OE.Audit requires the Administrator to assign Auditor roles, requires Auditors to define auditable events and to store exported audit data securely for forensic purposes.

OE.Trust requires the Administrator to be trustworthy in the sense that the Administrator does not create any proxy users with Auditor role.

OSP.SecEoL

O.Sanitize implements the required End of Life state.

A.SecureOp

OE.SecureOp defines the required level of security for the environment. It also states that the device shall be taken out of service if illicit access cannot be excluded. OE.Personnel reflects the requirements for trustworthy users, who may be allowed physical access to the TOE.

A.Maint

OE.Trust reflects A.Maint for all roles except Auditors, which is covered by OE.Audit.

A.5 Extended component definition

A.5.1 Quantum Key Distribution (FCS_QKD)

This clause describes the security functional requirements for the generation of QKD keys, which may be used as secrets for cryptographic purposes. The IT security functional requirements for a TOE are defined in an additional family Quantum Key Distribution (FCS_QKD) of the Class FCS (Cryptographic support).

Family Behaviour

Quantum Key Distribution relates to two or more end points (QKD modules) establishing a confidential, shared, random bit string. It uses a communication channel carrying quantum states, which by quantum physical principles cannot be eavesdropped on without introducing anomalies with high probability. The establishment is achieved using a protocol that limits the joint probability that the protocol does not abort and that

- any entity outside the modules has gained knowledge about the bit strings, or
- the shared bit strings are not identical in both QKD modules, or
- the distribution of bit strings has statistical properties different from uniform distribution

to a well-defined value. This value is called the security parameter of the quantum key distribution protocol.

Component levelling:

FCS_QKD.1 Prepare and Measure Quantum Key Distribution requires quantum key distribution between two QKD modules to be established using a prepare and measure protocol including information reconciliation and privacy amplification. The actual protocols and the algorithms for their application shall be chosen in accordance with the underlying security proof to support a claimed threshold value of the security parameter. The SFR depends on local random numbers to choose physical and cryptographic protocol parameters, and to randomly partition measurement data into private and public data. The SFR furthermore depends on an authenticated classical communication channel.

Management: FCS_QKD.1

There are no management activities foreseen.

Audit: FCS_QKD.1

There are no auditable events foreseen.

FCS_QKD.1 Prepare and Measure Quantum Key Distribution

Hierarchical to: No other components.

Dependencies: FCS_RNG.1 Random number generation
FPT_FLS.1 Failure with preservation of secure state
FTP_ITC.1 Inter-TSF trusted channel
FCS_CKM.4 Cryptographic key destruction

FCS_QKD.1.1 The TSF shall perform the quantum key distribution protocol according to [assignment: *QKD protocol*] [selection, choose one of: *between separate parts of the TOE, with a remote IT product*] in order to establish confidential, shared, random bit strings. The security parameter of the protocol shall not exceed [assignment: *security parameter threshold*] according to the associated composed security proof.

FCS_QKD.1.2 The TSF may repeat execution of the QKD protocol if it aborted or did not deliver a sufficient number of bits. The TSF shall ensure that the determining factors of the QKD protocol are assured for each individual execution of the QKD protocol. The TSF shall maintain a counter for all attempts of key establishment. The TSF shall [selection: *provide authorized users with the capability to request the current value of the attempt counter, deny protocol execution if the attempt counter exceeds [assignment: threshold for the attempt counter]*].

FCS_QKD.1.3 The TSF shall [selection: *prepare, measure*] [assignment: *description of quantum states*] and support [selection: *transmission, reception*] of these quantum states through an external interface.

FCS_QKD.1.4 The TSF shall perform [assignment: *list of post-processing algorithms before privacy amplification*] on the measurement data using the classical channel to establish a shared, corrected bit string.

FCS_QKD.1.5 The TSF shall keep track of deliberately disclosed information during post-processing and perform parameter estimation for [assignment: *list of parameters*]. Using these inputs the TSF shall deduce the privacy amplification ratio.

FCS_QKD.1.6 The TSF shall perform [assignment: *list of privacy amplification algorithms*] on the corrected bit strings using the classical channel to establish the confidential, shared, random bit strings based on the privacy amplification ratio.

User Application Notes

The dependency on FTP_ITC.1 refers to the classical channel. No confidentiality is required on this channel.

Implementations of FCS_QKD.1 may use preliminary data received on the classical channel. The confidential, shared, random bit string shall not be used, unless all communication on the classical channel pertaining to its establishment is proven to be authenticated.

The term "QKD protocol" refers to an algorithm which either aborts at any time or produces such a bit string in each module. FCS_QKD.1 requires that there is a valid security proof for the QKD protocol. This proof shall formally establish an upper bound for the joint probability that the QKD protocol does not abort and at least one of the properties "confidential", "shared", "random" cannot be assured, for all relevant attackers. This upper bound is denoted as the "security parameter"[12]. The said properties of the bit strings established by FCS_QKD.1 shall be interpreted as follows:

"confidential" means that no information about the bit strings (with the exception of their length) can be gained by eavesdropping or manipulating any information on any communication channel in between the modules,

"shared" means that the bit strings established in each module are identical, and

"random" means that the distribution of established bit strings is uniform, and their sequence is unpredictable; i.e., knowledge of any part of a bit string does neither provide any information on other bits already generated, nor on bits that will be generated in the future.

The QKD protocol may abort the establishment of the bit string based e.g., on parameter estimation results, and retry. FCS_QKD.1 includes any repeated executions of the QKD protocol until it either succeeds, or a failure of the TOE is detected[13]. In this case the TOE shall not execute the QKD protocol anymore and enter a secure state modelled by the FPT_FLS.1 dependency.

The TSF may use parts of the established bit string for internal purposes as TSF data e.g., for refreshing any secrets required for FTP_ITC.1. The "QKD key" is the part of the bit string, which either becomes TSF data used in any context unrelated to FCS_QKD.1 or user data. The TSF shall ensure that any parts of the bit string used internally by FCS_QKD.1 are used for a single purpose and are not exported as parts of QKD keys. Partitioning of internal shared bit strings into internal TSF data and QKD keys shall be consistent throughout the entire TOE.

FCS_QKD.1 may repeat the execution of the QKD protocol to match length requirements for the QKD key. FCS_QKD.1 may also maintain a pool of pre-generated bit strings as data under control of the TSF.

The security parameter denotes the maximum probability that any of the properties of the bit strings is not assured during a single execution of the QKD protocol. The actual value of a single protocol run is usually a composition of an ideal protocol run and variable values, e.g. concerning the security parameters of the authentication protocol. The security parameter threshold shall provide an upper bound for such current values for single protocol runs.

Therefore, the TSF shall track any factors that may influence the current value of the security parameter, e.g. by using TSF data taken from bit strings established in previous executions of the protocol. The TSF shall take such effects into account in considering the claim of the security parameter threshold in FCS_QKD.1.1.

The choice of the value of the security parameter threshold will be tied to an assumption about how often a QKD generation attempt is made. The key generation attempt counter tracks the number of these attempts. FCS_QKD.1.2 allows the user to query this counter and perform risk management on the users’ side or requires the TSF to enforce a limit. PP/ST authors may use the FMT_MTD family to manage the limit. The key generation attempt counter shall never be reset. The conditions for the limit management and any security implications related to limit management shall be detailed in the user guidance. If automatic denial of protocol execution is selected in FCS_QKD.1.2, then denial shall be implemented by FPT_FLS.1.

[12] For the definition of QKD protocol security see e.g. [i.8], page 4 for perfect security, and page 5 for approximate security. Note that this PP defines security only in terms of secrecy and correctness as defined in this reference. The concept of "robustness" introduced in the reference, which involves modelling the quantum channel in the absence of an eavesdropper, is excluded and it is appropriate to set the robustness parameter formally to zero.

[13] This shall not imply resetting any internal states when the protocol succeeds.

The security parameter for a single run of the QKD protocol might not be known by the end user but FCS_QKD.1.1 enforces that it does not exceed the security parameter threshold, which is generally known in advance by end user applications.

Security proofs may assume properties such as but not limited to ideal random number generators (cf. FCS_RNG.1 dependency) or ideal classical channels (FTP_ITC.1). The security statements about the QKD protocol may be deduced from security statements about individual components. In such cases the exact security parameters of some components might not be known and an educated guess may be used instead. If such security parameters are assumed or chosen as some value, the ST/PP author shall detail these choices explicitly.

Editorial note: The following paragraph contains preliminary text and further revision is likely.

Evaluation of the security proofs themselves is not part of the evaluation of FCS_QKD.1. The security proof shall be approved by the responsible certification body. A certification body may take the opinion of a reputable group, such as a standards developing organisation, into account in deciding whether or not to approve a security proof. The evaluation of FCS_QKD.1 of class ASE shall determine the adequacy of the chosen security proof. The evaluation of class ADV shall determine whether and how the assumptions of the security proof are ensured by the implementation of FCS_QKD.1. The evaluation of class AVA shall determine whether and how any limitations of the model underlying the security proof, or any imperfections of its implementation impact the claimed properties of the confidential, shared, random bit strings. It is not required to determine how such effects affect the security parameters.

In order to support the evaluation, the developer or sponsor shall deliver the complete, correct, and comprehensible security proof, and a detailed mapping of the assumptions of the security proof to the implementation.

The term "privacy amplification" refers to the process of distilling confidential data from potentially compromised data. The "privacy amplification ratio" determines the amount of confidential information that can be distilled from the shared, corrected bit string.

Operations

Assignment:

In FCS_QKD.1.1, the PP/ST author should specify the QKD protocol such that it is unambiguously linked to a valid security proof.

Selection:

In FCS_QKD.1.1, the PP/ST author should select whether the TOE contains all modules i.e., the bit strings are established between separated parts of the same TOE, or the TOE refers to only a single module communicating with another IT product.

Assignment:

In FCS_QKD.1.1, the PP/ST author should specify the upper limit on the security parameter for a single run of the composed QKD protocol. This choice may affect the post-processing during the establishment of the bit string. The security parameter threshold refers to the composed security parameter including all sub-protocols, e.g. authentication. It shall take into account that values of security parameters of sub-protocols may accumulate.

Selection:

In FCS_QKD.1.2, the PP/ST author should select whether the TOE shall report its key generation attempt counter or shall shift to failure state, when a defined threshold is exceeded. Both options may be selected together.

Assignment:

In FCS_QKD.1.2, the PP/ST author, dependent on the selection, should specify the threshold for the key generation attempt counter, which when exceeded will cause the TSF to shift to failure state.

Selection:

In FCS_QKD.1.3, the PP/ST author should select whether the TSF prepare or measure quantum states or do both. A TOE comprising all modules will necessarily require both selections.

Assignment:

In FCS_QKD.1.3, the PP/ST author should specify the quantum states exchanged (e.g., coherent states), the physical instantiation of those states (e.g., photons or electrons) and the type of quantisation bases (e.g., polarisation) used for the quantum channel.

Selection:

In FCS_QKD.1.3, the PP/ST author, dependent on the selection, should select whether the TOE transmits or receives quantum states or does both. This is immediately linked to whether it is preparing and thus transmitting or measuring and thus receiving quantum states.

Assignment:

In FCS_QKD.1.4, the PP/ST author should list all post-processing algorithms implemented by the TSF and used before privacy amplification. The algorithms listed shall be clearly defined. References to the security proof might be sufficient if it details the algorithms appropriately.

In FCS_QKD.1.5, the PP/ST author should list the parameters determined by the TSF to deduce the required privacy amplification ratio and select algorithms along with their parameters for privacy amplification such that the claimed value of the security parameter threshold is assured.

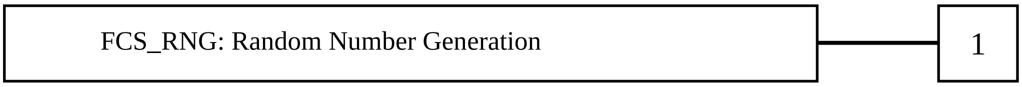
In FCS_QKD.1.6, the PP/ST author should list all privacy amplification algorithms implemented by the TSF. The algorithms listed shall be clearly defined. References to the security proof might be sufficient if it details the algorithms appropriately.

A.5.2 Random number generation (FCS_RNG)

Family Behaviour

This family defines quality requirements for the generation of random numbers that are intended to be used for security critical mechanisms such as cryptographic purposes or choices of QKD protocol parameters.

Component levelling:



FCS_RNG.1 Random number generation, requires that the random number generator implements defined security capabilities and that the random numbers meet a defined quality metric.

Management: FCS_RNG.1

There are no management activities foreseen.

Audit: FCS_RNG.1

There are no auditable events foreseen.

FCS_RNG.1 Random number generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*] random number generator that implements: [assignment: *list of security capabilities*].

FCS_RNG.1.2 The TSF shall provide random numbers that meet [assignment: *a defined quality metric*].

NOTE: (Informative) A physical RNG produces high-entropy random numbers using a dedicated noise source based on physical random processes. This includes RNGs based on quantum principles. A non-physical true RNG uses non-dedicated noise sources such as system data (e.g. interrupts) or human interaction (e.g. keystrokes or mouse movements). A deterministic RNG produces random numbers by applying a deterministic algorithm to a high-entropy random seed. A hybrid RNG combines the principles of physical and deterministic RNGs. A hybrid physical RNG is a physical RNG with cryptographic post-processing with memory that produces high-entropy random numbers. A hybrid deterministic RNG is a deterministic RNG that is regularly reseeded with high-entropy inputs.

A.5.3 Sanitizing on State Change (FDP_RIP.4)

Family Behaviour

The family is defined in [2]. In this PP another component is defined.

Component levelling:

FDP_RIP: Residual Information Protection
4

FDP_RIP.4 Sanitizing on State Change, requires that a well-defined set of data is erased, when the TSF detect some event.

NOTE: (Informative) FDP_RIP.4 was chosen since FDP_RIP.3 has already been defined for different purposes in another PP.

Management: FDP_RIP.4

There are no management activities foreseen.

Audit: FDP_RIP.4

There are no auditable events foreseen.

FDP_RIP.4 Sanitizing on State Change

	Hierarchical to:	No other components.
	Dependencies:	No dependencies.
FCS_RIP.4.1	The TSF shall ensure that any previous information content about [assignment: *list of assets, user data, TSF data*] is made unavailable upon [assignment: *list of events detected by the TSF*].	

A.5.4 Emanation of TSF and user data (FPT_EMS)

The family FPT_EMS (TOE Emanation) of the class FPT (Protection of the TSF) is defined here to describe the IT security functional requirements of the TOE. The TOE shall prevent attacks against secret data stored in and used by the TOE where the attack is based on external observable physical phenomena of the TOE. Examples of such attacks are evaluation of TOE’s electromagnetic radiation, simple power analysis (SPA), differential power analysis (DPA), timing attacks, etc. This family describes the functional requirements for the limitation of intelligible emanations being not directly addressed by any other component of CC part 2.

Family Behaviour

This family requires that leakage of the TOE cannot be used to compromise sensitive TSF data or user data. The leakage may occur when TSF data is transferred or processed by the TOE hardware.

Component levelling:

FPT_EMS: Emanation of TSF and user data
1

FPT_EMS.1 Emanation of TSF and user data, requires the TOE to protect TSF data and or user data against leakage that may be generated during transfer or processing of such data inside the TOE.

Management: FPT_EMS.1

There are no management activities foreseen.

Audit: FPT_EMS.1

There are no auditable events foreseen.

FPT_EMS.1 Emanation of TSF and user data

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_EMS.1.1 The TSF shall ensure that the TOE does not emit emissions over its attack surface in such amount that these emissions enable access to TSF data and user data as specified in the following table:

Table 2: Definition of Side-channel Protection

ID	Emanation	Attack Surface	TSF data	User Data
1	[assignment: list of types of emissions]	[assignment: list of types of attack surface]	[assignment: list of types of TSF data]	[assignment: list of types of user data]

A.6 Security requirements

The CC allows several operations to be performed on functional requirements: *refinement*, *selection*, *assignment*, and *iteration.* Each of these operations is used in this PP.

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is (i) denoted by the word "refinement" in **bold** text and the added/changed words are in bold text, or (ii) directly included in the requirement text as **bold** text. In cases where words from a CC requirement component were deleted, these words are ~~crossed out~~.

The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections that have been made by the PP authors are denoted as *italic* text and the original text of the component is given by a footnote. Selections to be filled in by the ST author appear in square brackets with an indication that a selection is to be made, [selection:], and are *italicized*.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments that have been made by the PP authors are denoted by showing as *italic* text and the original text of the component is given by a footnote. Assignments to be filled in by the ST author appear in square brackets with an indication that an assignment is to be made [assignment:] and are *italicized*.

The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing a slash "/" and the iteration indicator after the component identifier.

A.6.1 Security functional requirements

A.6.1.1 User Identification and Management

The base PP assumes that access to the TOE is controlled by the environment and that only trustworthy personnel may be granted such access. Therefore, the SFR only model identification. Authentication of users is handled in packages or may be modelled by the ST author.

FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:

(1) User Identity,

(2) Role[14].

FIA_USB.1 User-subject binding

Hierarchical to: No other components.

Dependencies: FIA_ATD.1 User attribute definition

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:

[14] [assignment: *list of security attributes*]

(1) User Identity,

(2) Role[15].

