International Civil Aviation Organization

WORKING PAPER

TAG/TRIP/5-WP/18
27/09/25
English only

TECHNICAL ADVISORY GROUP ON THE TRAVELLER IDENTIFICATION PROGRAMME (TAG/TRIP)

FIFTH MEETING

Montréal, 12 to 14 November 2025

Agenda Item 2: Activities of the New Technologies Working Group (NTWG)

QUANTUM SAFE MECHANISMS FOR THE DOCUMENT ISSUING PKI AND PASSIVE AUTHENTICATION

(Presented by ISO WG3/Task Force 5 on behalf of NTWG)

SUMMARY

The rise of quantum computing poses a significant threat to the cryptographic security mechanisms of electronic Machine Readable Travel Documents (eMRTDs), in particular mechanisms relying on RSA and ECC algorithms. In response, ICAO's New Technologies Working Group (NTWG) and the ISO WG3 initiated efforts to assess and mitigate these risks. These efforts included investigations to evaluate threats imposed by quantum computers and to explore migration strategies to quantum-safe alternatives.

Two key Information Papers were issued, addressing risks to existing security mechanisms such as Passive Authentication (PA), which is especially vulnerable, and identifying options for quantum-safe solutions including hybrid cryptographic schemes. The NTWG recommends the creation of a new work item to develop technical specifications for transitioning the Document Issuing PKI and Passive Authentication to quantum-safe mechanisms.

The information papers are in Appendix A and Appendix B consecutively.

Action by the TAG/TRIP is in paragraph 5.

Element of the ICAO TRIP Strategy:	MRTDs *Document Issuance/Control* *Inspection Systems/Tools*
Financial implications:	
References:	

1. INTRODUCTION

1.1 The International Civil Aviation Organization (ICAO) standard for electronic Machine Readable Travel Documents (eMRTD) specifies cryptographic mechanisms to strengthen the security of travel documents. These mechanisms rely heavily on public key cryptography using Rivest–Shamir–Adleman (RSA), Finite Field and Elliptic Curve (ECC) cryptography. A quantum computer, however, could break these cryptographic mechanisms; consequently, undermining the security mechanisms of current eMRTDs.

1.2 This paper summarizes the activities of the NTWG, in which they investigated the impact of quantum computers on the security mechanisms of travel documents, as well as possible mitigation strategies.

1.3 Based on this, the creation of a new work item is recommended that aims at drafting a Technical Specification for the implementation of quantum-safe mechanisms.

2. BACKGROUND

2.1 In order to develop mitigation strategies against emerging threats posed by quantum computers, the NTWG created the PQC subgroup, in November 22.

2.2 The NTWG's PQC subgroup tasked ISO WG3 to compile and assess information on the current developments of quantum computing and their potential impact on the cryptographic security mechanisms employed in electronic Machine-Readable Travel Documents (eMRTDs).

2.3 In response, WG3 established the Crypto Agility Subgroup (CASG). The subgroup's mandate was to review existing research on quantum computers, evaluate potential risks for currently implemented security mechanisms, and explore migration strategies to quantum-safe mechanisms for eMRTD security.

2.4 The Crypto Agility Subgroup has reviewed multiple sources, including academic research, industry assessments, and technical standards from international standardization bodies.

2.5 The CASG's work resulted in two Information Papers issued by ISO WG3: *Developments regarding Cryptographic Agility and Post Quantum Cryptography for eMRTD*; and *Quantum safe mechanisms for the Document*. The Information Papers are in the Appendix A and Appendix B.

3. INFORMATION PAPER ON "DEVELOPMENTS REGARDING CRYPTOGRAPHIC AGILITY AND POST QUANTUM CRYPTOGRAPHY FOR EMRTDS"

3.1 This Information Paper has revisited all cryptographic mechanisms currently implemented in eMRTDs, including:

a) Passive Authentication (PA) – verification of the authenticity and integrity of data stored on the eMRTD's chip, in barcodes, or DTC-VCs;

b) Chip Authentication (CA) and Active Authentication (AA) – protection against cloning or substitution of the eMRTD's chip;

c) PACE – establishment of secure communication between an inspection system and the eMRTD's chip to provide protection against sniffing and eavesdropping; and

d) Terminal Authentication (TA) – access control for sensitive biometric data.

