Cryptographic inventory and evidence under Articles 6 and 7(4)

Commission Delegated Regulation (EU) 2024/1774, regulatory technical standards on ICT risk management, supplementing Article 9(2) of Regulation (EU) 2022/2554 (DORA).
Target: `paramiko` · Generated: 2026-11-05T07:22:23Z · Tool: qrp-dora 0.13.0 · Scanner: qrp-mcp 0.44.0
Scan run at: 2026-11-05T07:22:23+00:00 to 2026-11-05T07:22:24+00:00

What this document is and is not. This is an **evidence map**: it describes the cryptography found in the scanned material and which clause of Articles 6 and 7(4) it relates to. The document **does not establish compliance** and cannot establish it. Compliance is declared by the entity and assessed by an auditor; the scan supplies part of the evidence they rely on. Clause wording here is **paraphrased, not quoted**; the binding text is the regulation itself.

Summary

Findings: 36 · algorithm families: 13. Families by category: broken today: 3 · quantum-vulnerable: 8 · post-quantum: 1.

Coverage: The scanner read 110 of 196 files present; the remaining 86 were not read, by the scanner's reason: 86 type_not_claimed (the tool does not claim this file type; a declared boundary, not a failure).

Read in part: of the files read, 5 were read only in part (53.57% of the files present were read in full).

Replacement profile: de-bsi — BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths (2026-01, 23 January 2026).

Clauses under which this scan found something to act on: Art. 6(3) — 3 families broken today; Art. 6(4) — 3 families broken today, 8 families quantum-vulnerable; Art. 6(5) — the measures adopted under 6(3) and 6(4) are to be recorded, with the reasons; Art. 7(4) — 26 certificate and key files read, 24 embedded private keys: candidate register entries.

1. Scope and completeness

Source code, continuous-integration configuration and infrastructure as code under the given path were examined.

Kind	Files
certificates and keys	26
CI configuration	1
configuration files	12

infrastructure as code	0
source code	71

Coverage (as reported by the scanner): 196 files present; 0 could not be read; 86 skipped as types the scanner does not claim; 0 directories could not be entered. Every file accounted for: yes. This scan read 110 of 196 files (56.12%); the remaining 86 (3 of 6, by extension — files with no extension, .rst and .png — being 96.51% of them) are listed with a reason each; of those read, 5 were read only in part or gave nothing it could decode, each named with a reason (53.57% of the files given were read in full).

Read in part (as reported by the scanner). Opened: 56.12% of the files given (coverage_pct); read in full: 53.57% (read_in_full_pct); read only in part: 5.

- tests/test_ecdsa_password_256.key:1 (EC PRIVATE KEY): password_protected — encrypted under a password, or damaged: it did not open without one
- tests/test_ecdsa_password_384.key:1 (EC PRIVATE KEY): password_protected — encrypted under a password, or damaged: it did not open without one
- tests/test_ecdsa_password_521.key:1 (EC PRIVATE KEY): password_protected — encrypted under a password, or damaged: it did not open without one
- tests/test_rsa_password.key:1 (RSA PRIVATE KEY): password_protected — encrypted under a password, or damaged: it did not open without one
- tests/blank_rsa.key: claimed_but_not_decoded — a claimed file type that was read and gave up no algorithm: encrypted content, or a structure this tool does not parse

Coverage says which files were **read**. It does not say that everything present in them was **found**; no scanner can state that about itself.

Who ran the scan, on what, with which scanner, and when.
Scan run by: Elkan Electronic
Scope, as stated: Public repository paramiko, commit 0e9ccb91e0f9a68f2587233d523ad7fe46e730fe
Scanner: qrp-mcp 0.44.0, as recorded in the scan result
Run at: 2026-11-05T07:22:23+00:00 to 2026-11-05T07:22:24+00:00

Completeness: partial. Running systems, certificates in use, key material in hardware modules and purchased software without available source **are not observed** and are therefore absent from this inventory. The absence of an algorithm from this document does not mean the entity does not use it; it means it was not in scope.

The reverse also holds: findings include test files, string constants and comments. An algorithm on the list means its name occurs in the code, not that it runs in a production system. Telling the two apart is the entity's task, which is why every occurrence is tied to a file and line.

1c. Control planted into a copy of your tree

A control planted into a copy of this tree — 27 cryptographic uses in 9 styles and 18 lines that name an algorithm without using it — was scanned with qrp-mcp 0.44.0 (revision cb4297acf558), seed 1. It found 23 of the 27 uses, was silent on 3 (a name carried by an alias or a constant: 1; a

name built from string pieces: 2); 1 found otherwise (read as not a use: 1), and reported 3 of the 18 non-uses as uses.

What it does not say. This says what the instrument does on those styles. It does not say what this tree contains, how much of it was found, or that the inventory is complete; it is not a recall or a precision figure, and it is for this version only (qrp-mcp 0.44.0 (revision cb4297acf558)). Nothing planted reaches `cbom.json`, the findings or the counts of this report: the copy is scanned apart and discarded.