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: *the initial role of the user is Unidentified User* [16].

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users:

(1) after successful identification of the user, the security attribute Role of the subject shall be set according to the UDR of the identified user.[17]

FIA_UID.1 Timing of identification

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UID.1.1 The TSF shall allow *no TSF-mediated actions*[18] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

FTA_SSL.3 TSF-initiated termination

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_SSL.3.1 The TSF shall terminate an interactive session after a [assignment: *time interval of user inactivity*].

FTA_SSL.4 User-initiated termination

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_SSL.4.1 The TSF shall allow user-initiated termination of the user's own interactive session.

FMT_MTD.1/Adm Management of TSF data – Administrator

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1 The TSF shall restrict the ability to

(1) create and delete[19] the *User Definition Records of an identified user*[20] to *Administrator*[21],

[15] [assignment: *list of user security attributes*]

[16] [assignment: *rules for the initial association of attributes*]

[17] [assignment: *list of security attributes*].

[18] [assignment: *list of TSF mediated actions*]

[19] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]

[20] [assignment: *list of TSF data*]

[21] [assignment: *the authorized identified roles*]

(2) *modify*[22] the *Role of an identified user*[23] to *Administrator*[24],

(3) *change_default*[25] the *Role of an identified user*[26] to *none*[27].

***Application Note 6*:** **The refinements of FMT_MTD.1.1 are made to avoid iterations of the component. Strictly,** Role is a security attribute and should be covered by FMT_MSA.1. The SFR has not been split to preserve the context for better readability. Therefore, this SFR may be used to resolve dependencies on FMT_MSA.1 in the context of the Access Control SFP.

A.6.1.2 Access Control

FDP_ACC.1 **Subset access control - Access Control SFP**

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1 The TSF shall enforce the *Access Control SFP*[28] on

subjects: Administrator, Auditor, Maintainer, Key Requester, [assignment: other roles];

objects: key distribution services, QKD keys, ADR;

operations: export, delete, access[29].

FDP_ACF.1 **Security attribute based access control - Access Control SFP**

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1 The TSF shall enforce the *Access Control SFP*[30] to objects based on the following:

(1) subjects: identified users (attribute: *Role*),

(2) objects: QKD keys (attributes: receiver, owner), key distribution services (attribute: operational state), ADR (attribute: exported)[31].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

(1) identified users with Role Key Requester are allowed to export QKD keys, if the receiver attribute of the QKD key matches the user identity

(2) identified users with Role Key Requester are allowed to access the key distribution services to request establishment of QKD keys,

(3) identified users with Role Auditor are allowed to export and delete ADR,

[22] [selection: *change_default, query, modify, delete, clear,[assignment: other operations]*]

[23] [assignment: *list of TSF data*]

[24] [assignment: *the authorized identified roles*]

[25] [selection: *change_default, query, modify, delete, clear,[assignment: other operations]*]

[26] [assignment: *list of TSF data*]

[27] [assignment: *the authorized identified roles*]

[28] [assignment: *access control SFP*]

[29] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

[30] [assignment: *access control SFP*]

[31] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

(4) [assignment: additional rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects][32].

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules:

[assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

(1) Neither the key distribution services nor any QKD keys shall be accessed, unless the operational state is QKD state,

(2) ADR shall not be deleted unless the attribute "exported" is true and the identified user has the Role Auditor,

(3) [assignment: additional rules, based on security attributes, that explicitly deny access of subjects to objects][33].

Application Note 7: The security attribute receiver may be implemented as a list of user identities, e.g. one for each QKD module.

NOTE: (Informative) The TSF ensure that each QKD key is exported only once per QKD module by deleting any exported QKD key from the QKD module immediately after export (cf. FCS_CKM.4).

The concept of having an owner of the key establishment process distinct from the receivers of the final key facilitates more sophisticated role models e.g., a role responsible to initiate key establishments for other users. It also allows to specify that a different user than the requester is allowed to receive the key, which does not require the initial Key Requester to fetch the key at both modules.

FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1 The TSF shall enforce the *Access Control SFP*[34] to restrict the ability to *modify*[35] the security attributes *operational state*[36] ~~to~~ ***according to the following list:***

(1) the Maintainer role may set Calibration state from any operational state except End of Life,

(2) the Maintainer role may set QKD state from Calibration state,

(3) the Key Requester may set the receiver attribute, if the owner attribute matches its user identity,

(4) the [assignment: list of authorized roles] may set End of Life from any operational state. [37]

[32] [assignment: *additional rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

[33] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

[34] [assignment: *access control SFP(s), information flow control SFP(s)*]

[35] [selection: *change_default, query, modify, delete,* [assignment: *other operations*]]

[36] [assignment: *list of security attributes*]

[37] [assignment: *the authorized identified roles*]

Application Note 8: The TOE shall maintain a state-machine for operational states as proposed in clause A.1.3, Life-cycle. For the base PP this state-machine consists of: Calibration state, QKD state, Failure state, and End of Life. The ST author shall refine FMT_MSA.1, if more operational states are supported. Changing the operational state to Failure state is performed by the TSF, e.g. FPT_TST.1.

For rule 3 the Key Requester may specify the receiver attribute with the initial request despite FMT_MSA.3.

FMT_MSA.2 Secure security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for *security attributes Role*[38].

Refinement: **An insecure value for the attribute Role is the assignment of an Auditor and Administrator Role to the same User Identity, even if they are not assigned simultaneously.**

The receiver attribute shall only refer to user identities, which hold the Key Requester Role.

FMT_MSA.3 Static attribute initialisation

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3.1 The TSF shall enforce the *Access Control SFP*[39] to provide *restrictive*[40] default values for security attributes that are used to enforce the SFP, **i.e. the receiver and owner attributes of a QKD key shall be the user identity of the Key Requester, who requested its establishment, and new ADR shall have the attribute "exported" set to false.**

FMT_MSA.3.2 The TSF shall allow ~~**the**~~ *no-one*[41] to specify alternative initial values to override the default values when an object or information is created.

NOTE: (Informative) There is no object created bearing the operational state, and initial values for Roles of identified users are handled in FIA_USB.1.

FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_ITT.1.1 The TSF shall protect TSF data-from *modification*[42] when it is transmitted between separate parts of the TOE.

FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1 The TSF shall restrict the ability to

[38] [assignment: *list of security attributes*]

[39] [assignment: *access control SFP(s), information flow control SFP(s)*]

[40] [selection, choose one of: *restrictive, permissive,* [assignment: *other property*]]

[41] [assignment: *the authorized identified roles*]

[42] [selection: *disclosure, modification*]

(1) *change_default, query, modify*[43] the *CD*[44] to *Maintainer*[45]*,*

(2) *set the exported attribute for*[46] the *ADR*[47] ***by actual export of the ADR*** *to Auditor*[48]*,*

(3) *select events to generate by FAU_GEN.1*[49] the *ADR*[50] to *Auditor*[51]*,*

(4) *define, modify*[52] the *threshold for actions to be taken according to FAU_STG.3*[53] to *Auditor*[54]

(5) *change_default, query, modify*[55] the *threshold for maximal number of consecutive unsuccessful QKD key establishment attempts according to FPT_TST.1*[56] to *[assignment: the authorized identified roles].*

FMT_MTD.1/QAK Management of TSF data

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1 The TSF shall restrict the ability to *establish, query, modify*[57] the *QAK*[58] to *none*[59].

A.6.1.3 Audit Data

Audit data generation is mainly intended for forensic purposes. It should at least be difficult for any single user to modify the TOE undetected. For that reason, the audit data are designed to reveal gaps. Unintentional loss of audit data is mitigated by requiring export before deletion. Since user administration and audit administration are strictly separated, dual-control is proposed. Finally, FDP_DAU.1 is refined to prevent forging of exported logs.

For high-security applications the ST author is advised to consult with the risk owner and their national CB to agree upon an audit policy.

[43] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]

[44] [assignment: *list of TSF data*]

[45] [assignment: *the authorized identified roles*]

[46] [selection: *change_default, query, modify, delete, clear,[assignment: other operations]*]

[47] [assignment: *list of TSF data*]

[48] [assignment: *the authorized identified roles*]

[49] [selection: *change_default, query, modify, delete, clear,[assignment: other operations]*]

[50] [assignment: *list of TSF data*]

[51] [assignment: *the authorized identified roles*]

[52] [selection: *change_default, query, modify, delete, clear,[assignment: other operations]*]

[53] [assignment: *list of TSF data*]

[54] [assignment: *the authorized identified roles*]

[55] [selection: *change_default, query, modify, delete, clear,[assignment: other operations]*]

[56] [assignment: *list of TSF data*]

[57] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]

[58] [assignment: *list of TSF data*]

[59] [assignment: *the authorized identified roles*]

FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

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 *not specified*[60] level of audit; and

c) start-up after power-up,

d) creation and deletion of User Definition Records (cf. FMT_MTD.1/Adm (1))

e) modification of the user security attribute Role *(cf. FMT_MTD.1/Adm (2))*

f) Failure with preservation of secure state (cf. FPT_FLS.1/Fail): entering and exiting secure state,

g) deletion and export of audit records (cf. FMT_MTD.1 (2), FDP_ACF.1)

h) selection, de-selection and clearance of events causing audit events (cf. FMT_MTD.1 (3))

i) changes with respect to possible audit storage failure (cf. FAU_STG.3)

j) requests and changes of calibration data (cf. FMT_MTD.1 (1)),

k) shifts in operational state, and recording the user's identity initiating the shift, for manual state shifts,

l) access to the key distribution services,

m) [assignment: additional specifically defined auditable events][61].

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

a) ~~Date and time of the event~~**[assignment: *information required to uniquely identify separate events and ensure their completeness and chronological order*],** type of event, subject identity (if applicable), 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*].

Application Note 9: The Auditor shall only be allowed to exclude the event l) and any additional auditable events m) from auditing. With the definition of the "*not specified* level of audit" in FAU_GEN.1.1 b) no additional events are required by the TSF to generate an audit record.

Application Note 10: Confidential user data and confidential TSF data shall not be contained in the audit logs.

FDP_DAU.1 Basic Data Authentication

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_DAU.1.1 The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of *ADR*[62].

FDP_DAU.1.2 The TSF shall provide *Auditors*[63] with the ability to verify evidence of the validity of the indicated information.

[60] [*selection: choose one of: minimum, basic, detailed, not specified*]

[61] [assignment: *other specifically defined auditable events*]

[62] [assignment: *list of objects or information types*]

[63] [assignment: *list of subjects*]

Refinement: **Validity shall include that the origin of the audit data can be verified even after export from the TOE.**

FAU_STG.1 Protected audit trail storage

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1.2 The TSF shall be able to *prevent*[64] unauthorized modifications to the stored audit records in the audit trail.

FAU_STG.3 Action in Case of Possible Audit Data Loss

Hierarchical to: No other components.

Dependencies: FAU_STG.1 Protected audit trail storage

FAU_STG.3.1 The TSF shall [assignment: *actions to be taken in case of possible audit storage failure*] if the audit trail exceeds *the limit defined by an Auditor*[65].

FCS_COP.1/Aud Cryptographic operation – Proof of Audit Data

Hierarchical to: No other components.

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

FCS_COP.1.1 The TSF shall ~~perform~~ **provide** *a proof of origin for audit logs*[66] in accordance with a specified ~~cryptographic~~ **signature** algorithm [assignment: *signature algorithm*][67] and cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].

Application Note 11: It is not acceptable to use message authentication codes relying on shared secrets, unless these are held in a tamper resistant IT device. If the Auditor may forge exported ADR, Auditors might by-pass forensic investigations.

A.6.1.4 Reaching and preserving secure states

FPT_PHP.3 Resistance to physical attack

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_PHP.3.1 The TSF shall resist *active probing via the QKD link*[68] to the *internal states of the TSF*[69] by responding automatically such that the SFRs are always enforced.

Refinement: **The TSF shall implement appropriate mechanisms to continuously, i.e. at any time during the operational life-cycle phase, counter active probing via the QKD link. As response entering FPT_FLS.1/Fail or FPT_FLS.1/EoL shall be chosen as appropriate.**

[64] [selection, choose one of: *prevent, detect*]

[65] [assignment: *pre-defined limit*]

[66] [assignment: *list of cryptographic operations*]

[67] [assignment: *cryptographic algorithm*]

[68] [assignment: *physical tampering scenarios*]

[69] [assignment: *list of TSF devices/elements*]

FPT_EMS.1 Emanation of TSF and user data

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_EMS.1.1 The TSF shall ensure that the TOE does not emit emissions over its attack surface in such amount that these emissions enable access to TSF data and user data as specified in the following table:

Table 3: Definition of Side-Channel Protection

ID	Emanation	Attack Surface	TSF data	User Data
1	Timing of signals	QKD link	any confidential TSF data	any confidential user data
2	Signal strength, waveform, or quantum state	QKD link	any confidential TSF data	any confidential user data

Application Note 12: The ST author shall ask the certification body, whether additional emanations and attack surfaces are to be considered and refine FPT_EMS.1 accordingly.

NOTE: (Informative) As a reminder, data sent intentionally through the QKD link is not considered confidential.

FPT_TST.1 TSF testing

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TST.1.1 The TSF shall run a suite of self-tests *during initial start-up, periodically during normal operation, at the request of the authorized user, and at the additional conditions: [assignment: additional conditions under which self-test should occur]*[70] to demonstrate the correct operation of *the TSF*[71].

FPT_TST.1.2 The TSF shall provide authorized users with the capability to verify the integrity of *TSF data*[72].

FPT_TST.1.3 The TSF shall provide authorized users with the capability to verify the integrity of *all cryptographic operations, including random number generators (according to FCS_RNG.1), establishment of confidential, shared, random bit strings (according to FCS_QKD.1); the TSF implementation; [assignment: additional parts of TSF]*[73].

Application Note 13: The ST author shall define the Roles authorized to request self-tests and to use the capabilities provided by the TSF as stated in FPT_TST.1.2 and FPT_TST.1.3. The author may use iterations to restrict the capability to verify the integrity of parts of TSF data or parts of TSF to specific authorized user Roles.

FRU_FLT.2 Limited fault tolerance

Hierarchical to: FRU_FLT.1 Degraded fault tolerance

Dependencies: FPT_FLS.1 Failure with preservation of secure state

FRU_FLT.2.1 The TSF shall ensure the operation of all the TOE's capabilities when the following ~~failures~~ **circumstances** occur: *exposure to operating conditions which are not detected in the requirement FPT_FLS.1/EoL (Failure with preservation of secure state)*[74].

Application Note 14: Note that the TOE does not always actually detects faults or failures and then corrects them in order to guarantee further operation of all the TOE's capabilities. The TOE will ensure the operation of the TOE's capabilities by stable functional design within the limits of operational

[70] [selection: *during initial start-up, periodically during normal operation, at the request of the authorized user, at the conditions [assignment: conditions under which self test should occur]*]

[71] [selection: *[assignment: parts of TSF], the TSF*]

[72] [selection: *[assignment: parts of TSF data], TSF data*]

[73] [selection: *[assignment: parts of TSF], TSF*]

[74] [assignment: *list of type of failures*].

conditions (which may include but are not limited to power supply, temperature, mean number of photons per pulse, …).

FPT_FLS.1/Fail **Failure with preservation of secure state**

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur:

(1) self-test (FPT_TST.1) fails recoverable,

(2) runs of the QKD protocol according to the requirement FCS_QKD.1 abort or the authentication fails [assignment: a defined number of consecutive times] consecutive times,

(3) no new QAK is available at the end of a QKD transaction[75].

Refinement: **In this state the security attribute operational state shall be set to Failure state.**

FPT_FLS.1/EoL **Failure with preservation of secure state**

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures **or circumstances** occur:

(1) self-test (FPT_TST.1) fails irrecoverable,

(2) exposure to operating conditions which may not be tolerated according to the requirement FRU_FLT.2 (Limited fault tolerance) and where therefore a malfunction could occur,

(3) an authorized user requests entering this state.[76]

Refinement: **In this state all confidential data shall be deleted from the TOE. If data cannot be erased, it shall be stored inaccessible considering high attack potential. In this case ratings shall consider that the environment for the TOE in this state may be very different from the operational environment reflected by the assumptions in this PP.**

Stored ADR may be accessible and may be erased in end of life state. The TSF may offer a pre-defined Auditor account for this purpose.

A.6.1.5 Secure classical channel

FTP_ITC.1 **Inter-TSF trusted channel – Classical Channel**

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_ITC.1.1 The TSF shall provide a communication channel, **called the classical channel, in** between ~~itself and another trusted IT product~~ **both QKD modules** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification ~~**or disclosure**~~.

FTP_ITC.1.2 The TSF shall permit ***[selection: QKD Transmitter, QKD receiver, both QKD modules]***[77] to initiate communication via the ~~trusted channel~~ **classical channel**.

[75] [assignment: *list of types of failures in the TSF*].

[76] [assignment: *list of types of failures in the TSF*].

[77] [selection: *the TSF, another trusted IT product*]

FTP_ITC.1.3 The TSF shall initiate communication via the ~~**trusted channel**~~ **classical channel** for *all classical communication required as authenticated by the QKD protocol (FCS_QKD.1).*[78]

.FCS_COP.1/CCI Cryptographic operation – Classical Channel Integrity

Hierarchical to: No other components.