3.2 The threat to Passive Authentication (PA), which relies on digital signatures from the Document Issuing Public Key Infrastructure (PKI), has been identified to be the most systemic, i.e., an attacker in possession of a quantum computer could calculate the private keys of an Issuing State's signing authority (CSCA) and then fraudulently issue eMRTDs that would appear cryptographically genuine.

3.3 In conclusion of this outcome, the NTWG has tasked ISO WG3 to identify options for the migration of the Document Issuing PKI and Passive Authentication to quantum-safe mechanism.

3.4 The Information Paper on "Developments regarding Cryptographic Agility and Post Quantum Cryptography for eMRTDs" will be continuously updated to reflect the current status of potential risks to cryptographic security mechanisms and potential mitigation options.

4. INFORMATION PAPER ON "QUANTUM SAFE MECHANISMS FOR THE DOCUMENT ISSUING PKI AND PASSIVE AUTHENTICATION"

4.1 This Information Paper investigates potential options for migrating the security mechanisms that ensure the authenticity and integrity of data stored by eMRTDs and their impacts on security, readiness, and backward compatibility. Hereby, the following aspects have been addressed:

a) quantum-safe signature algorithms that are currently standardized, including all algorithms currently standardized by ISO and those from NIST's Post-Quantum Cryptography Standardization project;

b) options for issuing certificates in the document-issuing PKI that support quantum safe algorithms or a hybrid scheme that combines classical (RSA/ECC) with quantum-safe algorithms; and

c) options to store digital certificates and signatures on the chip of an eMRTD to accommodate quantum-safe signatures without disrupting interoperability.

4.2 The paper furthermore analyses strategies for the migration of the PKI to quantum-safe mechanisms while simultaneously considering the recommended lifespan of up to 10 years of currently issued documents, and balancing security requirements with the long validity period of existing travel documents.

5. ACTION BY THE TAG/TRIP

5.1 The TAG/TRIP is invited to:

a) take note of the information provided by the Information Papers *Developments regarding Cryptographic Agility and Post Quantum Cryptography for eMRTDs* and *Quantum-safe mechanisms for the Document Issuing PKI and Passive Authentication*;

b) issue a State Letter asking countries to participate in a survey regarding their opinion on the next steps for standardizing quantum safe mechanisms for the Document Issuing PKI and Passive Authentication and to circulate the information provided by this working paper through complementary informal channels (NTWG, ICBWG, PKD Board) to encourage broader engagement;

c) approve the continued work of the NTWG PQC Subgroup;

d) approve the creation of a new work item aimed at producing a technical specification on the implementation of quantum-safe mechanisms in the Document Issuing PKI and Passive Authentication; and

e) encourage WG3 to establish and maintain liaisons with relevant international standardization bodies and working groups in order to identify and develop missing building blocks necessary for this transition.

— — — — — — — —

Appendix A

Information Paper

Developments regarding Cryptographic Agility and Post Quantum Cryptography for eMRTDs

Version – 0.07

Date – Dec 19, 2024

ISO/IEC JTC 1/SC 17/WG 3 "Traveller identification"
Convenorship: **SSC**
Convenor: **Rajeshkumar Raja Mr**

TF5_IP - CASG_Developments regarding CAPQC_v0.07

Document type	Related content	Document date	Expected action
Project / Draft		2024-12-23	**COMMENT/REPLY** by 2025-03-14

MACHINE READABLE TRAVEL DOCUMENTS

Information Paper

Developments regarding Cryptographic Agility and Post Quantum Cryptography for eMRTDs

Version – 0.07
Date – Dec 19, 2024

ISO/IEC JTC1 SC17 WG3/TF5
FOR THE
NEW TECHNOLOGIES WORKING GROUP

Author : ISO/IEC JTC1 SC17 WG3/TF5

Release Control

Release	Date	Description
0.01	Nov 2023	Initial Draft
0.03	Jan 2024	Included contributions from Thales, Idemia & Veridos
0.04	Feb 2024	Incorporation of changes agreed during 3rd CASG meeting on Jan 18
0.05	Mar 2024	Comment resolution from v0.04 and new chapter 4
0.06	Nov 2024	New input for chapter 2 (development status of quantum computers)
0.07	Dec 2024	Added summary and minor editorial changes