Planted uses, by style.

Style	Planted	Found	Silent
a direct call	3	3	0
a name carried by an alias or a constant	3	1	1
a name built from string pieces	3	1	2
a call split over lines	3	3	0
a shell command inside a string	3	3	0
a cert-manager YAML key	3	3	0
a Terraform tls key	3	3	0
a Java properties setting	3	3	0
a UTF-16 file without a byte-order mark	3	3	0

Planted non-uses, by kind.

Line that names an algorithm without using it	Planted	Reported as a use
a setting that switches an algorithm off	3	0
a list of algorithms refused	3	0
a log message naming an algorithm	3	0
a variable named after an algorithm	3	3
a comment	3	0
a key label naming another algorithm	1	0
an sshd_config line that removes algorithms	2	0

Rows on planted lines, by the confidence the scanner gave them.

Confidence	On a planted use	On a planted non-use

high	15	0
medium	8	0
low	0	3

Seed 1 · harness SHA-256 91f58ae17b8ed423213c9fb64bc07b77bb82ec823dbb468041dc42ca7783ddaa · result SHA-256 f716a9696aaf595bf2b5e3fec6c1fa083442527ab5719878fe70f4721cd3bdfa

2. What was found

Total findings: 36. Distinct algorithm families: 13. By category: broken today: **3** · quantum-vulnerable: **8** · post-quantum: **1** · post-quantum, not counted as in use: **0** · other: **1**.

Switched off in the files read, not in use and not counted: DH (tests/test_transport.py:1111, tests/test_transport.py:1132); Ed25519 (tests/test_client.py:685, tests/test_client.py:687); MD5 (tests/test_transport.py:1109, tests/test_transport.py:1130); RSA (tests/auth.py:160, tests/auth.py:178, tests/auth.py:187, tests/auth.py:203, tests/auth.py:213, tests/test_transport.py:1110, tests/test_transport.py:1131, tests/test_transport.py:1181, tests/test_transport.py:1237).

Algorithm	Classification	Posture	Severity	Why
3DES	broken today	—	high	Broken by today's standards, regardless of quantum computers. Falls short of leading practice: replace, or adopt a mitigation measure under Art. 6(3) and record it under Art. 6(5).
DH	classical, quantum-vulnerable	quantum-vulnerable	high	Breakable by Shor's algorithm on a cryptographically relevant quantum computer. To be planned for under Art. 6(4).
EC	classical, quantum-vulnerable	quantum-vulnerable	high	Breakable by Shor's algorithm on a cryptographically relevant quantum computer. To be planned for under Art. 6(4).
ECDH	classical, quantum-vulnerable	quantum-vulnerable	high	Breakable by Shor's algorithm on a cryptographically relevant quantum computer. To be planned for under Art. 6(4).

ECDSA	classical, quantum-vulnerable	quantum-vulnerable	high	Breakable by Shor's algorithm on a cryptographically relevant quantum computer. To be planned for under Art. 6(4).
Ed25519	classical, quantum-vulnerable	quantum-vulnerable	high	Breakable by Shor's algorithm on a cryptographically relevant quantum computer. To be planned for under Art. 6(4).
Ed448	classical, quantum-vulnerable	quantum-vulnerable	high	Breakable by Shor's algorithm on a cryptographically relevant quantum computer. To be planned for under Art. 6(4).
MD5	broken today	—	high	Broken by today's standards, regardless of quantum computers. Falls short of leading practice: replace, or adopt a mitigation measure under Art. 6(3) and record it under Art. 6(5).
ML-KEM	post-quantum	post-quantum	info	Post-quantum family; not affected by Shor's algorithm.
RSA	classical, quantum-vulnerable	quantum-vulnerable	high	Breakable by Shor's algorithm on a cryptographically relevant quantum computer. To be planned for under Art. 6(4). Key size not read: whether the key is below the classical minimum is not established.
SHA1	broken today	—	high	Broken by today's standards, regardless of quantum computers. Falls short of leading practice: replace, or adopt a mitigation measure under Art. 6(3) and record it under

				Art. 6(5).
X25519	classical, quantum-vulnerable	quantum-vulnerable	high	Breakable by Shor's algorithm on a cryptographically relevant quantum computer. To be planned for under Art. 6(4).
private_key ×24	unknown	—	critical	Not classified by the scanner (`unknown`). Private key material committed to the repository. 24 findings, each with its file and line in `evidence/scan.json` (`evidence.embedded_keys`).

2b. In this code since: dates from git

Of the 889 findings with a file and a line: 889 dated from git at commit 0e9ccb91e0f9; 373 walked back past their last edit; 18 reached a bound (the first commit or the step limit).

Per family. In this code at least since: the earliest date the walk reaches on a line still in the tree. Newest line: the latest date a line carrying it was written (basis blame).