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

FCS_COP.1.1 The TSF shall perform *data authentication*[79] in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].

Refinement: The TSF shall limit the use of any cryptographic keys and enforce session termination or re-keying when the key may be overused, i.e. [assignment: *list of conditions for overuse*].

Application Note 15: Where the data authentication is not included in the composed security parameter that would necessarily prevent overuse of keys, "Conditions for overuse" shall include at least a maximum number of elementary operations for a single key, e.g. single message block operations for a block cipher, and a maximum time a single key may be used. (See the User Application Notes for FCS_QKD.1 in A.5.1).

A.6.1.6 QKD Key Establishment

FCS_QKD.1 Prepare and Measure Quantum Key Distribution

Hierarchical to: No other components.

Dependencies: FCS_RNG.1 Random number generation
FPT_FLS.1 Failure with preservation of secure state
FTP_ITC.1 Inter-TSF trusted channel
FCS_CKM.4 Cryptographic key destruction

FCS_QKD.1.1 The TSF shall perform the quantum key distribution protocol according to [assignment: *QKD protocol*] *between separate parts of the TOE*[80] in order to establish confidential, shared, random bit strings. The security parameter of the protocol shall not exceed [assignment: *security parameter threshold*] according to the associated composed security proof.

FCS_QKD.1.2 The TSF may repeat execution of the QKD protocol if it aborted or did not deliver a sufficient number of bits. The TSF shall ensure that the determining factors of the QKD protocol are assured for each individual execution of the QKD protocol. The TSF shall maintain a counter for all attempts of key establishment. The TSF shall *provide authorized users with the capability to request the current value of the attempt counter* ***and*** *deny protocol execution if the attempt counter exceeds [assignment: threshold for the attempt counter]*[81].

FCS_QKD.1.3 The TSF shall *prepare* ***and*** *measure*[82] [assignment: *description of quantum states*] and support *transmission* ***and*** *reception*[83] of these quantum states through an external interface.

FCS_QKD.1.4 The TSF shall perform [assignment: *list of post-processing algorithms before privacy amplification*] on the measurement data using the classical channel to establish a shared, corrected bit string.

[78] [assignment: *list of functions for which a trusted channel is required*]

[79] [assignment: *list of cryptographic operations*]

[80] [selection, choose one of: *between separate parts of the TOE, with a remote IT product*]

[81] [selection: *provide authorized users with the capability to request the current value of the attempt counter, deny protocol execution if the attempt counter exceeds [assignment: threshold for the attempt counter]*].

[82] [selection: *prepare, measure*]

[83] [selection: *transmission, reception*]

FCS_QKD.1.5 The TSF shall keep track of deliberately disclosed information during post-processing and perform parameter estimation for [assignment: *list of parameters*]. Using these inputs the TSF shall deduce the privacy amplification ratio.

FCS_QKD.1.6 The TSF shall perform [assignment: *list of privacy amplification algorithms*] on the corrected bit strings using the classical channel to establish the confidential, shared, random bit strings based on the privacy amplification ratio.

Application Note 16: Guidance for the use of the SFR can be found in the User Application Notes to the extended component definition in sect. A.5.1.

The threshold for the *attempt counter* in FCS_QKD.1.2 shall be chosen to be consistent with high attack potential. ST authors are advised to consult with their responsible certification body for adequate choices.

FCS_RNG.1 Random number generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a **[selection: *physical, hybrid physical*]**[84] random number generator that implements: [assignment: *list of security capabilities*].

FCS_RNG.1.2 The TSF shall provide [selection: *bits, octets of bits, numbers [assignment: format of the numbers]*] that meet [assignment: *a defined quality metric*].

Application Note 17: The evaluation of the random number generator shall follow a recognized methodology e.g., AIS31 cf. [i.5]. Clause A.8 provides examples for the security capabilities and quality metrics used in some national certification schemes.

FDP_ETC.1 Export of user data without security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FDP_ETC.1.1 The TSF shall enforce the *Access Control SFP*[85] when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.1.2 The TSF shall export the user data without the user data's associated security attributes.

Application Note 18: The ST author may require FDP_ETC.2 instead of the stated FDP_ETC.1, if a more complex internal key storage is implemented.

A.6.1.7 Management

FMT_SMR.1 Security roles

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1 The TSF shall maintain the roles: *Unidentified User, Identified User, Administrator, Auditor, Maintainer, Key Requester, [selection: [assignment: other roles], no other roles]*[86].

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

[84] [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*]

[85] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

[86] [assignment: *authorized identified roles*]

(1) Management of User Definition Records and their security attributes (FMT_MTD.1/Adm),

(2) Management of TSF data for audits and calibrations (FMT_MTD.1),

(3) Management of QKD Authentication Keys (FMT_MTD.1/QAK),

(4) [assignment: list of additional security management functions to be provided by the TSF][87]*.*

FCS_CKM.4 Cryptographic key destruction

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]

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*].

Refinement: The destruction of cryptographic keys or QKD keys shall ensure that any previous information content of the resource about the key is made unavailable upon the deallocation of the resource. The resource of the successfully established QKD key shall be deallocated in the respective QKD module immediately after export to the user, after a defined time-out [assignment: *maximum time-out value*], and [assignment: *other events to trigger deletion of the QKD key*]. Cryptographic keys as well as QKD keys and UDR shall be destroyed before an End of Life state is reached.

Application Note 19: The cryptographic keys required for the communication using the classical channel between both QKD modules shall be destroyed shortly after each QKD transaction. After their usage, the QKD Authentication Keys shall exist at most for the duration required for any subsequent cryptographic key derivation.

The term "maximum time-out value" shall allow ST authors to manage the time-out e.g., by refining FMT_MTD.1.1. However, any managed time-out value shall not exceed the value given here.

A.6.2 Security assurance requirements

The PP requires the TOE to be evaluated to EAL4 augmented with AVA_VAN.5 and ALC_DVS.2.

A.6.2.1 Security assurance requirements rationale

QKD is considered to provide security in the presence of quantum computers and other bespoke attack techniques, which are currently available or is anticipated to become available to institutional attackers. Such attacks may compromise standard cryptographical security involving a high attack potential. Therefore, the augmentation by AVA_VAN.5 has been chosen to provide assurance against high attack potential.

EAL 4 as base package was chosen since it is the smallest assurance package, which fulfils all dependencies of AVA_VAN.5.

Since for high security applications institutional attackers may try to compromise development and manufacturing, ALC_DVS.2 has been chosen to provide more stringent processes, which make such interference more complicated or detectable.

A.6.3 Security requirements rationale

A.6.3.1 Dependency rationale

This chapter demonstrates that each dependency on the security requirements is either satisfied, or justifies the dependency not being satisfied.

[87] [assignment: *list of management functions to be provided by the TSF*]

Table 4: Dependency rationale

SFR	Dependencies of the SFR	SFR components
FAU_GEN.1	FPT_STM.1 Reliable time stamps	Dependency on FPT_STM.1 is not fulfilled (see rationale for O.Audit)
FAU_STG.1	FAU_GEN.1 Audit data generation	FAU_GEN.1
FAU_STG.3	FAU_STG.1 Protected audit trail storage	FAU_STG.1
FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	FCS_QKD.1
FCS_COP.1/Aud	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	The ASK used by this SFR is installed when delivered; no import or generation required. FCS_CKM.4
FCS_COP.1/CCI	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	Initial QAK delivered by manufacturer, subsequent QAK are provided by FCS_QKD.1 FCS_CKM.4
FCS_QKD.1	FCS_RNG.1 Random number generation FTP_ITC.1 Inter-TSF trusted channel	FCS_RNG.1 FTP_ITC.1
FCS_RNG.1	No dependencies	No dependencies
FDP_ACC.1	FDP_ACF.1 Security attribute based access control	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.1 FMT_MSA.3
FDP_DAU.1	No dependencies	No dependencies
FDP_ETC.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	FDP_ACC.1
FIA_ATD.1	No dependencies	No dependencies
FIA_UID.1	No dependencies	No dependencies
FIA_USB.1	FIA_ATD.1 User attribute definition	FIA_ATD.1
FMT_MSA.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.1 FMT_SMR.1 FMT_SMF.1
FMT_MSA.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FDP_ACC.1 FMT_MSA.1 is resolved by FMT_MTD.1/Adm FMT_SMR.1
FMT_MSA.3	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1 FMT_SMR.1
FMT_MTD.1	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMR.1 FMT_SMF.1
FMT_MTD.1/Adm	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMR.1 FMT_SMF.1
FMT_MTD.1/QAK	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMR.1 FMT_SMF.1
FMT_SMF.1	No dependencies	No dependencies
FMT_SMR.1	FIA_UID.1 Timing of identification	FIA_UID.1
FPT_EMS.1	No dependencies	No dependencies
FPT_FLS.1/EoL	No dependencies	No dependencies
FPT_FLS.1/Fail	No dependencies	No dependencies

FPT_ITT.1	No dependencies	No dependencies
FPT_PHP.3	No dependencies	No dependencies
FPT_TST.1	No dependencies	No dependencies
FRU_FLT.2	FPT_FLS.1 Failure with preservation of secure state	FPT_FLS.1/EoL
FTA_SSL.3	No dependencies	No dependencies
FTA_SSL.4	No dependencies	No dependencies
FTP_ITC.1	No dependencies	No dependencies

A.6.3.2 Rationale for security objectives

Table 5: Security objective rationale for the base PP

	O.Identify	**O.AccCtrl**	**O.QKD**	**O.QKDAuth**	**O.Audit**	***O.TST***	***O.EMSec***	***O.Sanitize***	**O.SessionLimit**
FAU_GEN.1					x				
FAU_STG.1					x				
FAU_STG.3					x				
FCS_CKM.4			x	x				x	
FCS_COP.1/Aud					x				
FCS_COP.1/CCI				x					
FCS_QKD.1			x	x					
FCS_RNG.1			x						
FDP_ACC.1		x							
FDP_ACF.1		x		x	x	x			
FDP_DAU.1					x				
FDP_ETC.1		x	x						
FIA_ATD.1	x	x							
FIA_UID.1	x								
FIA_USB.1	x	x							
FMT_MSA.1		x						x	
FMT_MSA.2		x							
FMT_MSA.3		x			x				
FMT_MTD.1		x			x				
FMT_MTD.1/Adm		x							
FMT_MTD.1/QAK		x							
FMT_SMF.1		x							
FMT_SMR.1		x			x				
FPT_EMS.1							x		
FPT_FLS.1/EoL			x			x		x	
FPT_FLS.1/Fail				x		x			
FPT_ITT.1		x	x						
FPT_PHP.3						x	x		
FPT_TST.1						x			
FRU_FLT.2						x			
FTA_SSL.3									x
FTA_SSL.4									x
FTP_ITC.1			x	x					

O.Identify

FIA_ATD.1 requires the TSF to maintain the list of security attributes User Identity, and Role from individual users to enable the identification of users.

FIA_USB.1 requires the TSF to associate each user initially with the unidentified user role, and only after identification associate them with their respective Role.

FIA_UID.1 requires the TSF to deny access to any controlled resources before the user is identified. It also requires to associate each user with a role.

O.AccCtrl

FIA_ATD.1 defines the security attributes of individual users including their Role used for the subset access control Access Control SFP. Access Control SFP is described by the SFR FDP_ACC.1. FDP_ACF.1 defines the access control rules and restricts access to key distribution services, QKD keys, and ADR, based on the identified users, their associated Roles, and the operational state. The requirement to export the QKD keys is defined by FDP_ETC.1.

FMT_MSA.1 defines the operational state and how it may be changed. FIA_USB.1 binds identified users to their Roles including secure initial values. For QKD keys and ADR FMT_MSA.3 defines initial values for security attributes. Initialization of the operational state is not required as this is not bound to any subjects or objects, which may be created.

The capabilities for management of TSF data is defined by FMT_SMF.1.

FMT_MTD.1 defines the management functions of the ADR and CD. It restricts management of ADR to Auditors and access to CD to Maintainers.

FMT_MTD.1/QAK defines the QAK as not manageable, since Personalization state is not an operational state in the base PP.

FMT_MTD.1/Adm defines the user management, management of the UDR and restricts this to the Administrator. The allowed values for the security attribute Role are restricted by FMT_SMR.1.

FMT_MSA.2 ensures that the TSF prohibit the same User Identity to hold the Roles Administrator and Auditor at once.

FMT_MSA.1 allows the Key Requester to specify the authorized users allowed to receive the requested key. FMT_MSA.3 sets the default to the requesting user and FMT_MSA.2 restricts the receivers to Key Requesters. FPT_ITT.1 ensures that the corresponding security attributes cannot be modified when transferred in between the QKD modules.

O.QKD

FCS_QKD.1 supplies the said P&M protocol for quantum key distribution. FTP_ITC.1 implements the required authenticated channel for the classical communication on the QKD link. The details are handled in O.QKDAuth below.

FCS_QKD.1 requires to formally quantify conceptual imperfections of the P&M protocol compared with an ideal key establishment protocol by the security parameter. It keeps track of the life-time count of attempts of key establishment using an attempt counter. Therefore, it tracks the relevant key design figures, which may enter the security proof of any external application using the output of FCS_QKD.1. FCS_QKD.1 maintains an upper limit for the attempt counter and will enter FPT_FLS.1/EoL, if the limit is exceeded. This will enforce that the assumptions of any composed system will be held.

FPT_ITT.1 ensures that any information required beyond the QKD protocol, e.g. partitioning of the bit string for internal use and export as QKD key, is transferred without modification in between the two QKD modules. FCS_RNG.1 defines the physical random number generator as required for the correct and secure operation of FCS_QKD.1.

FCS_CKM.4 is used to delete internally stored QKD keys after export (FDP_ETC.1) or after a defined time-out.

O.QKDAuth

FTP_ITC.1 requires the TSF to provide a communication channel with assured identification of the TOE’s QKD modules and to protect the integrity of the data exchanged through this channel. The authenticity of the exchanged data is based on the fact that the QAK is not known outside the TOE, since it has been securely generated this way by the manufacturer and it is securely updated by the TOE (FCS_QKD.1) during operation.

FCS_COP.1/CCI defines the cryptographic mechanisms using the QKD Authentication Keys and ensuring the authenticity of data exchanged through the classical channel as required by O.QKD.

The initial QAK is pre-installed by the manufacturer. For the update of the QAK FCS_QKD.1 is used, which requires that each QKD transaction requires the regeneration of a new QAK. If no QAK is available at the end of a QKD transaction, FPT_FLS.1/Fail case (3) requires the TSF to change to Failure state, which by FDP_ACF.1 denies any further access to the key distribution services.

A QKD transaction is closed by deleting the current QAK using FCS_CKM.4. FCS_COP.1/CCI has been refined to prevent overuse of the QAK by requiring re-keying or session termination when the QAK has been used too many times or for too long.

Application Note 20: If the QAK is updated or derived using either a more complex or a different approach than using shared, confidential random TSF data of FCS_QKD.1 to establish new QAK, the ST author shall model the update mechanism and show that all necessary security objectives of the QKD Authentication Keys are preserved.

Similarly, the TOE may support running several transactions in parallel using distinct QAK. In this case the ST author shall model at least how the required pool of QAK is managed, how the independence of used random numbers is assured, and how any other physical and logical cross-talk is mitigated.

O.Audit

FAU_GEN.1 requires the TSF to generate audit records of auditable events, including administration, calibration, and use of key distribution services.

FAU_STG.1 and FAU_STG.3 require the TSF to reliably store the audit data to prevent loss of audit records.

FAU_GEN.1 prevents undetected deletion of audit records by generating an audit record about deletion and by providing means to uniquely identify separate events.

FDP_DAU.1 requires the TSF to provide evidence of authenticity and to enable the Auditor to verify the validity of the ADR. FCS_COP.1/Aud supplies the required cryptography for this purpose. In the base PP it is assumed that the relevant key, the ASK, is already installed in the TOE when delivered.

The Auditor is defined by FMT_SMR.1 and FMT_MTD.1 defines how the Auditor may configure the TSF as required by FMT_SMF.1.

FDP_ACF.1 allows the Auditor to export ADR, which by FMT_MTD.1 sets the "exported" security attribute, which in turn allows the Auditor to delete exported entries by FDP_ACF.1. FMT_MSA.3 ensures that freshly generated ADR are not marked as exported i.e., have to be exported before deletion.

O.TST

FPT_TST.1 requires the TSF to monitor its operational parameters, by running a suite of self-tests. If such tests fail, the TSF enter FPT_FLS.1/Fail or FPT_FLS.1/EoL depending whether the detected failure is recoverable or not. In either failure state the security attribute operational state is not QKD state and by FDP_ACF.1 access to both key distribution service*s and* QKD keys is denied.

For monitoring the QKD link FPT_PHP.3 is used to explicitly detect active probing using the QKD link. In case harmful conditions are detected, FPT_FLS.1/Fail or FPT_FLS.1/EoL is chosen as a secure fallback.

FRU_FLT.2 requires the TSF to operate correctly, if FPT_TST.1 does not detect any harmful condition.

O.EMSec

FPT_EMS.1 requires the TSF to limit emanations through the QKD link to a not intelligible level, for any confidential user data or TSF data.