Felix Bleckmann	Germany	BSI
Michael Hoppe	Germany	BSI
Guido Frank	Germany	BSI
Thomas Thaler	Switzerland	Thales
Alban Feraud	France	IDEMIA
Gilles Pauzie	France	Thales
Mike Neumann	USA	Thales
Jens Urmann	Germany	Veridos
Holger Funke	Germany	secunet

Table of contents

1. Executive Summary

The International Civil Aviation Organizations (ICAO) standard for electronic Machine Readable Travel Documents (eMRTD) relies heavily on public key cryptography using Rivest–Shamir–Adleman (RSA), Finite Field and Elliptic Curve (ECC) cryptography. These algorithms and their implementations are the state of the art and prove reliable and robust against classical computing attacks.

A Quantum Computer, however, introduces the properties and mathematics of quantum physics to computing. Combined with Peter Shor's algorithms which theoretically proved in 1994 that a quantum computer could break RSA, DH and ECC in very little time; consequently, undermining the current security mechanisms of eMRTDs.

On Q-Day, the day that a cryptographically relevant quantum computer (CRQC) exists and therefore Shor and Grover's mathematics are no longer theory, classical public key cryptography becomes insolvent [HE2024] as a security mechanism.

The questions are therefore "What is *most* at risk on Q-Day?" and "What is the Post-Quantum (PQ) Solution for each vulnerability?" The table below addresses only the most critical in terms of urgency and impact.

Table 1: Overview of CRQC primary threats against current eMRTD implementations

eMRTD Principle	**Doc 9303 Mechanisms on Q-Day**	**Threat Severity[1]**
Authenticity of stored data	Passive Authentication. PKI of signing authorities and the PKD are fully compromised. SOD signatures cannot be trusted.	Highest, Global
Anti-cloning	Protection against cloning or substitution of the eMRTD's chip would be no longer available.	High, Document
Privacy in communications	PACE. Chip/terminal communication is no longer protected from sniffing and/or eavesdropping. Terminal Authentication. Protection of highly sensitive biometric data on the chip (fingerprints or iris) is lost.	High, Individual Document or State (TA)

The threat to Passive Authentication is universally agreed to be the most systemic, e.g. a CRQC-possessing attacker could calculate the private keys of an issuing State/signing authority in the PKD certificates and then fraudulently issue eMRTDs that would appear cryptographically genuine.

[1] Threat Severity is the impact on eMRTDs with a CRQC present. It does not assess when one would be available. 'Global' is the entire issuance of a State. 'Document' is an individual document.

2. Scope

The scope of the present Information Paper is to keep ICAO NTWG updated on the developments regarding crypto agility and post quantum cryptography. Furthermore, the paper shall provide options for devising guiding core principles for a possible future development of technical specifications.

This document is intended to be a “Living Document” to chronicle the challenges and advancements in Cryptographic Agility and Post Quantum Cryptography for eMRTDs. This includes the critical next steps and decisions made as ICAO NTWG and SC17/WG3 work through each priority relating to:

- Identification of the next critical threat to address
- Standardization of Updates on eMRTD, Inspection System, and Issuance Solution Quantum Readiness
- Advancements in Hybrid solutions that can be Cryptographically Agile.

For example, Cryptographic Agility [NIST SP800-131A] has thus far addressed incremental improvements to classical cryptographic key lengths and software vulnerabilities, however, PQC will bring additional complexity to this concept.

3. Status Quo of Cryptographically Relevant Quantum Computers (CRQC)

Quantum-computers work with so-called qubits instead of bits and use the quantum mechanical effects of superposition and entanglement. On the one hand, this means that some problems that would traditionally require a lot of memory can be solved with relatively few qubits. On the other hand, the use of these effects leads to an intrinsic parallelization of some computations and thus to an acceleration that would not be possible with conventional computers.

A quantum algorithm leverages precisely this parallelization. The best-known quantum algorithms include the search algorithm of Lov Grover (1996) and the algorithms of Peter Shor (1994), which can be used to factorize integers and calculate discrete logarithms. The latter algorithms in particular have the potential to break today's public key cryptosystems such as RSA or Elliptic Curve Cryptography.