Family	Lines	In this code at least since (basis)	Newest line	Walked back past the last edit
3DES	11	2003-11-04 (bound)	2024-07-22	5
DH	21	2003-11-04 (bound)	2026-02-28	8
EC	84	2013-03-24 (walked)	2026-03-29	56
EC PRIVATE KEY	6	2013-03-24 (walked)	2016-04-25	0
ECDH	18	2017-05-01 (walked)	2026-08-09	10
ECDSA	210	2013-03-24 (walked)	2026-03-29	122
Ed25519	101	2017-05-26 (walked)	2026-10-17	31

Ed448	4	2023-04-27 (walked)	2023-04-27	0
MD5	11	2003-11-04 (bound)	2019-06-21	6
ML-KEM	42	2026-08-09 (walked)	2026-08-09	0
OPENSSH PRIVATE KEY	10	2017-05-26 (walked)	2026-02-15	0
PRIVATE KEY	1	2023-04-27 (walked)	2023-04-27	0
RSA	260	2003-11-04 (bound)	2026-10-10	102
RSA PRIVATE KEY	3	2004-09-25 (walked)	2023-05-08	0
SHA1	12	2003-11-04 (bound)	2026-02-15	9
X25519	71	2019-02-09 (walked)	2026-08-29	14
cryptography	1	2025-08-03 (walked)	2025-10-18	1
embedded_keys	4	2006-06-26 (walked)	2016-04-25	0
openssl-dev	1	2021-11-27 (walked)	2024-02-10	1
pynacl	1	2025-08-03 (walked)	2025-10-18	1
ssh	17	2004-09-25 (walked)	2023-05-08	7

Per finding (the first in the order of the scan).

Place	Family	Line last written (basis: blame), commit	In this code since (basis), commit
paramiko/__init__.py:69	RSA	2014-02-11, e7f41de2f2	2003-12-27 (walked), eb4c279ec4

paramiko/__init__.py:70	ECDSA	2014-02-11, e7f41de2f2	2013-03-24 (walked), 632129c427
paramiko/__init__.py:71	Ed25519	2017-05-26, a8ff223226	2017-05-26 (walked), a8ff223226
paramiko/__init__.py:115	ECDSA	2025-07-20, e534b1abcf	2023-04-16 (walked), 0146cf7201
paramiko/__init__.py:115	RSA	2025-07-20, e534b1abcf	2023-04-16 (walked), 0146cf7201
paramiko/__init__.py:115	Ed25519	2025-07-20, e534b1abcf	2023-04-16 (walked), 0146cf7201
paramiko/agent.py:45	RSA	2021-12-23, 2b66625659	2021-12-23 (walked), 2b66625659
paramiko/agent.py:46	RSA	2021-12-23, 2b66625659	2021-12-23 (walked), 2b66625659
paramiko/agent.py:53	RSA	2021-12-23, 2b66625659	2021-12-23 (walked), 2b66625659
paramiko/agent.py:54	RSA	2021-12-23, 2b66625659	2021-12-23 (walked), 2b66625659
paramiko/agent.py:490	RSA	2023-05-24, fdb08b7cb9	2023-05-24 (walked), fdb08b7cb9
paramiko/auth_handler.py:473	RSA	2026-02-15, a4489456b6	2026-02-15 (walked), a4489456b6
paramiko/auth_handler.py:474	RSA	2026-02-15, a4489456b6	2026-02-15 (walked), a4489456b6
paramiko/auth_handler.py:475	RSA	2026-02-15, a4489456b6	2026-02-15 (walked),

			a4489456b6
paramiko/auth_handler.py:478	RSA	2026-02-15, a4489456b6	2026-02-15 (walked), a4489456b6
paramiko/auth_handler.py:714	RSA	2026-02-21, 6f02c1e3a4	2026-02-21 (walked), 6f02c1e3a4
paramiko/auth_handler.py:715	RSA	2026-02-21, 6f02c1e3a4	2026-02-21 (walked), 6f02c1e3a4
paramiko/client.py:421	RSA	2026-02-15, a4489456b6	2026-02-15 (walked), a4489456b6
paramiko/client.py:430	RSA	2026-02-15, a4489456b6	2026-02-15 (walked), a4489456b6
paramiko/client.py:431	RSA	2026-02-15, a4489456b6	2026-02-15 (walked), a4489456b6
paramiko/client.py:432	RSA	2026-02-15, a4489456b6	2026-02-15 (walked), a4489456b6
paramiko/client.py:433	RSA	2026-02-15, a4489456b6	2026-02-15 (walked), a4489456b6
paramiko/client.py:434	RSA	2026-02-15, a4489456b6	2026-02-15 (walked), a4489456b6
paramiko/client.py:453	RSA	2026-02-28, 1ecc933b34	2026-02-15 (walked), a4489456b6
paramiko/client.py:454	RSA	2026-02-28, 1ecc933b34	2026-02-15 (walked), a4489456b6

864 more in `evidence/in-this-code-since.json`.