FPT_PHP.3 requires the TSF to react to active probing in order to prevent forced leakage.

O.Sanitize

FPT_FLS.1/EoL requires the TSF to enter an End of Life state, if it cannot ensure the TSF. FCS_CKM.4 is used to delete all confidential data in this state.

FMT_MSA.1 allows anyone to sanitize the TOE from any operational state.

O.SessionLimit

FTA_SSL.4 requires the TSF to allow each user to terminate the own session. FTA_SSL.3 requires the TSF to terminate inactive sessions.

A.7 Packages

A.7.1 Trusted User Interfaces with Authentication

A.7.1.1 Identification

Package Identifier: Trusted user interfaces with authentication (TUI+A)

A.7.1.2 Introduction

The base Protection Profile assumes (A.SecureOp) that the TOE is operated in a secure environment and that only authorized users have access to the user interfaces of the TOE. For in any way scalable installations this is very inconvenient, and it obviously requires that all consumers of a QKD key are also located inside the same secure environment. This will often require additional personnel to enter the room in order to maintain the key consuming equipment connected to the security services of the TOE.

This package defines trusted paths for the user interfaces as an alternative to physical access control. The trusted paths also identify and authenticate users and thus replace OE.Personnel, since impersonation is mitigated technically by the TSF. OE.SecureOp is slightly refined, since the user interfaces may be outside of the secure environment.

If impersonation is the only concern, the Local Authentication of Users package described in clause A.7.4 may be chosen instead. This package is mutually exclusive to clause A.7.4, since both packages address the same security problem by different approaches. However, ST authors are free to add an additional user authentication through the trusted path, when using this package, although, this is not required to support the TSP.

This package refines the TOE overview in the PP introduction, clause A.1.3.

TOE definition

Users connect to the TOE by means of secure terminals, which set up a secure link to the TOE authenticating both end points, i.e., the TOE and the user terminal. The secure link in general will require some cryptographic protocol, which in turn requires secret information stored in the secure terminal or other IT devices attached to it (e.g. chip-cards).

The identity of the remote end point of the trusted path as indicated towards the TOE is considered the user's identity. Authentication is performed using some cryptographic protocol. The user generates Authentication Verification Data (AVD) using some secret for which the user is uniquely accountable for. The TOE contains Authentication Reference Data (ARD) associated with a unique user identity, which can be used to verify that the sender of the AVD is in possession of the accountable secret. Depending on the protocols used for the authentication and encryption of the trusted path the TOE may be required to manage additional cryptographic keys.

The IT device storing and ideally solely processing the secrets for the user authentication by some cryptographic protocol is assumed in the possession of the user. This allows to uniquely map user identities to the identity indicated by the trusted path.

Life-cycle

Since all users have to be authenticated using corresponding ARD, at least the ARD of a single Administrator needs to be present before the TOE can be operational. This ARD shall be pre-defined by the manufacturer during pre-personalization. The user shall change the credentials of any pre-defined accounts before entering the operational use of the TOE. Any data or IT device that is required for the user to generate the corresponding AVD shall be delivered with the TOE. The delivery procedure shall ensure that any confidential data is accountable to an individual user.

NOTE: (Informative) If ARD shall not be pre-defined by the manufacturer consider the package from clause A.7.3.

Non-TOE hardware/software/firmware available to the TOE

The TOE requires secure terminals as end points for the trusted paths, which are associated with authorized users. These end points shall ensure the confidentiality and integrity and verify the authenticity of the exported QKD key. They shall also support the users' method of producing their Authentication Verification Data for authentication and shall not disclose any confidential data to set-up an authenticated link.

A.7.1.3 Security Problem Definition

A.7.1.3.1 Introduction

Assets and TSF data

This package does not define additional assets. The following TSF data are required for this package:

ARD Authentication Reference Data is data stored in the TOE used by the TSF to verify the authenticity of a user, i.e. the end point of the trusted path. The **integrity** of this data shall be protected. Whether or not confidentiality is also required depends on the authentication protocol.

Application Note 21: The ST author shall detail whether **confidentiality** is required for ARD and provide a rationale.

AVD Authentication Verification Data sent by or on behalf of the user to the TSF to prove his identity. There are no protection requirements for AVD.

UTK User Transaction Keys: a set of distinct cryptographic keys, where each key is used exclusively to protect data on the trusted path either against modification or disclosure. The **integrity** of the UTK shall be protected. **Confidentiality** is required for at least some parts of the key set.

Application Note 22: The ST author shall detail for which parts of the UTK **confidentiality** is required and provide a rationale.

Users and subjects

Using this package changes the user communication as defined in Users and subjects in clause A.3.1. Instead of local terminals, users communicate through trusted paths. Users may be human users or IT products consuming QKD keys, which eventually operate on behalf of human users. Throughout this package, the term "remote entities" is used to cover both and point out a potentially indirect communication. Formally, the term is synonymous with "user".

Although there may be several systems in between the human user and the TOE, or human users may have delegated their account to automated devices, this Protection Profile assumes that there is a distinct human user accountable for each transaction. All other IT equipment involved is considered as the terminal.

The package requires another user meta-role, which is not exposed to actual users, i.e., users who may have identified themselves, but are not yet successfully authenticated.

Unauthenticated user is another meta-role without access permissions similar to the unidentified user.

Objects

This package does not define additional user data objects.

Security attributes

This package does not define additional security attributes for subjects or user data objects.

A.7.1.3.2 Threats

This package defines additional threats, which shall be considered and mitigated, because A.SecureOp from the base PP has been dropped. This allows the adversary to tap on the user interfaces.

T.DataCompr **Eavesdropping on data on user interfaces**

An adversary gets knowledge of the QKD key by eavesdropping on data transferred between the TOE and authenticated external entities.

T.DataMani **Generation or manipulation of communication data**

An adversary generates or manipulates data transferred between the TOE and authenticated external entities in order to compromise the integrity of the QKD key.

T.Combine **Analysing and combining information at different interfaces**

An adversary obtains measurable properties from any interface of the TOE and analyses them in order to get knowledge about any confidential asset. The adversary may correlate or combine such data from different interfaces for this purpose.

T.Masqu **Generation or manipulation of data on user interfaces**

An adversary generates or manipulates data on the user interfaces in order to gain unauthorized access to key distribution services of the TOE, or to configure TSF data in order to compromise the TSF.

T.Impersonate **Impersonation of other users**

An authorized user generates or manipulates data on any user interface in order to get access to key distribution services of the TOE or QKD keys as another user.

A.7.1.3.3 Assumptions

A.SecComm **Secure communication**

remote entities support trusted paths with the TOE using cryptographic mechanisms. They ensure that individual users are uniquely accountable for initiating trusted paths with a given identity and for all communication through it. They also ensure that confidential information is not compromised in the TOE’s environment.

Application Note 23: This assumption only requires the user terminal as a required IT device in the environment. It has no effects on the TSF.

The developer shall provide guidance for the user to ensure that the level of protection of the remote entities in their environment matches the attack potential claimed in this PP.

A.7.1.4 Security Objectives

A.7.1.4.1 New objectives for the TOE

O.TPath **Trusted path with user authentication**

For communication between the TSF and remote entities, the TSF provides trusted paths using secure cryptographic mechanisms. The TSF provides authentication functionality for both communication end points of the trusted path (TOE and remote entities) and ensures the confidentiality and integrity of the communication data exchanged with the remote entities through the trusted path. For these purposes, the TSF establishes cryptographic User Transaction Keys (UTK) in a way that the confidentiality and integrity of any secret User Transaction Key is not compromised by eavesdropping on or manipulation of any part of the communication. Each User Transaction Key is used for a limited time and a limited number of operations only.

O.AuthFail **Reaction to failed user authentication**

The TSF shall verify the claimed identity of the user before providing access to any controlled resources. The TSF authenticates remote entities using secure cryptographic mechanisms. The TSF detects and reacts to failed authentication attempts.

A.7.1.4.2 Refined objectives for the TOE

O.EMSec **Emanation Security**

The TSF is designed in order to prevent leakage of any intelligible confidential user data or TSF data through the QKD link and the user interface. This includes leakage induced by any active probing.

Even by correlating or combining information from all available interfaces the TSF does not leak any information that would invalidate the security proof for the chosen QKD protocol.

A.7.1.4.3 New objectives for the environment

OE.SecComm **Protection of communication channel**

remote entities shall support trusted paths with the TOE using cryptographic mechanisms. Each trusted path shall have an identity which is uniquely mapped to a user identity. The trusted path establishment shall require the successful authentication of the accountable user of the trusted path by the remote end point or its environment as a prerequisite.

These remote entities in their respective environment shall not disclose any secret authentication data of any users and shall faithfully receive / present communication from / to the user. Confidential information shall only be disclosed to the authorized user.

OE.AuthData Secrecy and generation of authentication data

The authorized users of the TOE keep the confidential information of their authentication data secret. The generation of this secret data ensures that it cannot be guessed and is sufficiently complex such that it cannot be exhaustively searched during the validity period.

A.7.1.4.4 Refined objectives for the environment

NOTE: (Informative) This package transfers security services from the TOE environment to the TOE itself. Therefore, the corresponding properties of the security objectives for the environment as defined in the base PP shall be provided by the security objectives for the TOE in the context of this package.

OE.SecureOp Secure Operational environment

The TOE shall be stored and operated inside an access controlled area, which ensures that only authorized personnel can physically access the TOE ~~and its user interfaces~~. If access to the TOE by unauthorized personnel cannot be excluded, the TOE shall be removed from operation and all QKD keys created since it was last assured to have been continuously inaccessible to unauthorized personnel shall be considered as compromised. When designing the security perimeter it shall be taken into account that the PP claims high attack potential, i.e. the adversary may be backed by organized crime. Standard commercial warehouse protection shall not be considered as adequate protection.

~~**The security perimeter shall ensure that any emanations of the TOE, e.g. electromagnetic, acoustic, power consumption profiles, cannot be detected outside the access controlled area, except signals or emanations conveyed on the QKD link.**~~

OE.Personnel Trustworthy personnel

Personnel authorized to use the TOE is trustworthy and well trained. They will not intentionally misuse the TSF.-In particular, users ~~**won't identify as other users and**~~ will close sessions, while they do not actively interact with the TOE. ~~**Organizational means shall be in place to mitigate potential misconduct. Sample measures may comprise:**~~

~~**1) assignment of user IDs, which are not obvious to other users and shall be kept confidential by the users,**~~

~~**2) verification of correspondence of the logs for room access and TOE use, i.e. detection of users, who shouldn't have been in the room,**~~

~~**3) security screening of personnel by national security agencies.**~~

~~**While none of these proposals is considered mandatory, any single one of these is neither considered sufficient.**~~

A.7.1.4.5 Rationale for the refinements

O.EMSec

In the base PP only the QKD link is available to the adversary. In this package users may be remote, i.e., the physical user interfaces of the TOE may pass through uncontrolled environment, despite any trusted path protocol executed via these interfaces. The trusted path itself may be analysed by side-channel attacks.

Although the adversary cannot analyse the contents inside the trusted path, side-channel information e.g., about timing and quantity of data exchanged, may be accessible. The adversary may combine data obtained at different interfaces.

OE.SecureOp

It is the purpose of this package to have self-protected user interfaces. The threats T.DataCompr, T.DataMani, and T.Masqu consider an adversary with full access to the user interfaces of the TOE.

OE.Personnel

T.Impersonate consider misleading identification of users as a threat. Therefore, it is not necessary to assume that users will refrain from doing so. However, authentication in general requires secret knowledge where a particular user is accountable to use. The corresponding requirement has been added as OE.AuthData and therefore does not impact OE.Personnel.

A.7.1.4.6 Rationale for security objectives

T.Observe

OE.SecureOp excludes that an adversary has access to the TOE and thus cannot observe the TOE locally, i.e. the adversary is restrained to monitoring or probing the QKD link **or the interfaces to remote entities**. *O.TST* explicitly detects or suppresses active probing signals on the QKD link and stops operation in presence of such signals. O.EMSec requires the TSF to not leak any intelligible information on the QKD link.

T.DataCompr

O.TPath requires the TOE to support trusted paths between TSFs and remote entities to ensure the confidentiality of the communication and thus the transmitted QKD key. It furthermore ensures that the cryptographic keys used cannot be obtained by eavesdropping.

OE.SecComm defines requirements to the IT systems acting as user terminals. Since the trusted path ends inside these terminals, these have to prevent leakage.

T.DataMani

O.TPath requires the TOE to support trusted paths between TSFs and remote entities to ensure the integrity of the communication and thus the transmitted QKD key. The generation or modification of data impacts the transferred data's integrity.

OE.SecComm defines requirements to the IT systems acting as user terminals. Since the trusted path ends inside these terminals, these need to also ensure integrity of the users' communication.

T.Masqu

O.Identify requires the TSF to deny access to key distribution services unless the user identity is verified. O.AuthFail requires that the remote entities are authenticated, and to react on failed attempts to gain unauthorized access.

O.TPath requires the TOE to support trusted paths between TSFs and remote entities to ensure the integrity of the communication and thus any other entity cannot modify the communication of an already authenticated user.

O.SessionLimit requires the TSF to close unused sessions, which might be hijacked or piggybacked by other users or an adversary.

OE.AuthData ensures that the secret data required to verify the claimed identity of the remote entities cannot be known to any other external entity. Therefore, the adversary cannot generate valid user authentication; neither to access the key distribution services, nor to claim any role allowed to configure TSF data.

OE.SecComm ensures that the said secret data does not leak at the external IT devices used by the user to establish the trusted path.

T.Impersonate

O.Identify requires the TSF to deny access to key distribution services unless the identity of the remote entity is verified. In addition, O.AuthFail requires that the remote entities are authenticated, and to react on failed attempts to gain unauthorized access.

OE.AuthData ensures that the secret data required to verify the claimed identity of the remote entity cannot be known to any other entity. Therefore, the user cannot generate valid authentication for a different user.

A.SecComm

This assumption is satisfied immediately by OE.SecComm. OE.AuthData supports this assumption in order to keep the trusted paths accountable to individual users; otherwise these could not be trusted.

A.7.1.5 Security requirements

A.7.1.6 New requirements for the TOE

A.7.1.6.1.1 Trusted Path to remote users

FTP_TRP.1 **Trusted path**

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_TRP.1.1 The TSF shall provide a communication path between itself and *remote*[88] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from *modification* ***and*** *disclosure*[89].

FTP_TRP.1.2 The TSF shall permit *remote* ***entities*** ***~~users~~***[90] to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for *all interactions of authenticated users*[91].

Application Note 24: The TSF may permit the TSF to initiate communication via a trusted path (FTP_TRP.1) already established by remote entities. When using this package, the TSF shall not initiate the establishment of a trusted path.

remote entities are understood as users linked by means of external terminals. It does not exclude proximity of the user to the TOE. ST authors might even integrate the terminals with the TOE. Local users defined as human users interacting directly with the TOE are not supported.

If the trusted path does not provide information theoretical security the security statement of QKD keys transported through this path may be weakened.

FCS_COP.1/TRP **Cryptographic operation**

Hierarchical to: No other components.

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

FCS_COP.1.1 The TSF shall perform ***[selection: data encryption / decryption, data integrity failure detection, data authentication]***[92] in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].

Application Note 25: If the cryptographic operations rely on several cryptographic algorithms, the ST author shall iterate FCS_COP.1/TRP for each algorithm.

FCS_CKM.1/UTK **Cryptographic key generation**

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

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

[88] [selection: *remote, local*]

[89] [selection: *modification, disclosure, [assignment: other types of integrity or confidentiality violation]*]

[90] [selection: *the TSF, local users, remote users*]

[91] [selection: *initial user authentication, [assignment: other services for which trusted path is required]*]

[92] [assignment: *list of cryptographic operations*]

cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].

Application Note 26: The ST author may replace FCS_CKM.1/UTK by FCS_CKM.5/UTK, or any other suitable key generation / establishment function, if it fits the chosen protocol. The UTK pertains to the trusted path implemented by FTP_TRP.1.

FIA_UAU.6 Re-authenticating

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions: *session termination both by the user or automatic, or when the UTK has been used [assignment: conditions for excessive use of the UTK]*[93].

Refinement: If the session has not been terminated the TSF may support re-keying of the UTK. If re-keying is supported, the TSF shall provide an adequate key generation function.

Application Note 27: For "*conditions for excessive use of the UTK*", the ST author shall specify at least thresholds for the maximum number of elementary operations e.g., single message block operations for a symmetric block cipher, performed using a single UTK and a maximum life-time for a single UTK.

A.7.1.6.1.2 User Authentication

FIA_UAU.2 User authentication before any action

Hierarchical to: FIA_UAU.1 Timing of authentication

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.2.1 The TSF shall require each ~~**user**~~ **remote entity** to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

FIA_UAU.3 Unforgeable authentication

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.3.1 The TSF shall [selection: *detect, prevent*] use of authentication data that has been forged by any user of the TSF.

FIA_UAU.3.2 The TSF shall [selection: *detect, prevent*] use of authentication data that has been copied from any other user of the TSF.

FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1 The TSF shall detect when [selection: *[assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]*] unsuccessful authentication attempts occur related to *user authentications*[94].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been [selection: *met, surpassed*], the TSF shall **generate an ADR and** [assignment: *list of actions*].