3.1 Shor’s algorithms

The security of public key cryptography relies heavily on assumptions about the complexity of certain mathematical problems. In the case of RSA, this involves the problem of efficiently decomposing large natural numbers into their prime factors, and in case of elliptic curve cryptography (ECC) or the Diffie-Hellmann-Key-Exchange (Finite-Field-DH/ECDH), the discrete logarithm problems. For both problems no algorithm is known that allows to solve them efficiently, which would imply that it solves the problem within a runtime that is polynomial in the length of the number to be factorized or to find its discrete logarithm.

Shor has proposed two quantum algorithms that can solve both problems efficiently, i.e. that runs in polynomial time. Consequently, the existence of a cryptographically relevant quantum computer would allow one to break the RSA, ECC and Finite-Field-DH/ECDH public-key cryptography schemes.

3.2 Grover's algorithm

Grover's algorithm is a quantum algorithm that finds a searched element in an unsorted list of N elements with high probability after √N steps. With classical algorithms, finding a searched element can only be guaranteed after N steps. Thus, the algorithm provides at most a quadratic speedup over classical algorithms for unstructured search, but does not provide polynomial runtime.

Applied to cryptography Grover's algorithm could facilitate searching a secret key of a symmetric cipher within its key space. With keys of 128 bits in length, the key space can theoretically be searched in around 2^{64} quantum operations using Grover's algorithm. However, already with a key length of 256 bits, an order of magnitude of 2^{128} quantum operations is required, which is not considered feasible today. Furthermore, from a practical point of view ,the implementation of a Grover search on quantum computers places high demands on the quantum computer circuits.

In consequence, in order to maintain a security level of 120 bit in symmetric cryptography assuming the availability of CRQCs, it suffices to use keys with at least 240 bit for ideal symmetric ciphers.

3.3 Resource estimation for cryptanalysis of classic cryptographic algorithms using a CRQC

The goal of this and the next section is to give an impression of the requirements and the current state of quantum computing from the perspective of its applicability to cryptanalysis and to give an overview of the forecast of the future availability of a CRQC.

A common computational architecture used to estimate the performance of quantum algorithms consists of the number of logical qubits required by the algorithm, and the size of the quantum circuit that must be computed on these qubits. The circuit size is defined as the number of gates that are computed and can be translated into the time necessary to finish the computation given the time required to calculate a single gate in the circuit. Note that this model is a high-level abstraction of a quantum computation. In practice, different quantum gates in the circuit require different amounts of resources [NC20, FMMC12]. Furthermore, physical qubits are not perfect, are error-prone, and cannot be used for large computations. Therefore, the estimates in this section give only the number of so-called logical qubits, which are assumed to be perfect and ignore physical error rates and decoherence times. A more detailed discussion of physical and logical qubits is given in section 2.4.

The data in Table 2 is based on the results of Amy et al. [AMG+16], who estimate the cost of a Grover-based preimage attack on two specific variants of the SHA family of hash functions. In particular, they estimate the SHA-256 [NIS15a] and SHA3-256 [NIS15b] hash functions. The circuit size is rounded to the nearest power of two.

Table 2: Resource estimation for attacking symmetric primitives using Grover's algorithm

Algorithm	Attack	Circuit Size	Number of logical qubits
AES-128	Key recovery	2^{83}	10001
AES-192		2^{116}	19505
AES-256		2^{149}	20529
SHA-256	Preimage	2^{150}	2402
SHA3-256		2^{153}	3200

Estimates listed in Table 3 for factorization and the elliptic curve discrete logarithm problem are taken from [RNSL17]. Significant factors in the concrete size of the quantum circuit are

the circuits that implement modular arithmetic and elliptic curve operations. In [RNSL17], the authors study the widely used NIST standard curves P-192, P-224, P-256, P-384, and P-521 [NIST SP800-186]. They use a quantum circuit designed by Takahashi et al. [TTK10] to implement modular addition and doubling. To realize modular multiplication, the authors implement Montgomery multiplication [Mon85].