How it was established. Plain git blame per file: the commit that last wrote the line in this file and its author date (basis blame). For every dated line, the walk back: the scanner reads the file at each parent commit and steps back while the same family is on the counterpart line (basis walked; bound at the first commit or after

15 steps). Per family: at least since = the earliest walked date of a line still in the tree (a bound: a removed line may have carried it earlier); newest line = the latest blame date of a line carrying it. Naive blame dates the last edit of a line, not when the algorithm entered it: on four public repositories the last edit only changed a line that already carried the algorithm in 19% to 41% of sampled lines. A squash merge dates the merge, not the branch's own commits (measured: median 3 days, up to 243). Upstream syncs and vendored copies date the copy, not the original. The walk maps a line to its parent version by text difference; a line moved far and rewritten at once can stop it early, and a file renamed and more than half rewritten in one commit is a new file to git. Dates are author dates from git; durations in the evidence file are measured to this scan's start (2026-11-05T07:22:23+00:00). A date with basis *walked* is where the scanner, reading each earlier version of the file, no longer finds the algorithm on that line; *bound* means the walk reached the first commit or its step limit, so the algorithm is there at least since that date. Dates and durations only; nothing here ranks one finding above another.

3. Evidence

Every finding is tied to a file and line, so it can be checked independently rather than taken on trust. Lines of code are quoted at level: masked -- only the characters that name the algorithm are kept; every other letter and digit is replaced by an asterisk, so the shape of the line shows without its content.

3DES — 11 occurrences

File and line	Excerpt
paramiko/pkey.py:50	`# ********* ** ****** ****` `` `************.******.**********.*******.********** ``
paramiko/pkey.py:59	`**** ************.******.********.*******.********** ****** *********`
paramiko/pkey.py:61	`**** ************.******.**********.*******.********** ****** *********`
paramiko/pkey.py:140	`"***-****-***": {`
paramiko/pkey.py:141	`"******": *********,`
paramiko/transport.py:126	`# ********* ** ****** ****` `` `************.******.**********.*******.********** ``

Showing 6 of 11; the remaining 5 are in `evidence/scan.json`.

DH — 19 occurrences

File and line	Excerpt
paramiko/kex_ecdh_nist.py:2	`********* ******** ***** ******-******* (**DH) *** ********`
paramiko/kex_gex.py:51	`**** = "******-*******-*****-********-******"`
paramiko/kex_group14.py:21	`******** *** *** ******** ("***" ** *** ***** ***** ****). ******-*******  **`
paramiko/	`**** = "******-*******-*******-******"`

kex_group14.py:45	
paramiko/kex_group16.py:20	******** *** *** ******** ("***" ** *** ***** ***** ****). ******-******* **
paramiko/kex_group16.py:30	**** = "******-*******-*******-******"

Showing 6 of 19; the remaining 13 are in evidence/scan.json.

EC — 84 occurrences

File and line	Excerpt
paramiko/ecdsakey.py:43	********** * **ec**** EC*** ***** (********, ********, ***).
paramiko/ecdsakey.py:105	_EC********(ec.*EC******, "********"), · curve P-256
paramiko/ecdsakey.py:106	_EC********(ec.*EC******, "********"), · curve P-384
paramiko/ecdsakey.py:107	_EC********(ec.*EC******, "********"), · curve P-521
paramiko/ecdsakey.py:167	*** = ec.**********.****_*******_*****(
paramiko/ecdsakey.py:250	*** *******_***(****) -> ********[ec.**********]:

Showing 6 of 84; the remaining 78 are in evidence/scan.json.

ECDH — 18 occurrences

File and line	Excerpt
paramiko/kex_ecdh_nist.py:2	********* ******** ***** ******-******* (ECDH) *** ********
paramiko/kex_ecdh_nist.py:21	**** = "ecdh-****-********" · curve P-256
paramiko/kex_ecdh_nist.py:71	* = ****.*.********(**.ECDH(), ****.*_*)
paramiko/kex_ecdh_nist.py:117	* = ****.*.********(**.ECDH(), ****.*_*)
paramiko/kex_ecdh_nist.py:143	**** = "ecdh-****-********" · curve P-384
paramiko/kex_ecdh_nist.py:149	**** = "ecdh-****-********" · curve P-521

Showing 6 of 18; the remaining 12 are in evidence/scan.json.

ECDSA — 210 occurrences

File and line	Excerpt
paramiko/__init__.py:70	`**** ********.ecdsa*** ****** ECDSA***`
paramiko/__init__.py:115	`***_******* = [******, **********, ECDSA***]`
paramiko/client.py:690	`*** ****_***** ** (******, ECDSA***, **********):`
paramiko/client.py:730	`(ECDSA***, "ecdsa"),`
paramiko/ecdsakey.py:20	`ECDSA ****`
paramiko/ecdsakey.py:41	`***** _ECDSA*****:`

Showing 6 of 210; the remaining 204 are in `evidence/scan.json`.