[93] [assignment: *list of conditions under which re-authentication is required*]

[94] [assignment: *list of authentication events*]

A.7.1.6.2 Refined requirements for the TOE

FPT_PHP.3 **Resistance to physical attack**

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_PHP.3.1 The TSF shall resist *active probing via the QKD link* ***or the user interfaces***[95] to the *internal states of the TSF*[96] by responding automatically such that the SFRs are always enforced.

Refinement: The TSF shall implement appropriate mechanisms to continuously, i.e. at any time during the operational life-cycle phase, counter active probing via the QKD link **or the user interface**. As response entering FPT_FLS.1/Fail or FPT_FLS.1/EoL shall be chosen as appropriate.

FPT_EMS.1 **Emanation of TSF and user data**

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_EMS.1.1 The TSF shall ensure that the TOE does not emit emissions over its attack surface in such amount that these emissions enable access to TSF data and user data as specified in the following table:

Table 6: Definition of Side-Channel Protection

ID	Emanation	Attack Surface	TSF data	User Data
1	Timing of signals	QKD link **and user interface**	any confidential TSF data	any confidential user data
2	Signal strength, waveform, or quantum state	QKD link **and user interface**	any confidential TSF data	any confidential user data

A.7.1.6.3 SFR Dependency rationale

Table 7: SFR Dependency rationale

SFR	Dependency resolution
FCS_COP.1/TRP	FCS_CKM.1/UTK generates the UTK FCS_CKM.4 may delete the UTK; otherwise, ST/PP authors shall iterate FCS_CKM.4, if a different method is used for UTK
FCS_CKM.1/UTK	FCS_COP.1/TRP uses the UTK FCS_CKM.4 may delete the UTK; otherwise, ST/PP authors shall iterate FCS_CKM.4, if a different method is used for UTK
FIA_AFL.1	FIA_UAU.2 is hierarchical to FIA_UAU.1
FIA_UAU.2	FIA_UID.1 provides user identification in the base PP
FIA_UAU.3	No dependencies
FIA_UAU.6	No dependencies
FTP_TRP.1	No dependencies

[95] [assignment: *physical tampering scenarios*]

[96] [assignment: *list of TSF devices/elements*]

A.7.1.6.4 Rationale for the security requirements

Table 8: Rationale for the security requirements

	O.EMSec	O.TPath	O.AuthFail
FCS_COP.1/TRP		×	
FCS_CKM.1/UTK		×	
FCS_CKM.4		×	
FIA_AFL.1			×
FIA_UAU.2			×
FIA_UAU.3			×
FIA_UAU.6		×	
FPT_EMS.1	×		
FPT_PHP.3	×		
FTP_TRP.1		×	

O.EMSec

FPT_EMS.1 *requires the TSF to limit emanations through the* QKD link *and the user interface to a not intelligible level, for any confidential user data or TSF data*.

FPT_PHP.3 requires the TSF to react to active probing in order to prevent forced leakage.

O.TPath

FTP_TRP.1 requires the TSF to support a trusted path to local or remote users with assured identification of its end points and protection of data from modification and disclosure. FCS_COP.1/TRP supplies the required cryptographic procedures for data encryption / decryption, data integrity failure detection and data authentication using the UTK. The latter is established using FCS_CKM.1/UTK. After termination of the trusted path FCS_CKM.4 is used to delete the UTK.

FIA_UAU.6 requires the TSF to re-authenticate and thus terminate the session, if the current UTK has been used for excessive operations or for an excessively long period of time.

Application Note 28: It is assumed that the UTK cannot be established, unless the user is authenticated successfully. The AVD is considered an input parameter to FCS_CKM.1/UTK or its surrogate.

O.AuthFail

FIA_UAU.2 requires that identified users need to be authenticated successfully before any other TSF mediated action. This includes the trusted path (O.TPath). FIA_UAU.3 requires a secure authentication protocol i.e., any static transmission of AVD is not considered adequate. FIA_AFL.1 requires reaction to failed authentication attempts.

A.7.2 TOE self-protection

A.7.2.1 Identification

Package Identifier: TOE self-protection (PROT)

A.7.2.2 Introduction

The base Protection Profile assumes (A.SecureOp) that the TOE is operated in a secure environment. A simple reason among others is that an attacker may simply penetrate the TOE and obtain sensitive information about its state. A.SecureOp requires that the attacker cannot approach the device to perform this attack or that the device is taken out of service, if access by an attacker cannot be excluded.

While a secure environment according to A.SecureOp at the first glance sounds like a building with fence and a locked door, this Protection Profile claims resistance to high attack potential. The level of perimeter security may be thought of in terms of bank vaults or depots of nuclear material. It may involve alarm systems, thick walls and guards reaching a potential breaching attempt sooner than it can possibly succeed. Please see the minimum site security requirements [i.4] for further reference concerning aspects and processes to consider.

In order to reduce this costly infrastructure the TOE may be equipped with sufficient self-protection. The corresponding security problem and requirements are the subject of this package.

According to table 1 A.SecureOp is reflected by OE.SecureOp and OE.Personnel. These objectives for the environment however support O.Identify, by allowing that only authorized personnel will have access to the user interfaces of the TOE and requiring that users will not impersonate other users.

This Protection Profile does not mandate storage encryption and storage integrity protection as dedicated SFR. This security functionality is often required for devices used in security applications. It is recommended that ST authors add respective SFR to meet such requirements.

Application Note 29: If this package is chosen, the ST author shall also choose a package for user authentication, e.g. clause A.7.1 Trusted User Interfaces with Authentication or clause A.7.4 Local Authentication of Users, to provide the security functionality required by OSP.Audit and OSP.QKDService.

A.7.2.3 Security Problem Definition

A.7.2.3.1 Introduction

Assets and TSF data

This package does not define additional assets or TSF data.

Users and subjects

This package does not refine users or subjects.

Objects

This package does not define additional user data objects.

Security attributes

This package does not define additional security attributes for subjects or user data objects.

A.7.2.3.2 Threats

T.PhysAttack **Physical attacks**

An adversary obtains intelligence on the internal state of the TSF or modifies the TSF such that the confidentiality of the QKD key is compromised or the adversary gains unauthorized access to the key distribution services of the TOE by

a) physical probing or manipulation of the TOE,

b) applying environmental stress to the TOE, or

c) exploiting information leakage from the TOE.

Application Note 30 Attacks or cross-talk, which may induce or expose a bias, prefer bit patterns or similarly affect the statistics of the QKD key, including correlations to any previously generated QKD keys or correlations to results of other QKD links or transactions, shall be considered as compromising the confidentiality.

Type (a) attacks are invasive or use local interfaces. Attacks involving the QKD link are already covered by T.Observe in the base section of this PP.

Type (b) attacks aim at forcing malfunctions of the TSF.

Type (c) attacks may be combined with type (a) and (b) to force such leakage.

A.7.2.3.3 Assumptions

A.SecureOp

~~**The TOE is installed and operated at a secure area, i.e. only authorized personnel can obtain physical access to the TOE. This**~~ **The** authorized personnel will not intentionally misuse the TOE. ~~**The environment will detect any unauthorized access and the TOE will be taken out of service upon such detection.**~~

A.7.2.4 Security Objectives

A.7.2.4.1 New objectives for the TOE

O.PhysProt **Physical protection**

The TSF detects any attempt for physical probing or manipulation which may compromise the TSF or QKD keys both stored and during establishment, and denies any key distribution service unless the TSF are ensured. If the TSF cannot be ensured or the End of Life state is reached, all confidential data is either deleted or made inaccessible in a secure and persistent way, if not possible to delete.

A.7.2.4.2 Refined objectives for the TOE

O.EMSec **Emanation Security**

The TSF is designed in order to prevent leakage of any intelligible confidential user data or TSF data ~~**through the QKD link**~~ **outside of the TOE boundary, including the QKD link**. This includes leakage induced by any active probing.

A.7.2.4.3 Refined objectives for the environment

NOTE: (Informative) This package transfers security services from the TOE environment to the TOE itself. Therefore, the corresponding properties of the security objectives for the environment as defined in the base PP shall be provided by the security objectives for the TOE in the context of this package.

OE.SecureOp **Secure Operational environment**

This objective is dropped for this package.

A.7.2.4.4 Rationale for the refinements

O.EMSec

In the base PP OE.SecureOp requires that the adversary cannot gain local access to the TOE. Therefore, the adversary only has access to the QKD link. By dropping A.SecureOp OE.SecureOp cannot be claimed and the adversary gains local access to the TOE and can thus monitor data at the entire TOE boundary. With this refinement T.Observe is still mitigated.

OE.SecureOp

OE.SecureOp requires that the TOE is stored and operated inside an access controlled area. This package is however intended to remove this limitation by adequate self-protection. According to table 1 OE.SecureOp is interdependent with the following items:

T.ExplMal requires OE.SecureOp to restrain the adversary from locally inducing malfunctions. T.PhysAttack type (b) explicitly requires the TSF to mitigate this scenario.

T.Observe is mitigated using the refinement to O.EMSec.

OSP.QKDService uses OE.SecureOp to uphold user identification. This package requires to include a package for user authentication, which solves these requirements by technical means.

OSP.Audit uses OE.SecureOp to uphold user identification. This package requires to include a package for user authentication, which solves these requirements by technical means.

A.SecureOp has been refined in this package to avoid conflicts.

A.7.2.4.5 Rationale for the security objectives

T.PhysAttack

O.PhysProt counters type (a) attacks by requiring the TSF to detect any attempt for physical probing or manipulation that may compromise the TSF or QKD keys. O.TST counters type (b) attacks by denying access to the key distribution services and QKD keys unless the TSF are ensured. If the TSF cannot by assured, O.PhysProt makes the key distribution services and QKD keys permanently inaccessible. The refined O.EMSec requires the TSF to not leak any intelligible information outside the TOE boundary, thus mitigating type (c) attacks.

A.SecureOp

This package supplies security functions for the TOE to protect itself in the presence of an adversary with local access to the TOE. The environment cannot detect any unauthorized access, which eventually results in dropping OE.SecureOp. A.SecureOp is therefore reduced to the assumption that authorized users won’t misuse the TSF, which is reflected by OE.Personnel. Obviously, an adversary could easily impersonate an authorized user, unless an appropriate user authentication package is also chosen as required by this package.

A.7.2.5 Security requirements

A.7.2.5.1 Introduction

As clarified in Application Note 29 this package also requires user authentication. The SFRs for user identification are not defined in this clause and have to be defined by the ST author. If a pre-defined user authentication package is used, i.e. one of clause A.7.1 or A.7.4, the SFRs defined there shall be added.

A.7.2.5.2 New requirements for the TOE

FPT_PHP.3/MOD **Resistance to physical attack**

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_PHP.3.1 The TSF shall resist *attempts for physical probing or manipulation of the TOE*[97] to the *TSF*[98] by responding automatically such that the SFRs are always enforced.

Refinement: **The TSF will implement appropriate mechanisms to continuously counter physical manipulation and physical probing. Due to the nature of these attacks (especially manipulation) the TSF can by no means detect attacks on all of its elements. Therefore, permanent protection against these attacks is required ensuring that security functional requirements are enforced. Hence, "automatic response" means here**

(i) assuming that there might be an attack at any time and

(ii) countermeasures are provided at any time.

If the TSF cannot be enforced otherwise, the End of Life state shall be entered.

A.7.2.5.3 Refined requirements for the TOE

FPT_EMS.1 **Emanation of TSF and user data (refined from base PP)**

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_EMS.1.1 The TSF shall ensure that the TOE does not emit emissions over its attack surface in such amount that these emissions enable access to TSF data and user data as specified in the following table:

[97] [assignment: *physical tampering scenarios*]

[98] [assignment: *list of TSF devices/elements*]

Table 9: Definition of Side-Channel Protection

ID	Emanation	Attack Surface	TSF data	User Data
1	Timing of signals	QKD link	any confidential TSF data	any confidential user data
2	Signal strength, waveform, or quantum state	QKD link	any confidential TSF data	any confidential user data
3	**Power consumption**	**QKD module and user interfaces**	**any confidential TSF data**	**any confidential user data**
4	**Electromagnetic emission**	**QKD module and user interfaces**	**any confidential TSF data**	**any confidential user data**
5	**Acoustic emission**	**QKD module and user interfaces**	**any confidential TSF data**	**any confidential user data**

A.7.2.5.4 SFR Dependency Rationale

Table 10: SFR Dependency Rationale

SFR	Dependency resolution
FPT_PHP.3/MOD	No dependencies

A.7.2.5.5 Rationale for the Security Requirements

Table 11: Rationale for the Security Requirements

	O.PhysProt	O.EMSec
FPT_EMS.1		×
FPT_FLS.1/EoL	×	
FPT_PHP.3	×	×
FPT_PHP.3/MOD	×	×

O.PhysProt

FPT_PHP.3/MOD detects any attempts to physically probe or manipulate the TSF locally on either QKD module. FPT_PHP.3 from the base PP covers the QKD link, i.e. the entire attack surface of the TOE is covered. FPT_FLS.1/EoL supplies the fail-safe state to assume, when an attack is detected, which cannot be countered otherwise. This state already requires the deletion of all confidential data.

O.EMSec

FPT_PHP.3 requires the TSF to react to active probing on the QKD link in order to prevent forced leakage. FPT_PHP.3/MOD prevents active probing on the QKD modules, themselves.

The refined FPT_EMS.1 requires the TSF to limit emanations through both the QKD link and the TOE boundary of the QKD modules to a not intelligible level, for any confidential user data or TSF data.

A.7.3 Provisioning after delivery

A.7.3.1 Identification

Package Identifier: Provisioning and Re-Personalization after delivery (PERSO)

A.7.3.2 Introduction

The base PP assumes that the TOE is delivered with full trust provisioning performed by the manufacturer. Since this puts a lot of trust into the manufacturer, this may not be desirable by customers. It will neither allow replacements of single QKD modules and may have many more drawbacks for any given business model or security policy.

This package aims at the other extreme for the pre-operational phase. All pre-operational tasks are performed after delivery. The TOE contains a manufacturer ASK for the recipient to verify that the TOE is pristine. For ALC_DEL evaluators shall verify that delivery processes enforce the chain of trust e.g., by using trusted and accountable couriers for the TOE and a separate and authentic channel for conveying some verification token for the ASK.

This package does not provision the TOE with any pre-defined credentials for an initial Administrator account. It is recommended to augment this package by such a pre-defined account with credentials that have to be changed at the first use and are unique per TOE.

Life-cycle

Since trust provisioning is left with the user in this package the pre-personalization (see figure 4) is empty. Instead the provisioning is performed in Personalization state after delivery.

Personalization state:

In Personalization state an Administrator receives the QKD modules in a secure environment. The Administrator verifies that both QKD modules and the manufacturer's ASK verification token e.g., public key of the ASK, have undergone a trusted delivery, that the audit data logs are clean and properly signed by the manufacturer's ASK, and then performs trust provisioning by:

1) creation of the initial Administrator account with adequate credentials,

2) pairing the QKD modules to form a QKD system. This is achieved by requesting the TSF to agree on a new QAK[99],

3) optionally, create or import the user's ASK,

4) optionally, import further TSF data. E.g. if the package from clause A.7.1 was also chosen, import Authentication Reference Data (ARD)**.**

Once the trust provisioning is finalized, the QKD system may be installed into its intended environment. Note that even if the self-protection package from clause A.7.2 has been chosen the secure environment is required for the Personalization state. However, that package may facilitate a less restrictive transport of the QKD modules to their final destination.

An Administrator may return a failed QKD system to the secure environment in order to repeat the personalization, e.g. when the QAK went out of synchronization.

Application Note 31: Regenerating QAK using an uncontrolled QKD link is explicitly prohibited.

NOTE: (Informative) Developers may consider using more than one QAK and switch to a fresh QAK in case of lost synchronization. The TOE may use the TSF to create new QAK for future use while there are still valid QAK available. This is not modelled in this package and would have to be defined by the ST author.

[99] While it would also be acceptable to inject QAK into both modules, this would require an external, secure random number generator. This would furthermore require additional security functionality to ensure secure import of the QAK.

A.7.3.3 Security Problem Definition

A.7.3.3.1 Introduction

Assets and TSF data

This package does not define additional assets or TSF data.

ST authors may handle the manufacturer's ASK as an asset separate from the user's ASK.

Users and subjects

This package defines the Initializer as a new role. The Initializer is only available during Personalization state, and if there is no Administrator UDR defined. There are no credentials associated with the Initializer account. It is used to perform the initial personalization which includes the definition of the first Administrator UDR. Once an Administrator UDR is defined, the Initializer is no longer available.

Objects

This package does not define additional user data objects.

Security attributes

This package does not define additional security attributes for subjects or user data objects.

However, when using this package the TOE is delivered without an UDR for an Administrator.

A.7.3.3.2 Threats

T.Inititalize **Compromised initialization of TSF data**

An adversary may modify, replace or eavesdrop on the initialization of TSF data during Personalization state and use this information during QKD state to

a) exploit knowledge of the QAK to modify data on the QKD link in order to compromise the QKD key without detection by the TSF,

b) exploit knowledge of ARD, if applicable, to authenticate as an authorized user and access the key distribution service, read established QKD keys, or compromise the TSF by assuming Maintainer and Auditor roles, or

c) inject ARD, if applicable, to authenticate as an authorized user and access the key distribution service or compromise the TSF by assuming Maintainer and Auditor roles.