Table 3: Resource estimation for attacking Factorization and ECDLP using Shor's algorithm

Problem	Instance Size	Circuit Size	Number of logical qubits
Elliptic Curve DLP	160	2^{32}	1466
	224	2^{36}	2042
	256	2^{37}	2330
	384	2^{39}	3484
	512	2^{40}	4719
Factorization	1024	2^{37}	2050
	2048	2^{40}	4098
	3072	2^{42}	6146
	7680	2^{46}	15362

By taking the base-two logarithm of the circuit size, one can derive the bit security of the scheme, which is the equivalent expected number of operations to guess a random sequence of bits. For example, the bit security of AES-256 is 149, but a factorization instance of size 1024 has a bit security of only 37 bits. In general, a bit security less than 80-bits is considered to be insecure. Hence, all instances in Table 2 should be considered as broken. On the other hand, AES-128 and AES-192 are below the 128-bit security level, which is often the desired minimum security level because it leaves a reasonably large margin for potential future improvements in cryptanalysis.

3.4 State of the art of quantum computing

As mentioned earlier, quantum computers use the physical effect of superposition to gain a computational advantage over classical computation. In particular, a qubit can be set in a state of superposition, but any interference with the environment, such as an attempt to read the qubit's state, collapses the superposition to a classical bit value. Ideally, the only interference with the qubit is intentional attempts to read its state. In practice, a qubit may also interact with other particles that are out of our control. If a qubit's superposition collapses before the computation terminates, the readout may be erroneous. Thereby the quality of a physical qubit is often characterized by its decoherence time, which is the expected time for a qubit's superposition to unintentionally collapse, and its error rate, which is the amount of error introduced during the computation.

There are several ways to physically realize qubits and manipulate them to build a quantum computer. Currently, the most promising and explored techniques are superconducting qubits, trapped ions, neutral atoms, semiconductor quantum dots, and photonics.

- **Superconducting qubits**: A superconducting qubit is an electronic circuit in which energy levels, such as charge or flux, take on quantized values. The energy levels are used to represent different quantum states. The quantum chip must be cooled to cryogenic temperatures so that thermal energy does not disturb the qubit's state. The qubit is controlled through electromagnetic coupling with microwave pulses.
- **Trapped ions**: In a trapped ion system, a dynamic electric field is used to isolate and hold a single electrically charged atom (ion) in place. A trapped ion can transition between states by interacting with a laser pulse or a microwave field. Trapped ions have relatively low error rates and coherence times of commercially available quantum systems, but gate times are longer, requiring tens of microseconds.

- **Neutral atoms**: This technology has many similarities to trapped ions. Atoms are trapped in a magneto-optical field to implement qubits. Unlike trapped ions, qubits are encoded in the nuclear spin of the trapped atom. Operations are performed by applying lasers to the qubits.
- **Photonics**: Qubits in photonic systems are implemented by associating two waveguides for each qubit. The presence of a photon in any of the two waveguides represents the quantum state. The qubits are manipulated using beam splitters and phase shifters. Measurements are performed by single-photon detectors. The photons themselves can operate at room temperature, and do not interact with each other, reducing the error rate.
- **Semiconductor quantum dots**: Quantum dots are areas where single electrons can be trapped so that their spin degree of freedom can be used to represent a quantum state. The computation is performed by applying electrical voltages to gates surrounding the quantum dot to control the number of electrons confined within it. Quantum dots have demonstrated relatively long coherence times and offer higher qubit densities than other approaches.

Table 4 gives a possibly incomplete overview of the quantum devices with the highest number of physical qubits available at the time of writing this document. Note that the number of bits refers to physical qubits, and the table does not specify decoherence times or error rates as these may vary even within the same physical technology used or the data is not public. Furthermore, with a few exceptions, it is not clear how many logical qubits may be implemented on these machines. Nevertheless, the table gives us an intuition of the current state of the art regarding the availability of physical qubits.

Table 4: Overview of the current state-of-the-art quantum computing devices

Technology	Company	Number of Physical Qubits	Year
Superconducting	IBM	1121	2023
	Google	53	2019
	Quantware	64	2023
	Intel	49	2018
	IQM	20	2023
	Quantware	64	2023
	Rigetti	84	2023
	RIKEN	53	2023
	CAS	504	2024
Trapped ion	Alpine Quantum Technologies	24	2021
	IonQ	32	2022
	Quantinuum	56	2023
Neutral Atoms	Atom Computing	1180	2023
	M Squared Lazers	200	2022
Photonics	Xanadu	216	2022
Semiconductor	Intel	12	2023
	QuTech at TU Delft	6	2022

Because qubits are prone to errors that increase with time and computation, it's challenging to implement useful algorithms. To create logical qubits that are more resistant to errors, quantum computers rely on quantum error correcting codes. These codes require many physical qubits to implement a single logical qubit. However, it's important to remember that error-correcting codes only reduce errors to tolerable levels. The technical realization of logical qubits is still not perfect, even with the use of these codes.