Ed25519 — 99 occurrences

File and line	Excerpt
paramiko/__init__.py:71	`**** ********.ed25519*** ****** Ed25519***`
paramiko/__init__.py:115	`***_******* = [******, Ed25519***, ********]`
paramiko/client.py:690	`*** ****_***** ** (******, ********, Ed25519***):`
paramiko/client.py:731	`(Ed25519***, "ed25519"),`
paramiko/ed25519key.py:30	`***** Ed25519***(****):`
paramiko/ed25519key.py:32	`************** ** ** `Ed25519 <*****://ed25519.**.**.**/>`_ ***.`

Showing 6 of 99; the remaining 93 are in `evidence/scan.json`.

Ed448 — 4 occurrences

File and line	Excerpt
tests/_support/ed448.key:1	`*.*.***.***`
tests/pkey.py:74	`# ***** **** ** ** *** *******. ***** Ed448 ******** ** *******`
tests/pkey.py:76	`******* = _*******("ed448.***")`
tests/pkey.py:79	`****** **********(***.*****.***_****, Ed448**********)`

MD5 — 9 occurrences

File and line	Excerpt
paramiko/pkey.py:390	`****** md5(****.*******()).******()`
paramiko/transport.py:193	`"****-md5",`
paramiko/transport.py:195	`"****-md5-**",`
paramiko/transport.py:295	`"****-md5": {"*****": md5, "****": **},`
paramiko/transport.py:296	`"****-md5-**": {"*****": md5, "****": **},`
tests/test_transport.py:232	`*******.******* = ("****-md5-**",)`

Showing 6 of 9; the remaining 3 are in `evidence/scan.json`.

ML-KEM — 42 occurrences

File and line	Excerpt
paramiko/kex_mlkem.py:2	`ML-KEM ****** *** ******** *** ***.`
paramiko/kex_mlkem.py:5	`********* ** *****-****-****-*****-******-*** (*.*.`
paramiko/kex_mlkem.py:6	``` ``**************-******``). ```
paramiko/kex_mlkem.py:8	`**** ****** ******** ** ML-KEM *** ************* **** * ***********`
paramiko/kex_mlkem.py:28	`***** = ****`
paramiko/kex_mlkem.py:37	`# *** *****-****-****-*****-******-***, *** ****** *** ****** *******`

Showing 6 of 42; the remaining 36 are in `evidence/scan.json`.

RSA — 251 occurrences

File and line	Excerpt
paramiko/__init__.py:69	`**** ********.rsa*** ****** RSA***`
paramiko/__init__.py:115	`***_******* = [RSA***, **********, ********]`
paramiko/agent.py:45	`***_*****_RSA_****_*** = *`
paramiko/agent.py:46	`***_*****_RSA_****_*** = *`

paramiko/agent.py:53	`"rsa-****-***": ***_*****_RSA_****_***,`
paramiko/agent.py:54	`"rsa-****-***": ***_*****_RSA_****_***,`

Showing 6 of 251; the remaining 245 are in `evidence/scan.json`.

SHA1 — 12 occurrences

File and line	Excerpt
paramiko/config.py:449	`"%*": sha1(******.******()).*********(),`
paramiko/hostkeys.py:294	`**** = **.*******(sha1().******_****)`
paramiko/hostkeys.py:299	`****** ***(****) == sha1().******_****`
paramiko/sftp_file.py:376	`*** *******, ``*****('sha1', *, ****, ***)`` **** ****** * ****** **`
paramiko/sftp_file.py:382	`*** **** ** *** **** ********* ** *** (******** ``"sha1"`` **`
paramiko/sftp_server.py:83	`_****_***** = {"sha1": sha1, "***": ***}`

Showing 6 of 12; the remaining 6 are in `evidence/scan.json`.

X25519 — 71 occurrences

File and line	Excerpt
paramiko/kex_curve25519.py:7	`X25519**********,`
paramiko/kex_curve25519.py:8	`X25519*********,`
paramiko/kex_curve25519.py:20	`***** *************:`
paramiko/kex_curve25519.py:30	`X25519**********.********()`
paramiko/kex_curve25519.py:40	`"****'* ********** ****** ***** *** ***** *****"`
paramiko/kex_curve25519.py:45	`****.*** = X25519**********.********()`

Showing 6 of 71; the remaining 65 are in `evidence/scan.json`.

Total hidden from the tables above: 760 occurrences. Nothing was dropped from `evidence/scan.json`; all of them are there.

4. Mapping to the clauses of Articles 6 and 7(4)

For each clause: what the scan contributes and what it does not. The second part is not a caveat; it is the list of work that remains with the entity.

Article 6(1)

Develop, document and implement a policy on encryption and cryptographic controls.

What the scan contributes: The inventory of algorithms actually present in the scanned estate, which the policy must account for.