Application Note 32: The threat type (a) applies to the base PP and all packages defined in this document. Types (b) and (c) only apply, if a package was chosen, which defines ARD as TSF data.

If the ST author defines additional TSF data, which are initialized during Personalization state, the ST author shall also refine this threat accordingly.

A.7.3.3.3 Assumptions

A.SecureOp

The TOE is installed and operated at a secure area, i.e. only authorized personnel can obtain physical access to the TOE. This authorized personnel will not intentionally misuse the TOE. The environment will detect any unauthorized access and the TOE will be taken out of service upon such detection.

Personalization of the TOE occurs in a secure environment by trusted personnel. Initial credentials are of adequate quality.

Application Note 33: This refinement can be combined with the refinement defined in the self-protection package from clause A.7.2.

A.7.3.4 Security Objectives

A.7.3.4.1 New objectives for the TOE

O.Personalization Access control to personalization

The TSF maintain a Personalization state, which allows to initialize TSF data: QAK, ASK, and, if applicable, ARD for an initial Administrator. In this state the key distribution service is not available and no QKD keys can be established. To enter this state the TSF either

a) enforce that all TSF data, which can be initialized in Personalization state, is cleared along with all information about QKD keys, which may have been established previously or are still in the establishing phase, or

b) if user authentication is supported, require clearance by at least two authenticated Administrators for re-personalization.

The TSF require local, physical access for the initial Administrator to both QKD modules to initialize the TSF data.

The initialization of the QAK is performed by the TSF on request of the initial Administrator. It is only available in Personalization state. The TSF ensure an adequate quality of the established initial QAK.

O.Pristine Proof of intactness after initial delivery

The TSF allows to read audit data before initial personalization and signs exported logs with the manufacturer loaded ASK.

A.7.3.4.2 New objectives for the environment

NOTE: (Informative) This package transfers security services from the TOE developer to the TOE itself and its environment.

OE.Initialize Secure environment for initialization

Initialization shall occur in a secure environment, where both QKD modules and the QKD link are under the control of the initial Administrators. Physical access control shall ensure that any person potentially able to monitor, eavesdrop, or modify data at any interface of the TOE is known and trusted.

Before the first start the Initializer shall verify that the TOE has been delivered using a trusted and accountable courier, that any delivery notices pertain to the actual TOE instance e.g., by checking model name and serial number, and that an ASK verification token for the TOE instance has been securely delivered.

For the first personalization the Initializer shall verify that the audit logs are properly signed by the manufacturer’s ASK. The logs shall be examined for any evidence of any ADR deleted previously, or for any previous personalization activities. If previous personalization activities cannot be excluded by the Initializer, the TOE shall be rejected.

A.7.3.4.3 Rationale for the refinements

A.SecureOp

This assumption is extended to the Personalization state, which was before delivery in the base PP. Even if the requirement for a secure environment during operation has been dropped by the self-protection package from clause A.7.2, this refinement adds the secure environment for the Personalization state.

A.7.3.4.4 Rationale for security objectives

T.Inititalize

O.Personalization defines the Personalization state as a well-defined state, which is clearly separate from all operational states. OE.Initialize requires the Personalization state to occur in a controlled environment without access for any adversary. This organizational requirement is supported by O.Personalization requiring simultaneous local access to both modules, which discourages initialization over uncontrolled QKD links. It furthermore requires the adversary to have such access while trying to enter Personalization state without authorization.

If no package with user authentication is chosen, OE.SecureOp will prohibit local access to the TOE.

Otherwise, as O.Personalization option (a) requires to clear all TSF data including any ARD the TSF will deny the key distribution service to the legitimate users due to missing credentials. This provides evidence of such a manipulation and prohibits leakage of established QKD keys.

O.Personalization option (b) is only possible, if authenticated by at least two Administrators. In this case, OE.AuthData ensures that the adversary cannot misuse this option. OE.AuthData also ensures that any initial ARD are of adequate quality.

O.Pristine allows the Initializer to verify that the TOE has not been tampered with before it was received at the secure environment for initial personalization. OE.Initialize requires the Initializer to perform this verification.

A.SecureOp

OE.Initialize requires the Personalization state to occur in a controlled environment without access for any adversary. If applicable, OE.AuthData ensures that any initial ARD are of adequate quality.

This assumption is extended to the Personalization state, which was before delivery in the base PP. Even if the requirement for a secure environment during operation has been dropped by the self-protection package from clause A.7.2, this refinement adds the secure environment for the Personalization state.

A.7.3.5 Security requirements

A.7.3.5.1 New requirements for the TOE

FDP_RIP.4 Sanitizing on State Change

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RIP.4.1 The TSF shall ensure that any previous information content about QAK, QKD keys, internal states of FCS_QKD.1, [assignment: *data to be initialized in Personalization state, other confidential data*][100] is made unavailable upon *changing the operational state to Personalization state*[101].

A.7.3.5.2 Refined requirements for the TOE

FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1 The TSF shall enforce the *Access Control SFP*[102] to restrict the ability to *modify*[103] the security attributes *operational state*[104] ~~to~~ ***according to the following list:***

(1) the Maintainer role may set Calibration state from any operational state except End of Life,

(2) the Maintainer role may set QKD state from Calibration state,

(3) any role may set End of Life from any operational state,

(4) simultaneous local interaction, e.g. pressing a button on both QKD modules, of any role including unidentified users on both QKD modules in Failure state may set

[100] [assignment: *list of assets, user data, TSF data*]

[101] [assignment: *list of events detected by the TSF*]

[102] [assignment: *access control SFP(s), information flow control SFP(s)*]

[103] [selection: *change_default, query, modify, delete,* [assignment: *other operations*]]

[104] [assignment: *list of security attributes*]

Personalization state. If user authentication is supported, two identified users with Administrator role may be required to jointly authorize this step.

(5) simultaneous local interaction on both QKD modules in Personalization state may set Calibration state.[105]

Application Note 34: The TOE shall maintain a state-machine for operational states as proposed in clause A.1.3, Life-cycle. For the base PP this state-machine consists of: Calibration state, QKD state, Failure state, and End of Life. **This package adds the Personalization state, also included in figure 4.** The ST author shall refine FMT_MSA.1, if more operational states are supported. Changing the operational state to Failure state is performed by the TSF, e.g. FPT_TST.1.

FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

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 not specified[106] level of audit; and

c) start-up after power-up,

d) creation and deletion of User Definition Records (cf. FMT_MTD.1/Adm (1))

e) modification of the user security attribute Role *(cf. FMT_MTD.1/Adm (2))*

f) Failure with preservation of secure state (cf. FPT_FLS.1/Fail): entering and exiting secure state,

g) deletion and export of audit records (cf. FMT_MTD.1 (2), FDP_ACF.1)

h) selection, de-selection and clearance of events causing audit events (cf. FMT_MTD.1 (3))

i) changes with respect to possible audit storage failure (cf. FAU_STG.3)

j) requests and changes of calibration data (cf. FMT_MTD.1 (1)),

k) shifts in operational state, and recording the user's identity initiating the shift, for manual state shifts,

l) access to the key distribution services,

m) ***all TSF initialization events performed in Personalization state,***

n) [assignment: additional specifically defined auditable events][107].

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

a) ~~Date and time of the event~~**[assignment:** ***information required to uniquely identify separate events and ensure their completeness and chronological order*****],** type of event, subject identity (if applicable), 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*].

NOTE: (Informative) As compared to the base PP item m) has been added for this package.

[105] [assignment: *the authorized identified roles*]

[106] [*selection: choose one of: minimum, basic, detailed, not specified*]

[107] [assignment: *other specifically defined auditable events*]

FMT_MTD.1/Adm Management of TSF data – Administrator

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1 The TSF shall restrict the ability to

(1) *create and delete*[108] the *User Definition Records of an identified user*[109] to *Administrator*[110],

(2) *modify*[111] the *Role of an identified user*[112] to *Administrator*[113]**,**

(3) *change_default*[114] the *Role of an identified user*[115] to *none*[116]*,*

(4) ***create*[117] the *first UDR for an initial Administrator*[118] to *Initializer*[119].**

FMT_MTD.1/QAK Management of TSF data

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1 The TSF shall restrict the ability to ~~***establish,***~~

(1) *query, modify*[120] the *QAK*[121] to *none*[122]*,*

(2) ***establish*[123] the *QAK*[124] to *Administrator*[125] while in Personalization state**.

[108] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]

[109] [assignment: *list of TSF data*]

[110] [assignment: *the authorized identified roles*]

[111] [selection: *change_default, query, modify, delete, clear,[assignment: other operations]*]

[112] [assignment: *list of TSF data*]

[113] [assignment: *the authorized identified roles*]

[114] [selection: *change_default, query, modify, delete, clear,[assignment: other operations]*]

[115] [assignment: *list of TSF data*]

[116] [assignment: *the authorized identified roles*]

[117] [selection: *change_default, query, modify, delete, clear,[assignment: other operations]*]

[118] [assignment: *list of TSF data*]

[119] [assignment: *the authorized identified roles*]

[120] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]

[121] [assignment: *list of TSF data*]

[122] assignment: *the authorized identified roles*

[123] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]

[124] [assignment: *list of TSF data*]

[125] assignment: *the authorized identified roles*

Application Note 35: The refinement has been chosen to avoid iteration of the component. The ST author shall model how the QAK is established. A simple approach would be using FCS_RNG.1. Since the exchange happens in a controlled environment, the FPT_ITT family may not be required.

FDP_ACF.1 Security attribute based access control - Access Control SFP

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1 The TSF shall enforce the *Access Control SFP*[126] to objects based on the following:

(1) subjects: identified users (attribute: *Role)**, Initializer**,*

(2) objects: QKD keys (attributes: receiver, owner), key distribution services (attribute: operational state), ADR (attribute: exported)[127].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

(1) identified users with Role Key Requester are allowed to export QKD keys, if the receiver attribute of the QKD key matches the user identity

(2) identified users with Role Key Requester are allowed to access the key distribution services to request establishment of QKD keys,

(3) identified users with Role Auditor are allowed to export and delete ADR,

(4) [assignment: additional rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects][128].

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules:

(1) the Initializer i.e., the unidentified user logged on before any user has been created, is allowed to export ADR while the operational state is Personalization state.

[assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

(1) Neither the key distribution services nor any QKD keys shall be accessed, unless the operational state is QKD state,

(2) ADR shall not be deleted unless the attribute "exported" is true and the identified user has the Role Auditor,

(3) [assignment: additional rules, based on security attributes, that explicitly deny access of subjects to objects][129].

[126] [assignment: *access control SFP*]

[127] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

[128] [assignment: *additional rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

[129] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

FMT_SMR.1 **Security roles**

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1 The TSF shall maintain the roles: *Unidentified User, Identified User, Administrator, Auditor, Maintainer, Key Requester***, *Initializer***, *[selection: [assignment: other roles], no other roles]*[130].

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

Application Note 36: The Initializer is defined as an unidentified user during Personalization state, while no UDR exists in the TOE.

A.7.3.5.3 SFR Dependency Rationale

Table 12: SFR Dependency Rationale

SFR	Dependency resolution
FDP_RIP.4	No dependencies

A.7.3.5.4 Rationale for the Security Requirements

Table 13: Rationale for the Security Requirements

	O.Personalization	O.Pristine
FAU_GEN.1		×
FAU_STG.1		×
FAU_STG.3		×
FCS_RNG.1	×	
FDP_ACF.1	×	×
FDP_DAU.1		×
FDP_RIP.4	×	
FMT_MSA.1	×	
FMT_MTD.1/Adm	x	
FMT_MTD.1/QAK	×	
FMT_SMR.1	×	×

O.Personalization

FMT_MSA.1 defines the Personalization state and how it can be entered and exited. It requires local access to both QKD modules. According to FDP_ACF.1 key distribution service and QKD keys are only available in operational state, i.e. not in Personalization state. FDP_RIP.4 ensures that all data, which can be initialized in Personalization state and any pre-existing QKD keys are deleted when Personalization state is entered.

FMT_MSA.1 requires local access of the users initiating Personalization state. If user authentication is supported FMT_MSA.1 requires clearance by two Administrators.

FMT_MTD.1/QAK was refined to allow for establishing of QAK by Administrators. FCS_RNG.1 is used to generate a new QAK, which is agreed upon by the two QKD modules using the classical channel in plain text without authenticity

[130] [assignment: *authorized identified roles*]

requirements. This is adequately secure since OE.Initialize requires a secure environment for Personalization state. FCS_RNG.1 also ensures that the established QAK have a well-defined entropy.

FMT_MTD.1/Adm allows the Initializer to create the first Administrator user. FMT_SMR.1 defines the Initializer role.

O.Pristine

FDP_ACF.1 allows the Initializer to read ADR. FDP_DAU.1 will provide the proof of origin for exported ADR. FAU_STG.1 and FAU_STG.3 ensure that the audit data cannot be compromised. FAU_GEN.1 requires to log all activities during Personalization state to produce evidence for the Initializer that the TOE has not been tampered with. The creation of an Auditor user, who might delete audit data, would be logged and FAU_GEN.1 requires to log audit data deletion. Thus any previous personalization activities yield evidence.

FMT_SMR.1 defines the Initializer role.

A.7.4 Local Authentication of Users

A.7.4.1 Identification

Package Identifier: Authentication of local users (LUA)

A.7.4.2 Introduction

The base PP assumes (A.SecureOp) that the TOE is operated in a secure environment and that only authorized users have access to the user interfaces of the TOE. The package defined in clause A.7.1 allows for remote access of users, or access involving some external IT equipment even if used locally. This package is about local user authentication, i.e., users authenticate their identity while physically interacting with the TOE.

This package is mutually exclusive with clause A.7.1, i.e., these packages contain incompatible refinements and definitions. If the TOE shall support both, the ST author may use these as a starting point to model the corresponding security services of the TOE. This package can however be combined with clause A.7.2.

TOE definition

The TOE features user interfaces, which can be operated by a human user directly.

The user claims an identity on this interface and provides Authentication Verification Data (AVD) to prove this identity. The users shall be accountable for producing their AVD by using unique knowledge, unique things in his possession or unique intrinsic properties, e.g. it could be a secret password or biometrical data about the user. The TOE contains Authentication Reference Data (ARD) associated with a unique user identity, which can be used to verify that the sender of the AVD is in possession of the accountable secret.

Life-cycle

Since all users have to be authenticated using corresponding ARD, at least the ARD of a single Administrator needs to exist before the TOE can be operational. This ARD is pre-defined by the manufacturer during pre-personalization. Whatever data or IT device is required for the user to generate the appropriate AVD shall be delivered with the TOE. Delivery shall ensure that any confidential data is accountable to an individual user.

NOTE: (Informative) If ARD shall not be pre-defined by the manufacturer consider the package from clause A.7.3.

A.7.4.3 Security Problem Definition

A.7.4.3.1 Introduction

Assets and TSF data

This package does not define additional assets. The following TSF data are required for this package:

ARD Authentication Reference Data is data stored in the TOE used by the TSF to verify the authenticity of a user, i.e., the end point of the trusted path. The **integrity** and **confidentiality** of this data shall be protected.

AVD Authentication Verification Data sent by or on behalf of the user to the TSF to prove that user's identity. There are no protection requirements for AVD.

Users and subjects

The package requires another user meta-role, which is not exposed to actual users. Since users may have identified themselves, but not yet successfully authenticated

Unauthenticated user is another meta-role without access permissions similar to the unidentified user.

Objects

This package does not define additional user data objects.

Security attributes

This package does not define additional security attributes for subjects or user data objects.

A.7.4.3.2 Threats

T.Masqu **Generation or manipulation of data on user interfaces**

An **adversary** generates or manipulates data on any user interface in order to gain unauthorized access to key distribution services of the TOE, or to configure TSF data in order to compromise the TSF.

T.Impersonate **Impersonation of other users**

An authorized user generates or manipulates data on any user interface in order to get access to key distribution services of the TOE or QKD keys as another user.

A.7.4.3.3 Assumptions

A.AuthData **Secure authentication credentials**

Authentication credentials are known to unique users, and users will protect their credentials from disclosure.

Application Note 37: This assumption is about the quality of user credentials. Since the base PP does not support user authentication, it does not affect the security services stated in the base PP.

A.7.4.4 Security Objectives

A.7.4.4.1 New security objectives for the TOE

O.I&A ***Identification* and authentication of users**

The TSF shall uniquely identify users and verify the claimed identity of the user before providing access to any controlled resources. The TSF reject weak credentials. The TSF detects and reacts to failed authentication attempts.

A.7.4.4.2 New objectives for the environment

OE.AuthDataUI ***Secrecy* and generation of authentication data**

The authorized users of the TOE keep the confidential information of their authentication data secret. The generation of this secret data ensures that it cannot be guessed and is sufficiently complex such that it cannot be exhaustively searched during the validity period.

The entry of the authentication on the user interfaces of the TOE shall not be observable by other people.

A.7.4.4.3 Rationale for security objectives

T.Masqu

O.Identify requires the TSF to deny access to key distribution services unless the user identity is verified. O.I&A requires that the user is authenticated, and to react on failed attempts to gain unauthorized access.