Given the technical challenges of performing quantum computation in practice, it is fair to ask what are the physical requirements for a quantum computer to run a cryptographically relevant quantum computation. To demonstrate how real physical systems would perform, and how many physical qubits are needed, Table 5 recalls the results of Gidney and Ekerå [GE21]. The authors give a time and space estimate for breaking factorization and finite field discrete logarithm problems, making realistic assumptions on the physical hardware and realizing logical qubits using surface error-correcting codes [FMMC12]. For example, they estimate that factoring 2048-bit numbers can take on an average about 8 hours and 20 million physical qubits. Their calculations assume an error rate of 0.001, that is one error per 1000 operations which is a typical error rate for realistic quantum setups. To support the required amount of computation they use surface codes parameterized to use 1568 physical qubits to implement a single logical qubit. It is worth mentioning, that despite the use of error-correcting codes, the probability of an error occurring in the computation is not nullified. For example, when factoring a 2048-bit number there is a 31% chance that an error will occur and the computation will have to be repeated. Table 5 already accounts for such an event and gives the approximate average time to expect. As noted in [GE21] the probability can be reduced by choosing different parameters of the error correcting code, but a significant increase in the number of physical qubits.

Table 5: Resource estimation according to [GE21]

Problem	Instance Size	Average Time	Physical Mega Qubits
Finite Field DLP	1024	1.8	9.7
	2048	7.0	26
	3072	16	38
	4096	29	55
	8192	120	140
Factorization	1024	1.3	9.7
	2048	8	20
	3072	12	38
	8192	86	140

Error-correcting codes to reduce logical qubit error rates, require multiple physical qubits to form a single logical qubit. Previously, it was uncertain whether this scaling would be feasible for larger quantum chips. To support larger computations and lower error rates, more physical qubits per logical qubit are required. However, larger systems are more vulnerable to environmental noise and cross-talk, which can increase the error rate and potentially offset the gains from error correction.

Recent results [GQAI23,BCG+24,SRB+24] showed that applications of carefully designed error-correcting codes are scalable and can be applied to real quantum setups with a larger number of qubits. In [SRB+24] the authors present experiments on a trapped-ion processor and implemented one logical qubit using seven physical qubits with error rates 9.8 to 500 times lower than the physical error rate, and two logical qubits using 12 physical qubits with error rates 4.7 to 800 times lower. In [BCG+24] the authors implement a logical qubit on a superconducting qubit device, using a surface code comprising 49 physical qubits and compared it against a logical qubit composed of 17 qubits, and showed better error correction capabilities of the 49 physical qubit system in practice. In [GQAI23] the authors present new versions of Bivariate Bicycle LDPC codes, that significantly outperform surface codes and require a lower number of physical qubits to achieve the same error rate. In particular, the authors implement 12 logical qubits capable of making nearly 1 million operations using only 288 physical qubits. This results in a logical to physical qubit ratio of 1:24. In comparison, one would need approximately 2904 physical qubits with previous surface codes [FMMC12]. In other words, the error-correcting codes achieve a logical to physical qubit ratio of 1:24 for the Bivariate Bicycle LDPC code variant, compared to a 1:242

ratio for surface codes. Note, however, that the logical-to-physical qubit ratio does not necessarily scale linearly with the number of qubits. In fact, for surface codes, the number of physical qubits grows quadratically with the number of logical qubits making their application in practical systems challenging. Unfortunately, there is no exact closed-form expression for the code distance of a Bivariate Bicycle LDPC code, hence it is hard to say how it scales with the number of qubits.