- algorithms found: 3DES, DH, EC, ECDH, ECDSA, Ed25519, Ed448, MD5, ML-KEM, RSA, SHA1, X25519, private_key

> **What the scan does NOT show:** Whether a policy exists, or whether it is approved.

Article 6(2)

Base that policy on data classification and the ICT risk assessment.

What the scan contributes: Which systems use which algorithms, as an input to the risk assessment.

> **What the scan does NOT show:** The data classification itself.

Article 6(2)(a)

Rules for encryption of data at rest and in transit.

What the scan contributes: Evidence of transport and storage cryptography found in source, CI and infrastructure-as-code.

> **What the scan does NOT show:** Deployed configuration, or data actually at rest.

Article 6(2)(c)

Rules for encryption of internal network connections and of traffic with external parties.

What the scan contributes: TLS and transport settings declared in code and configuration.

> **What the scan does NOT show:** What is negotiated on the live network.

Article 6(2)(d)

Rules on cryptographic key management, as set out in Article 7: correct use, protection and lifecycle of keys.

What the scan contributes: Where key material is generated or referenced in code, and key files found in the tree, as a starting list.

> **What the scan does NOT show:** Key custody, rotation practice, or HSM use.

Article 6(3)

Criteria for selecting cryptographic techniques that take leading practices and standards into account; where these cannot be followed, mitigation and monitoring measures.

What the scan contributes: The algorithms that fall short of leading practice today, each of which needs either replacement or a mitigation measure.

- broken today: 3DES, MD5, SHA1

> **What the scan does NOT show:** The selection criteria, or the mitigation measures taken.

Article 6(4)

Provisions to update or change the cryptographic technology, where necessary, on the basis of developments in cryptanalysis; where this is not possible, mitigation and monitoring measures.

What the scan contributes: The list of algorithms that are already broken, and the list that a cryptographically relevant quantum computer would break. This is the clause under which post-quantum exposure is reported.

- broken today: 3DES, MD5, SHA1
- quantum-vulnerable: DH, EC, ECDH, ECDSA, Ed25519, Ed448, RSA, X25519
- post-quantum: ML-KEM
- private key recoverable from the public key:
    - status: Broken now, whatever the quantum timeline: not a classical key to migrate on the post-quantum schedule. Anyone holding the public key holds the private key; replace it and treat what it signed or protected as exposed.
    - checked: the scanner checked the EC public keys it could locate for a small private scalar and found none; a compressed point, a key inside an encrypted container and a scalar above its bound are not checked
    - keys: *none*
- suggested replacement:
    - status: suggestion, not applied: an algorithm and the standard it comes from; no product or vendor is named
    - per: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths, 2026-01, 23 January 2026 (https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf?__blob=publicationFile), read on 2026-09-27; addressed to: recommendations of a BSI Technical Guideline. Each option names the standard it comes from; a FIPS or NIST SP there means the row is NIST's, which is what a profile falls back to where it sets no rule
    - 3DES:
        - options:
            - use: AES (AES-256 for long-lived data)
            - for: replacement

                - standards: SP 800-131A r2; NIST IR 8547
        - note: Three-key TDEA encryption is disallowed after 2023; decryption for legacy use only.
        - follows: NIST: this profile's document sets no minimum for this algorithm that has been checked, so NIST's row is shown
- DH:
    - options:
        - use: ML-KEM-768 or ML-KEM-1024; FrodoKEM-976 or FrodoKEM-1344; or Classic McEliece (mceliece460896, mceliece6688128, mceliece8192128, or their 'f' variants) -- each in hybrid with a classical key agreement
        - for: key establishment
        - standards: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths
    - note: Hybrid, under this profile: Lattice schemes are recommended only in hybrid form: the guideline 'currently only recommends the hybrid use' of quantum-safe methods. Hash-based signatures may stand alone.
- EC:
    - roles read from the lines: 43 signature, 12 key establishment, 25 undetermined
    - options:
        - use: ML-KEM-768 or ML-KEM-1024; FrodoKEM-976 or FrodoKEM-1344; or Classic McEliece (mceliece460896, mceliece6688128, mceliece8192128, or their 'f' variants) -- each in hybrid with a classical key agreement
        - for: key establishment
        - standards: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths
        - lines with this role: 12
        - lines undetermined: 25
        - use: ML-DSA-65 or ML-DSA-87 in the 'hedged' variant, in hybrid with a classical signature; or a hash-based signature alone
        - for: signatures
        - standards: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths
        - lines with this role: 43
        - lines undetermined: 25
        - use: hash-based signatures: SLH-DSA, or stateful LMS/HSS or XMSS/XMSS^MT, which 'can ... also be used alone (i.e. not in hybrid form)'
        - for: firmware and code signing with a bounded number of

signatures (stateful: the signer must never reuse a state -- and restoring the key from a backup, or copying it to a second module, reuses one; SP 800-208 prohibits exporting the key and, in Sec. 7, splits it across modules as a two-level HSS or XMSS^MT instead)

- standards: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths
- lines with this role: 43
- lines undetermined: 25

- note: Role read from each line of code: 43 signature, 12 key establishment, 25 undetermined -- a key generated or declared on a line is used elsewhere, and one line cannot say how. Hybrid, under this profile: Lattice schemes are recommended only in hybrid form: the guideline 'currently only recommends the hybrid use' of quantum-safe methods. Hash-based signatures may stand alone.