O.SessionLimit requires the TSF to close unused sessions, which might be hijacked or piggybacked by other users or an adversary.

OE.AuthDataUI ensures that the secret data required to verify the claimed identity of the user cannot be known to any other entity. Therefore, the adversary cannot generate valid user authentication; neither to access the key distribution services, nor to claim any role allowed to configure TSF data.

Finally, O.I&A rejects weak credentials as a second layer of assurance, if the original generation of credentials by OE.AuthDataUI may have missed the intended strength.

T.Impersonate

O.Identify requires the TSF to deny access to key distribution services unless the user identity is verified. O.I&A requires that the user is authenticated, and to react on failed attempts to gain unauthorized access.

OE.AuthDataUI ensures that the secret data required to verify the claimed identity of the user cannot be known to any other entity. Therefore, the user cannot generate valid authentication for a different user.

Finally, O.I&A rejects weak credentials as a second layer of assurance, if the original generation of credentials by OE.AuthDataUI may have missed the intended strength.

A.AuthData

OE.AuthDataUI immediately maps this assumption to management of individual secrets.

A.7.4.5 Security requirements

A.7.4.6 New requirements for the TOE

A.7.4.6.1.1 User Authentication

FIA_UAU.2/LUA **User authentication before any action – Local user authentication**

Hierarchical to: FIA_UAU.1 Timing of authentication

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

FIA_AFL.1/LUA **Authentication failure handling – Local user authentication**

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1 The TSF shall detect when [selection: *[assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]*] unsuccessful authentication attempts occur related to *user authentications*[131].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been [selection: *met, surpassed*], the TSF shall [assignment: *list of actions*].

FIA_SOS.1 **Verification of secrets**

Hierarchical to: No other components

Dependencies: No dependencies

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [assignment: *a defined quality metric*].

[131] [assignment: *list of authentication events*]

A.7.4.6.2 SFR Dependency Rationale

Table 14: SFR Dependency Rationale

SFR	Dependency resolution
FIA_AFL.1/LUA	FIA_UAU.2/LUA is hierarchical to FIA_UAU.1
FIA_SOS.1	No dependencies
FIA_UAU.2/LUA	FIA_UID.1 provides user identification in the base PP

A.7.4.6.3 Rationale for the Security Requirements

Table 15: SFR Dependency Rationale

	O.I&A
FIA_AFL.1/LUA	×
FIA_SOS.1	×
FIA_UAU.2/LUA	×

O.I&A

FIA_UAU.2/LUA requires that identified users are authenticated successfully before any other TSF mediated action may be performed. FIA_AFL.1/LUA requires reaction to failed authentication attempts. FIA_SOS.1 rejects weak credentials.

A.8 Guidance for SFR for RNG

A.8.1 Introduction

The quality of the random numbers produced by the random number generator FCS_RNG.1 is essential for the security claims of FCS_QKD.1. Some national certification bodies have issued recommendations for entropy sources. Although these have not been mutually recognized throughout the Common Criteria members, they provide a reasonable guidance for the requirements to FCS_RNG.1 in this PP.

ST authors shall choose the random number generator as close as possible to an ideal source and compatible with the assumed sources of randomness in the security proof relevant for FCS_QKD.1. ST authors are advised to ask the responsible certification body for adequate choices.

For purposes unrelated to FCS_QKD.1 ST authors may use iterations of FCS_RNG.1, which may have different security requirements.

A.8.2 RNG according to AIS 31

The German Federal Office for Information Security (BSI) published mandatory evaluation requirements for the German Common Criteria certification scheme [i.6]. These documents describe predefined classes of random number generators (cf. [i.5]). The class PTG.3 is appropriate for the TOE of this protection profile.

If the ST author selects the pre-defined class PTG.3 the SFR FCS_RNG.1 will look like this (operations shall be performed by the ST author):

FCS_RNG.1/PTG3 Random number generation – Physical random number generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a *hybrid physical*[132] random number generator that implements:

[132] [selection: *physical, hybrid physical*]

(PTG.3.1) A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure has been detected no random numbers will be output.

(PTG.3.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG [selection: prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source, generates the internal random numbers with a post-processing algorithm of class DRG.3 as long as its internal state entropy guarantees the claimed output entropy].

(PTG.3.3) The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG is started, and (ii) while the RNG is being operated. The TSF shall not output any random numbers before the power-up online test and the seeding of the DRG.3 post processing algorithm have been finished successfully or when a defect has been detected.

(PTG.3.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.

(PTG.3.5) The online test procedure checks the raw random number sequence. It is triggered [selection: externally, at regular intervals, continuously, upon specified internal events]. The online test is suitable for detecting nontolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.

(PTG.3.6) The algorithmic post-processing algorithm belongs to Class DRG.3 with cryptographic state transition function and cryptographic output function, and the output data rate of the post-processing algorithm shall not exceed its input data rate.[133]

FCS_RNG.1.2 The TSF shall provide random numbers that meet

(PTG.3.7) Statistical test suites cannot practically distinguish the internal random numbers from output sequences of an ideal RNG.

(PTG.3.8) The internal random numbers shall [selection: use PTRNG of class PTG.2 as random source for the post-processing, have [assignment: work factor], require [assignment: guess work]].[134]

A.8.3 RNG according to NIST SP 800-90

The National Institute of Standards and Technology (NIST) published NIST Special Publication 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation, January 2018 [i.7]. The recommendation for entropy sources [i.7] describes security requirements and test procedures that may be applied to the entropy source of a physical random number generator appropriate for the TOE.

If the ST author selects a physical random number generator compliant to [i.7] the SFR FCS_RNG.1 will look like this (operations shall be performed by the ST author):

FCS_RNG.1/ES Random number generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a *hybrid physical*[135] random number generator that implements:

[133] [assignment: *list of security capabilities*]

[134] [assignment: *a defined quality metric*]

[135] [selection: *physical, hybrid physical*]

(ES.1) Following continuous health tests for the noise source: [selection: Repetition Count Test, [assignment: alternative developer-defined test]] and [selection: Adaptive Proportion Test, [assignment: alternative developer-defined test]].

(ES.2) Conditioning component using one of the vetted algorithm: [selection: HMAC, CMAC, CBC-MAC, hash function, Hash_df, Block_Cipher_df] with [selection: AES128, AES256, SHA256, SHA384, SHA512].

(ES.3) [assignment: list of additional security capabilities]. [136]

FCS_RNG.1.2 The TSF shall provide [selection: *bits, octets of bits, numbers [assignment: format of the numbers]*] that meet

(ES.4) the output min-entropy value estimated according the estimating procedure is full entropy.[137]

Application Note 38: Note that non-vetted conditioning component is not acceptable because (ES.4) requires full entropy. The entropy estimation procedure is shown in NIST Special Publication 800-90B [i.7], clause 3.

A hybrid-physical design was chosen to ensure uniformly distributed random numbers even if the noise source is (temporarily) biased in a way that evades the health tests.

A.9 Reference documentation

A.9.1 Normative references

[1] Common Criteria for Information Technology Security Evaluation: "Part 1: Introduction and General Model", Version 3.1, Revision 5, CCMB-2017-04-001, April 2017.

[2] Common Criteria for Information Technology Security Evaluation: "Part 2: Security Functional Components", Version 3.1, Revision 5, CCMB-2017-04-002, April 2017.

[3] Common Criteria for Information Technology Security Evaluation: "Part 3: Security assurance components", Version 3.1, Revision 5, CCMB-2017-04-003, April 2017.

A.9.2 Informative references

[i.1] ETSI GS QKD 004: "Quantum Key Distribution (QKD); Application Interface", V1.1.1, 2010-12.

[i.2] ETSI GS QKD 008: "Quantum Key Distribution (QKD); QKD Module Security Specification", V1.1.1, 2010-12.

[i.3] ETSI GS QKD 005: "Quantum Key Distribution (QKD); Security Proofs", V1.1.1, 2010-12.

[i.4] Joint Interpretation Library: "Minimum Site Security Requirements", Version 2.2, April 2019.

[i.5] Bundesamt für Sicherheit in der Informationstechnik AIS31 — Wolfgang Killmann, Werner Schindler: "A proposal for: Functionality classes for random number generators", Version 2.0, 18 September 2011.

[i.6] Bundesamt für Sicherheit in der Informationstechnik: "Evaluation of random number generators", Version 0.8.

[i.7] NIST Special Publication 800-90B: "Recommendation for the Entropy Sources Used for Random Bit Generation", January 2018.

[136] [assignment: *list of security capabilities*]

[137] [assignment: *a defined quality metric*]

[i.8] Jörn Müller-Quade and Renato Renner: "Composability in quantum cryptography", New J. of Phys. 11, 085006 (2009).

NOTE: Available at http://doi.org/10.1088/1367-2630/11/8/085006

A.9.3 Bibliography

- Common Methodology for Information Technology Security Evaluation: "Evaluation Methodology", Version 3.1, Revision 5, CCMB 2017-04-004, April 2017.
- ISO 7498-2:1989: "Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 2: Security Architecture".
- ISO/IEC 18031:2011(en): "Information technology — Security techniques — Random bit generation".
- ISO/IEC 18031:2011/Cor 1:2014: "Information technology — Security techniques — Random bit generation — Technical Corrigendum 1".
- DIN EN ISO/IEC 19790:2020-08: "Information technology - Security techniques - Security requirements for cryptographic modules" (ISO/IEC 19790:2012, Corrected version 2015-12); German version EN ISO/IEC 19790:2020.
- ETSI GR QKD 007:" Quantum Key Distribution (QKD); Vocabulary", V1.1.1, 2018-12.
- NIST Special Publication 800-90A: "Recommendation for Random Number Generation Using Deterministic Random Bit Generators", Rev. 1, June 2015.
- Ivan B Djordjevic: "Physical-Layer Security and Quantum Key Distribution"; Springer International Publishing; Version 1, 2019.
- Christopher Portmann: "Key recycling in authentication"; IEEE Transactions on Information Theory, 60(7):4383–4396, July 2014.

A.10 Keywords and Abbreviations

Table 16: Glossary

Term	Description
active probing	physical probing with additional active physical interaction with the probed device NOTE: Active physical interactions may force the TOE to produce leakage that would otherwise not be emitted.
ADR Signing Key (ASK)	private key to sign ADR for export
Audit Data Records (ADR)	organized data generated for auditable events
Authentication Reference Data (ARD)	data used by the TOE to verify the AVD sent by a user and in turn authenticate the user
Authentication Verification Data (AVD)	data used by the user to authenticate themselves to the TOE
authenticity	property that ensures that the identity of an entity or the source of unmodified information is the one claimed (cf. ISO/IEC 7498-2:1989)
calibration	operation performed on calibration data by a user, including the comparison of measurement values delivered by the TOE with those of a calibration standard of known accuracy
calibration data	physical parameters of the underlying platform, that are adjustable and verifiable by a user, and that are required to be properly adjusted for the TOE to perform the QKD protocol securely NOTE: Calibration data is considered TSF data. Calibration data may also refer to physical properties requiring physical tools for modification.
certification body	body issuing Common Criteria certificates that is accredited by a nationally recognized accrediting body
coherent attack	most general type of eavesdropping attack on the quantum channel, where an adversary interacts multiple ancillas coherently with QKD signals and then performs a joint measurement on all the ancillas and/or QKD signals to extract information
cryptographic key	variable parameter that is used in and determines the functional output of a cryptographic algorithm or protocol
data integrity	property that data has not been altered or destroyed in an unauthorized manner (cf. ISO/IEC 7498-2:1989)
Maintainer	user authorized to perform calibrations
Operational State	state of the operational life-cycle as defined in clause A.1.3
private key	confidential key used for asymmetric cryptographic mechanisms like decryption of cipher text, signature-creation for authentication proof, where it is infeasible for the adversary to derive the confidential private key from the known public key
public key	publicly known key used for asymmetric cryptographic mechanisms like encryption of cipher text, signature-verification for authentication verification, where it is infeasible for an adversary to derive the confidential private key from the known public key
prepare and measure protocol	protocol for a QKD system to establish QKD keys in which one QKD module prepares quantum states and the other measures the quantum states
QKD Authentication Key (QAK)	shared secret used for authentication mechanisms between both QKD modules NOTE: The authentication is required to ensure the proper functionality of the prepare and measure protocol. The QKD authentication keys have to be available to the QKD modules before any communication using the QKD link can be established.
QKD key	pair of secret random bit strings established by a QKD system jointly in both QKD modules after successfully running a QKD protocol and considered to be identical NOTE: QKD keys are exportable to authorized users for further use.
QKD link	set of active and/or passive components that connect a pair of QKD modules to enable them to perform QKD
QKD module	set of hardware, software, and/or firmware components that implements a part of a QKD protocol as well as cryptographic functions to be capable of securely establishing shared, confidential, random bit strings with at least one other QKD module

QKD protocol	algorithm that either aborts at any time or produces a shared, random, confidential bit string in the transmitter and receiver modules
QKD system	pair of QKD modules, interconnected by a quantum channel and a classical channel, i.e. a QKD link
QKD transaction	set of information defined by the ST author that is exchanged over the classical channel in a QKD link using QAK(s) that are not used by any other QKD transaction and that is limited by time, data exchanged and other limitations
quantum key distribution	procedure involving the transport of quantum states to agree shared secret bit strings between remote parties using a protocol with security based on quantum entanglement or the impossibility of perfectly cloning or measuring the unknown transported quantum states
remote entities	human users or IT devices consuming QKD keys, which eventually operate on behalf of human users, and communicate through a trusted path with the TOE NOTE: The term is used solely in clause A.7.1 to point out a potentially indirect communication between human users and the TOE.
transaction	set of information defined by the ST author that is exchanged over a trusted path and limited by time, amount of data exchanged and additional limitations
trusted path	communication channel between QKD modules and remote entities that is logically distinct from other communication paths and that provides assured identification of its end points and protection of the communicated data from modification and disclosure
user	an entity using the TOE NOTE: A user can either be a machine (on behalf of a human or other machines) or a human interacting with the TOE.
User Definition Records (UDR)	information about known users and their associated roles
User Transaction Key (UTK)	set of distinct cryptographic keys, where each key is used exclusively to protect data on the trusted path either against modification or disclosure

Table 17: Abbreviations

Abbreviation	Term
A.xxx	Assumption
ADR	Audit Data Records
ARD	Authentication Reference Data
ASK	ADR Signing Key
AVD	Authentication Verification Data
CB	Certification Body
CC	Common Criteria
IT	Information Technology
ITS	Information Technology Security
n.a.	not applicable
O.xxx	Security Objective for the TOE
OE.xxx	Security Objective for the TOE Environment
OSP.xxx	Organisational Security Policy
PP	Protection Profile
P&M protocol	Prepare and Measure QKD protocol
QAK	QKD Authentication Key
QKD	Quantum Key Distribution
SAR	Security Assurance Requirements
SFP	Security Functional Policy
SFR	Security Functional Requirement
ST	Security Target
T.xxx	Threat
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSP	TOE Security Policy
UDR	User Definition Records
UTK	User Transaction Key

Annex B (informative): Roles, TOE users and TSFs

Editorial notes:

This is an informative annex and the text is not yet stable.

Verbs such as "shall" and "should" to be replaced (this annex is informative).

Use of more Common Criteria terminology would help link explanations to the text in the PP. E.g., the establishment of a "trusted path" by a remote user, or a description of managing the attributes used to make explicit access or denial-based decisions for "security attribute-based access control" via "User Definition Records" etc.

B.1 Rationale

In clause A.1.3 TOE users of this Protection Profile (hereinafter called this PP), four *roles*, i.e., *Administrator, Maintainer, Auditor and Key Requester*, are introduced as follows:

Administrator: successfully authenticated user allowed to access the TOE in order to perform user management functions.

Maintainer: successfully authenticated user allowed to access the TOE in order to perform management functions of specific cryptographic TSF to ensure proper functionality of the QKD modules and to impede physical attacks on the two QKD modules from beyond the security perimeters.

Auditor: successfully authenticated user allowed to access the TOE in order to perform management of auditable events and to access Audit data.

Key Requester: successfully authenticated user allowed to perform key distribution service operations.

This Annex B is devoted to clarifying what is meant by the term *roles*, by providing how *roles*, TOE users and TSFs are mutually related with the following pieces of information:

- A diagram that shows which individual *role* holds responsibility for which phase(s) (in clause B.2).
- Explanation of how TOE users become able to possess TSFs (in clause B.3).
- Diagrams that show how two QKD modules and TOE users are connected through user interfaces of the modules (in clause B.4).

B.2 A diagram that shows which individual role holds responsibility for which phase(s)

The PP mentions a generic life-cycle for the TOE within clause A.1.3 (page 15). According to the life-cycle model therein, the life-cycle is made up of several high-level phases starting from ‘Development’ to ‘End of Life’. Furthermore, the post ‘Delivery’ phases (‘Personalization’ to ‘End of Life’) are decomposed into subphases, which are the ones that the four *roles* administer.

Figure 5 shows which individual *role* holds responsibility for which subphase(s). Brief descriptions in speech balloons near the corresponding names of the roles are their traits, which are extracted out of their descriptions found in this PP.