3.4.1 Current challenges in the development of quantum computers

While there has been remarkable progress in the field of quantum computing, there are still major challenges to overcome to build a CRQC. The most relevant challenges from a cryptoanalytic point of view are:

- **Scaling quantum processors**: As shown above, the current number of (physical) bits is not sufficient to perform cryptographically relevant operations. Recall that Table 3 lists the number of physical qubits. Due to decoherence and errors, quantum devices must use quantum error-correcting codes. The application of such codes comes at the cost of implementing a single logical qubit with a number of physical qubits.
- **Quantum error correction**: As mentioned above, uncontrolled decoherence is a major obstacle to quantum computing. However, significant progress has recently been made in designing concretely efficient quantum error-correcting codes that reduce the number of physical bits needed to implement a logical qubit.
- **Scaling hardware**: Technologies such as superconducting qubits require that the bits be kept at cryogenic temperatures. This cooling is essential for quantum computation, as higher temperatures increase decoherence times and the error rates. However, providing such cooling is a significant cost factor. Researchers are currently working on methods and hardware to reduce the cost of operation.

3.4.2 Forecast on the availability of CRQC

Unfortunately, it is difficult to estimate when a CRQC will be available, and many past predictions have been inaccurate. The difficulty in making a reliable estimate is due to the fact that advances in quantum computing depend heavily on technical breakthroughs in various fields, such as quantum error correction, physics, and manufacturing. Such breakthroughs are notoriously difficult to predict. Nevertheless, some prediction models can provide a rough intuition. For example, Mosca [Mos15] estimates a 1/7 chance that RSA-2048 will be broken by 2026 and a 1/2 chance by 2031. However, the prediction seems unlikely given the data that we have today. Other sources [PQSt23] often give a range between one and two decades. However, it is unclear where these numbers come from. To put a little more confidence behind these forecasts, this section gives an overview of the growth of qubit numbers in recent years and an attempt to interpolate the growth.

One way to estimate the number of transistors in CPUs is Moore's law. A proposed quantum analogue to Moore's law is Rose's law for estimating the number of physical qubits on a chip. Rose's law, named after the co-founder of D-Wave, predicts that the number of physical qubits on a chip will double every 18 months or quadruple every three years. To help validate Rose's law, Table 4 shows the number of qubits for devices offered by D-Wave's quantum annealers and IBM's superconducting qubit devices. Surprisingly, the forecast perfectly aligns with the number of qubits of D-Wave's devices available between 2011 and 2020. However, note that this is for quantum annealers and not general-purpose quantum computers. If we take the number of qubits of devices offered by IBM in the year between 2016 and 2023, we get an average growth rate of 2.33 per year. This gives us a growth by a factor of around 10 every three years compared to a quadruple growth in the case of Rose's law. Note, however, that the growth rate is not stable and varies greatly from year to year,

depending on major technological breakthroughs. It is also important to keep in mind, that the growth rate concerns physical qubits, and as discussed earlier, the number of useful logical qubits realizable on such devices does not necessarily have to grow at the same rate.

Table 6: Number of physical qubits for D-Wave's quantum annealers and IBM's superconducting quantum computers

	2011	2013	2015	2016	2017	2018	2019	2020	2021	2022	2023
DWave	128	512	1152	-	2048	-	-	5760	-	-	
IBM	-	-	-	5	17	20	53	-	127	433	1121

According to Rose's law, and starting with 1000 qubits in 2023 (a little less than IBM's device to simplify calculations), one may expect to have a quantum chip with about 16 mega physical qubits around 2044. This would be enough to break 1024-bit instances of elliptic curve discrete logarithm or factorization problems when following the assumptions from [GE21] on error rate and error-correcting codes. Assuming a 9-fold improvement in error correction over [GE21], such instances could be solved around 2038. A more pessimistic picture is given by the speedup rate of IBM superconducting devices. According to such a growth rate, i.e., 2.33 per year, one could expect a quantum device capable of breaking 1024-bit ECDLP or factoring problems by 2038 under the assumptions from [GE21]. However, assuming only a 4-fold improvement in the error correction of such a massive number of physical qubits, the time would be 2035.

Either way, one should be prepared for a potentially operational CRQC in the next decade or two. The numbers seem to be in line with the general intent [PQSt23]. That is around the years 2035 and 2045. Also [NIST-IR8547] establishes the year 2035 as the primary target for completing the migration to quantum safe cryptographic algorithms. Furthermore, the number of logical or physical qubits is reported for device prototypes, and more work will be needed to offer an enterprise quantum computing product. Nevertheless, the history of sudden breakthroughs in error-correcting codes shows that rapid advances in quantum computing are possible and should be considered when assessing the risk to classical cryptographic systems.