- ECDH:
  - options:
    - use: ML-KEM-768 or ML-KEM-1024; FrodoKEM-976 or FrodoKEM-1344; or Classic McEliece (mceliece460896, mceliece6688128, mceliece8192128, or their 'f' variants) -- each in hybrid with a classical key agreement
    - for: key establishment
    - standards: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths
  - note: Hybrid, under this profile: Lattice schemes are recommended only in hybrid form: the guideline 'currently only recommends the hybrid use' of quantum-safe methods. Hash-based signatures may stand alone.
- ECDSA:
  - options:
    - use: ML-DSA-65 or ML-DSA-87 in the 'hedged' variant, in hybrid with a classical signature; or a hash-based signature alone
    - for: signatures
    - standards: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths
    - use: hash-based signatures: SLH-DSA, or stateful LMS/HSS or XMSS/XMSS^MT, which 'can ... also be used alone (i.e. not in hybrid form)'
    - for: firmware and code signing with a bounded number of signatures (stateful: the signer must never reuse a state -- and restoring the key from a backup, or copying it to a second module, reuses one; SP 800-208 prohibits exporting the key and, in Sec. 7, splits it across modules as a two-level HSS or

XMSS^MT instead)
                - standards: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths
            - note: Hybrid, under this profile: Lattice schemes are recommended only in hybrid form: the guideline 'currently only recommends the hybrid use' of quantum-safe methods. Hash-based signatures may stand alone.
    - Ed25519:
        - options:
            - use: ML-DSA-65 or ML-DSA-87 in the 'hedged' variant, in hybrid with a classical signature; or a hash-based signature alone
            - for: signatures
            - standards: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths
            - use: hash-based signatures: SLH-DSA, or stateful LMS/HSS or XMSS/XMSS^MT, which 'can ... also be used alone (i.e. not in hybrid form)'
            - for: firmware and code signing with a bounded number of signatures (stateful: the signer must never reuse a state -- and restoring the key from a backup, or copying it to a second module, reuses one; SP 800-208 prohibits exporting the key and, in Sec. 7, splits it across modules as a two-level HSS or XMSS^MT instead)
            - standards: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths
        - note: Hybrid, under this profile: Lattice schemes are recommended only in hybrid form: the guideline 'currently only recommends the hybrid use' of quantum-safe methods. Hash-based signatures may stand alone.
    - Ed448:
        - options:
            - use: ML-DSA-65 or ML-DSA-87 in the 'hedged' variant, in hybrid with a classical signature; or a hash-based signature alone
            - for: signatures
            - standards: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths
            - use: hash-based signatures: SLH-DSA, or stateful LMS/HSS or XMSS/XMSS^MT, which 'can ... also be used alone (i.e. not in hybrid form)'
            - for: firmware and code signing with a bounded number of signatures (stateful: the signer must never reuse a state -- and restoring the key from a backup, or copying it to a second module, reuses one; SP 800-208 prohibits exporting the key and, in Sec. 7, splits it across modules as a two-level HSS or

XMSS^MT instead)
                - standards: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths
        - note: Hybrid, under this profile: Lattice schemes are recommended only in hybrid form: the guideline 'currently only recommends the hybrid use' of quantum-safe methods. Hash-based signatures may stand alone.
- MD5:
    - options:
        - use: SHA-256, SHA-512/256, SHA-384 or SHA-512; or SHA3-256, SHA3-384 or SHA3-512 (TR-02102-1, Table 4.1)
        - for: replacement
        - standards: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths
    - note: RFC 6151: not acceptable where collision resistance is required; HMAC-MD5 is less urgent but should go too.
- RSA:
    - roles read from the lines: 103 signature, 0 key establishment, 89 undetermined
    - options:
        - use: ML-KEM-768 or ML-KEM-1024; FrodoKEM-976 or FrodoKEM-1344; or Classic McEliece (mceliece460896, mceliece6688128, mceliece8192128, or their 'f' variants) -- each in hybrid with a classical key agreement
        - for: key establishment
        - standards: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths
        - lines with this role: 0
        - lines undetermined: 89
        - use: ML-DSA-65 or ML-DSA-87 in the 'hedged' variant, in hybrid with a classical signature; or a hash-based signature alone
        - for: signatures
        - standards: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths
        - lines with this role: 103
        - lines undetermined: 89
        - use: hash-based signatures: SLH-DSA, or stateful LMS/HSS or XMSS/XMSS^MT, which 'can ... also be used alone (i.e. not in hybrid form)'
        - for: firmware and code signing with a bounded number of signatures (stateful: the signer must never reuse a state -- and restoring the key from a backup, or copying it to a second