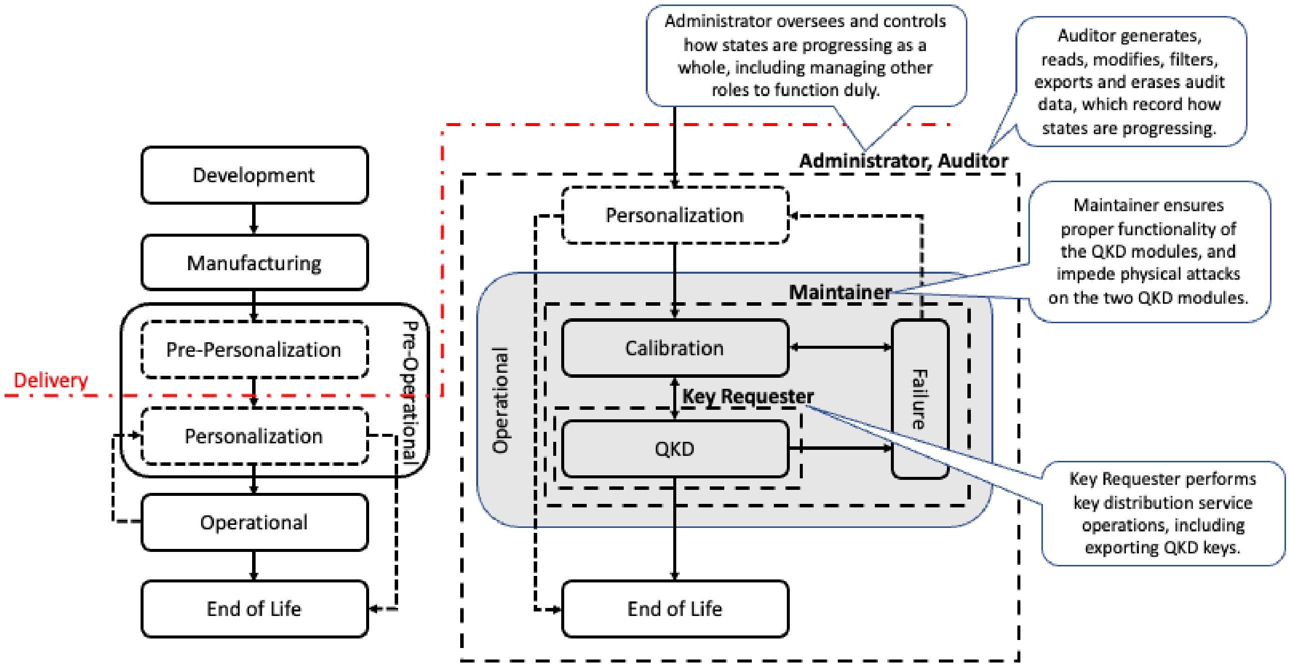

Development
Manufacturing
Pre-Personalization
Personalization
Pre-Operational
Delivery
Operational
End of Life
Administrator oversees and controls how states are progressing as a whole, including managing other roles to function duly.
Auditor generates, reads, modifies, filters, exports and erases audit data, which record how states are progressing.
Administrator, Auditor
Personalization
Maintainer
Maintainer ensures proper functionality of the QKD modules, and impede physical attacks on the two QKD modules.
Operational
Calibration
Key Requester
QKD
Failure
Key Requester performs key distribution service operations, including exporting QKD keys.
End of Life

Figure 5: Life-cycle model with individual *roles*. It is created by modification of Figure 4 in this PP

B.3 Explanation of how TOE users become able to possess TSFs

Following the discussions about which *role* administers which subphase(s) in the life-cycle for the TOE, this chapter examines how TOE users become able to possess TSFs. Through the examination, we will see that *roles* play an essential part to associate TOE users with TSFs. We will examine it in three stages: (1) defining a table to associate Roles with TOE users, (2) connecting Roles to TOE users, then (3) leading to TOE users' possessing specific security functionalities or TSFs.

NB In the following discussions, the names of SFRs and the associated TSFs thereof are correct at the time of writing this Annex A. They may be modified or even deleted afterwards.

B.4 Stage-1: defining a table to associate Roles with TOE users

To associate TOE users with TSFs, the first thing we do is to write up a table that defines the association of each individual Roles with TOE user IDs, where the association complies with the SFRs comprising FMT_SMR.1 Security roles, FMT_MSA.2 Secure security attributes and FIA_ATD.1 User attribute definition. Table 18 may be the simplest example for this purpose, and we will use the association in Table 18 in the following discussions.

Table 18: Association of each individual roles with TOE users

Roles	TOE user IDs
Administrator	Mr A (human)
Auditor	HostA (IT-device)
Maintainer	HostM (IT-device)
Key Requester	HostK (IT-device)
[Assignment: other roles]	

B.5 Stage-2: connecting Roles to TOE users

Having the association table (Table 18), then *FIA_USB.1 User-subject binding* requires connecting each individual Roles to TOE users. Specifically, Mr A (human) is now connected to *Administrator*, for example. Similarly, HostA (IT-device), HostM (IT-device) and HostK (IT-device) are now connected to *Auditor*, *Maintainer* and *Key Requester*, respectively.

B.6 Stage-3: TOE user's possessing specific functions

The establishment of links between roles and TOE users mean that Mr A will now become able to perform what are given to *Administrator* in terms of SFRs, for example. Similarly, HostA (IT-device), HostM (IT-device) and HostK (IT-device) can carry out what are given to *Auditor*, *Maintainer* and *Key Requester* in terms of TSFs, respectively.

Functions in the following SFRs will be shared among TOE users: *FDP_ACF.1 Security attribute-based access control*, *FMT_MTF.1 Management of TSF data*, *FMT_MTD.1/Adm Management of TSF data – Administrator*, and *FMT_MOF.1 Management of security functions behaviour*. Table 19 show which specific security functions are given to which TOE users.

Table 19: Allocation of security functions to TOE users

Security functions	Roles	TOE user IDs
FMT_MTD.1.1/Adm create and delete the Authentication Data Records of an authorized user to Administrator, modify the Authentication Reference Data of users to Administrator, modify the Role of an authorized user to Administrator,	Administrator	Mr A (human)
FMT_MTD.1.1 manually export, clear after export, select audited events in the audit records to Auditor, define, modify the thresholds for actions to be taken according to FAU_STG.3 to Auditor **FMT_MOF.1.1** determine the behaviour of the functions auditable events according to FAU_GEN. to Auditor. modify the behaviour of the functions assign additional auditable events according to FAU_GEN.1 to Auditor. determine and modify the behaviour of the functions actions to be taken in case of possible audit storage failure according to FAU_STG.3 to Auditor.	Auditor	HostA (IT-device)
FMT_MTD.1.1 change default, query, modify the calibration data to Maintainer,	Maintainer	HostM (IT-device)
FDP_ACF.1.3 Subject in Key Requester Role is allowed to export QKD keys, while the TSF is situated in the QKD state, Subject in Key Requester Role is allowed to access key distribution services, while the TSF is situated in the QKD state,	Key Requester	HostK (IT-device)
	[Assignment: other roles]	
The following functions shall be dealt with in ST, in terms of which roles will cover them. FDP_ACF.1.3		

B.7 Sequence of role allocation and key exporting

An example of actual implementation follows:

- Necessary information for cryptographic communication and authentication such as digital certificates or pre-shared key shall be provided to QKD modules and TOE users in advance.

- ID (IP address, etc.) of the TOE user to which QKD key is exported is stored in QKD module, and ID (IP address, etc.) of the QKD module where QKD key exports is stored in TOE user. Other IDs for administrator, maintainer and auditor are stored as well.

- When the TOE user logs in, a secure channel (such as TLS) is established from the TOE user to the TOE. The TOE authenticates TOE user with the certificates, and the TOE allocates the role (such as key requester) to the TOE user based on the user ID.

- Key requester sends a request establishment of QKD key to the QKD module. A secure channel (such as TLS) is established from the QKD module to the key requester if QKD module uses the separate channel for exporting QKD key.

- The QKD module exports generated QKD keys to the key requester through the secure channel.

Actual signalling for exporting QKD keys from the QKD module to the key requester may vary by protocols which are implemented in the QKD system such as push from the QKD module or pull from the key requester.

Figure 6 shows the sequence of role allocation and key exporting processes. A similar sequence can be applied to other TOE users (i.e. Administrator, Maintainer and Auditor).

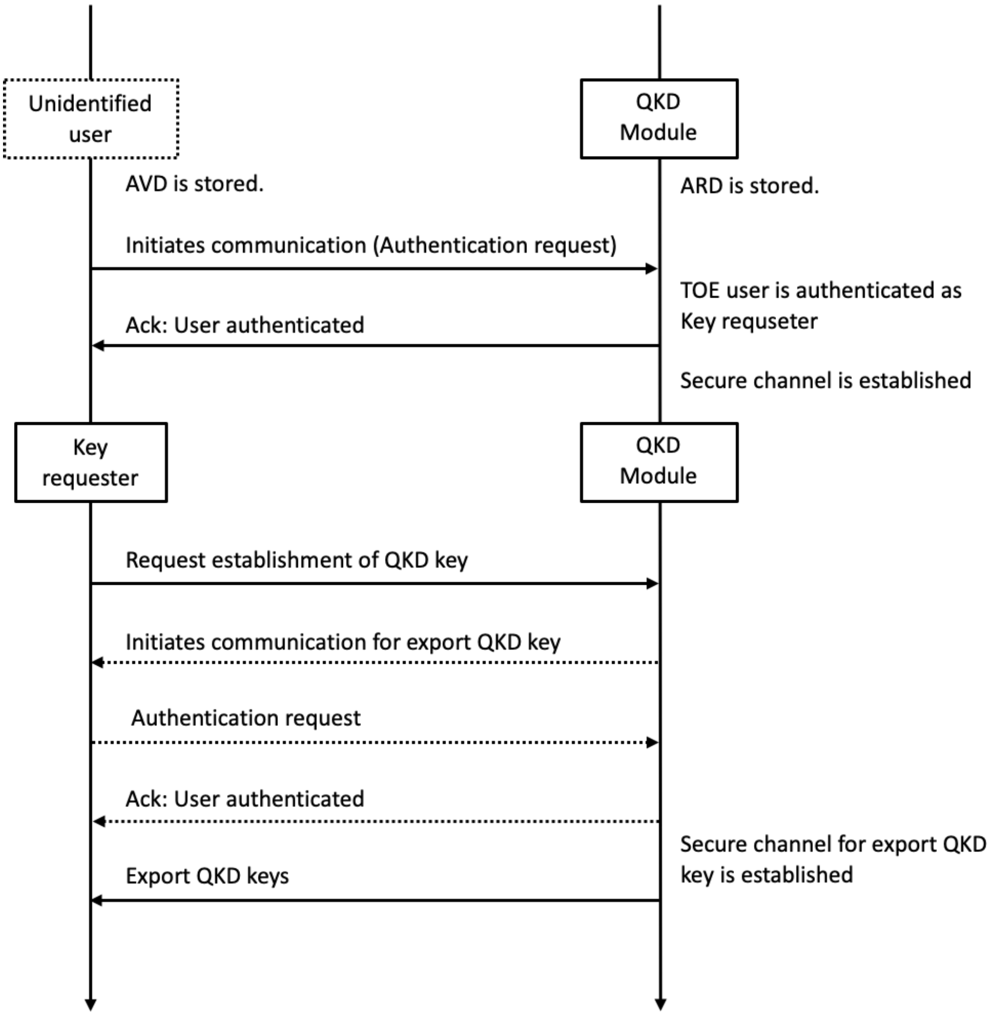

Unidentified user
QKD Module
AVD is stored.
ARD is stored.
Initiates communication (Authentication request)
TOE user is authenticated as Key requseter
Ack: User authenticated
Secure channel is established
Key requester
QKD Module
Request establishment of QKD key
Initiates communication for export QKD key
Authentication request
Ack: User authenticated
Secure channel for export QKD key is established
Export QKD keys

Figure 6: Role allocation and exporting QKD keys

B.8 How two QKD modules and TOE users are connected through user interfaces of the modules

Following clause B.3, this chapter is devoted to enhancing the picture of TOE users by giving diagrams of how they work with QKD modules. It should be noted that *Administrator, Auditor, Maintainer* and *Key Requester* in the following figures refer, not to *roles*, but to something that possess the security functions thereof. That is, these four names are to represent corresponding TOE users. For example, *Auditor* in the following figures refers to a TOE user that is able to perform a bunch of security functions in the name of *Auditor*.

Figure 7 shows an operational environment that depicts how two QKD modules and TOE users are connected through user interfaces of the modules, in the most practical case where QKD Transmitter, QKD Receiver and Key requester are being laid out within a physically protected area while other TOE users are outside the area.

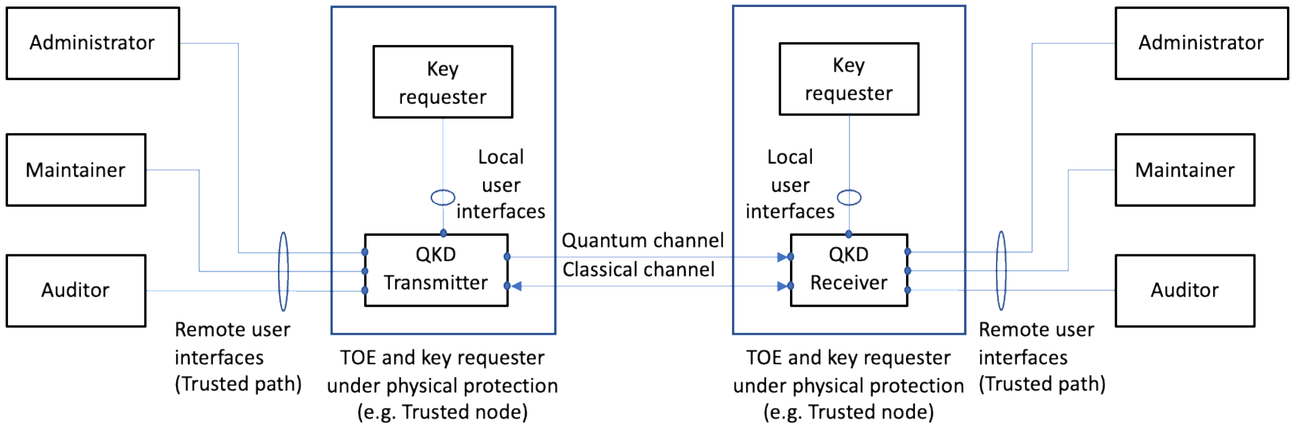

Administrator
Maintainer
Auditor
Key requester
Local user interfaces
QKD Transmitter
Remote user interfaces (Trusted path)
TOE and key requester under physical protection (e.g. Trusted node)
Quantum channel
Classical channel
Key requester
Local user interfaces
QKD Receiver
TOE and key requester under physical protection (e.g. Trusted node)
Remote user interfaces (Trusted path)
Administrator
Maintainer
Auditor

Figure 7: TOE and Key requesters are being laid out within a physically protected area

B.9 Summary

This Annex B has discussed what is meant by the term *roles* in terms of three different aspects as follows: which individual *role* holds responsibility for which phase(s) within the life-cycle for the TOE, how important *roles* are to associate TOE users with TSFs and how the two QKD modules are connected with TOE users. Although each individual *role* formally refers to something conceptual that possesses a bunch of certain TSFs, it is used as something physical depending on the context. When using the term, close attention should be paid to which of the two meanings the term is referring to.

Annex C (informative): Bibliography

- Common Methodology for Information Technology Security Evaluation: "Evaluation Methodology", Version 3.1, Revision 5, CCMB 2017-04-004, April 2017.
- ISO 7498-2:1989: "Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 2: Security Architecture".
- ISO/IEC 18031:2011(en): "Information technology — Security techniques — Random bit generation".
- ISO/IEC 18031:2011/Cor 1:2014: "Information technology — Security techniques — Random bit generation — Technical Corrigendum 1".
- DIN EN ISO/IEC 19790:2020-08: "Information technology - Security techniques - Security requirements for cryptographic modules" (ISO/IEC 19790:2012, Corrected version 2015-12); German version EN ISO/IEC 19790:2020.
- ETSI GR QKD 007:" Quantum Key Distribution (QKD); Vocabulary", V1.1.1, 2018-12.
- NIST Special Publication 800-90A: "Recommendation for Random Number Generation Using Deterministic Random Bit Generators", Rev. 1, June 2015.
- Ivan B Djordjevic: "Physical-Layer Security and Quantum Key Distribution"; Springer International Publishing; Version 1, 2019.
- Christopher Portmann: "Key recycling in authentication"; IEEE Transactions on Information Theory, 60(7):4383–4396, July 2014.

Annex D (informative): Change History

Date	Version	Information about changes
September 2021	V0.5.3	Transfer of content to GS skeleton.
September 2021	V0.5.4	Editorial changes only (mainly to improve language).
November 2021	V0.6.1	Changes merged up to and including QKD(21)000012.
December 2021	V0.6.2	Initial changes from QKD#31 and editorial changes.

History

<table>
<tr><th colspan="3">Document history</th></tr>
<tr><td>&lt;Version&gt;</td><td>&lt;Date&gt;</td><td>&lt;Milestone&gt;</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
</table>