module, reuses one; SP 800-208 prohibits exporting the key and, in Sec. 7, splits it across modules as a two-level HSS or XMSS^MT instead)
                - standards: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths
                - lines with this role: 103
                - lines undetermined: 89
            - note: Role read from each line of code: 103 signature, 0 key establishment, 89 undetermined -- a key generated or declared on a line is used elsewhere, and one line cannot say how. Hybrid, under this profile: Lattice schemes are recommended only in hybrid form: the guideline 'currently only recommends the hybrid use' of quantum-safe methods. Hash-based signatures may stand alone.
    - SHA1:
        - options:
            - use: SHA-256, SHA-512/256, SHA-384 or SHA-512; or SHA3-256, SHA3-384 or SHA3-512 (TR-02102-1, Table 4.1)
            - for: replacement
            - standards: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths
        - note: Disallowed for signature generation; verification allowed for legacy use.
    - X25519:
        - options:
            - use: ML-KEM-768 or ML-KEM-1024; FrodoKEM-976 or FrodoKEM-1344; or Classic McEliece (mceliece460896, mceliece6688128, mceliece8192128, or their 'f' variants) -- each in hybrid with a classical key agreement
            - for: key establishment
            - standards: BSI (Germany): Technical Guideline TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths
        - note: Hybrid, under this profile: Lattice schemes are recommended only in hybrid form: the guideline 'currently only recommends the hybrid use' of quantum-safe methods. Hash-based signatures may stand alone.

> **What the scan does NOT show:** Whether the entity has a process to act on it.

Article 6(5)

Record the mitigation and monitoring measures adopted under paragraphs 3 and 4, with a reasoned explanation.

What the scan contributes: A dated, reproducible record of what was in use at the time of the scan, and which findings would need such a record.

> **What the scan does NOT show:** The measures and the explanation, which are the entity's to write.

Article 7(4)

Keep an up-to-date register of all certificates and certificate-storing devices, at least for ICT assets that support critical or important functions.

What the scan contributes: Certificate and key files found in the scanned tree and the algorithms they declare, as candidate register entries.

- certificate and key files read: 26
- embedded key material found: 24
- register entries: *none*
- withheld: *none*
- register source: no document from another tool was supplied, so the certificate register is not established here

> **What the scan does NOT show:** Certificates in use on live systems, their expiry, and the devices that store them (HSMs, smart cards). Where a register is supplied above, the devices storing the certificates remain unanswered: CycloneDX carries only a class of mechanism (HSM, TPM, software), never an identified device.

5. What remains with the entity

The scan cannot produce any of the following. These tasks remain whatever the result above:

- **Art. 6(1)** — Whether a policy exists, or whether it is approved.
- **Art. 6(2)** — The data classification itself.
- **Art. 6(2)(a)** — Deployed configuration, or data actually at rest.
- **Art. 6(2)(c)** — What is negotiated on the live network.
- **Art. 6(2)(d)** — Key custody, rotation practice, or HSM use.
- **Art. 6(3)** — The selection criteria, or the mitigation measures taken.
- **Art. 6(4)** — Whether the entity has a process to act on it.
- **Art. 6(5)** — The measures and the explanation, which are the entity's to write.
- **Art. 7(4)** — Certificates in use on live systems, their expiry, and the devices that store them (HSMs, smart cards).

6. Reproducibility

The machine-readable inventory (`cbom.json`, CycloneDX 1.6 / ECMA-424) is serialised canonically, and its timestamp is **an input, not a clock reading**. The same input with the same timestamp gives the same bytes and the same digest:

```
SHA-256 of cbom.json: 953c1c56e539b7a15a2ca799363f959e297a25aaf6cf3960bc2e91b5864fa8ee
```

Which time is which. *Scan run at* (2026-11-05T07:22:23+00:00 to 2026-11-05T07:22:24+00:00) is when the scan ran: coverage.window of the scan result, from qrp-mcp's clock while it ran. *Generated* (2026-11-05T07:22:23Z) is the timestamp of `cbom.json` and of this report, clock when this report was made (no --at given); it is an input to the inventory, so it is not when the scan ran.

Measured: two runs over the same input produce byte-identical `cbom.json` and `dora-article6.json`. `evidence/scan.json` and `MANIFEST.json` are not: they record when the scan started and finished (`coverage.window`), read from the clock while it ran. The PDF itself is **not** byte-identical (the converter stamps a creation date and a document identifier), but **its text layer is identical**. The content is reproducible; the wrapper is not.

7. Who may use this report

Prepared for: **not stated** — no client was named when this report was made. The client may share it with its auditor, its supervisor and its insurers. It states what was read and what was found; it is not an auditor's opinion, and it does not assess whether the client's policies are adequate. Elkan Electronic's responsibility is as set out in its engagement with the client.