4. Analysis of threats to current cryptographic systems within eMRTD

This chapter provides on a comprehensive examination of the emerging threats posed by CRQCs to cryptographic algorithms implemented in eMRTDs. This includes an analysis of the vulnerabilities inherent in these algorithms when confronted with quantum algorithms and an exploration of the potential ramifications of their compromise. Moreover, the assets at risk within eMRTDs are identified, and the impact of algorithmic breaches on their security and functionality is assessed.

Each cryptographic algorithm is investigated in one section, which are sorted according to the risk if the respective protocol gets compromised. An overview is given in the following table. In summary, essentially all algorithms based on asymmetric cryptography are vulnerable, while symmetric or hash-based schemes remain secure by doubling the key length.

Table 7: Overview of the risks emerging from CRQCs for eMRTD protocols

Cryptographic Protocol	Impact of a cryptographically relevant quantum computer on current protocol implementation.	Threat Severity[2]
Passive Authentication	Cryptographic protection of an electronic travel document would be entirely compromised. Both the document issuing PKI (CSCA & Document/SealSigner) as well as the data stored by an eMRTD would be affected.	High
Chip/Active Authentication	Protection against cloning or substitution of the eMRTD's chip would be no longer available.	Medium
PACE	The inspection procedure of an eMRTD's chip would no longer be protected from sniffing and/or eavesdropping.	Medium
Terminal Authentication	Protection of highly sensitive biometric data on a chip (fingerprints or iris) would no longer be available.	Medium
Secure Messaging	None (if a sufficient key-length is used)	None

4.1 Authentication of data (Passive Authentication)

Passive Authentication ensures the authenticity and integrity of the Document Security Object (SOD) and LDS stored in the IC of an eMRTD or of the contents of a digital seal by means of digital signatures. It does not prevent exact copying data or chip substitution.

The eMRTD Public Key Infrastructure (PKI) has been deployed to create, manage and distribute digital certificates, that enable the creation and subsequent verification of digital signatures on the before mentioned eMRTD data objects to ensure the signed data is authentic and has not been modified. Currently, digital signatures may be implemented using one of the three algorithms, RSA, DSA or ECDSA.

Quantum computing will put at risk Passive Authentication thanks to Shor's algorithms, which can break all of the three currently allowed signature algorithms with a sufficiently powerful CRQC. In consequence the authenticity and integrity of any data stored on eMRTDs is no more protected by cryptographic mechanisms.

Furthermore, it is important to emphasize that Shor's algorithms allow retrieving the private key from the public key or the certificate which is publicly available (e.g. from the ICAO PKD). This would break the trust-model of the entire document issuing Public Key Infrastructure (PKI) due to the compromise of the signing keys for both the CSCA and Document / Seal Signers which would allow to forge the digital part of any eMRTD.

In summary, a CRQC will jeopardize the followings assets and security properties which are instrumental for the trust in eMRTDs:

Table 8: Assets for Passive Authentication

Assets	Security properties at risks
LDS stored in an eMRTD	Authenticity and integrity
Content of a Visible Digital Seal (e.g. signed barcode printed on a Visa, ETD, …)	Authenticity and integrity
Signed data within a DTC-VC	Authenticity and integrity

[2] The Threat Severity only indicates the impact on eMRTDs if a CRQC is present and does not assess if and when such a quantum computer is available.

Private key of the issuing authority (CSCA and DS/Seal Signer key) Note: the private key can be retrieved from the public key which are made available	Confidentiality, authenticity, integrity and availability

This will result in the following risks in the following conditions:

Table 9: Risk Analysis for Passive Authentication Assets

Assets	Risk	Conditions
LDS stored in an eMRTD	Creation of a forged LDS and thus: • impersonation of someone else's identity (combined with someone else's portrait); • Creation of a fake identity;	This risk applies where the authenticity of the physical document is not verified (e.g. no visual inspection).
Content of a Visible Digital Seal (e.g. signed barcode on a Visa, ETD, …)	Creation of a forged barcode and thus: • impersonation of someone else's identity (combined with someone else's portrait); • Creation of a fake identity;	This risk always holds
DTC-VC	Creation of a forged DTC-VC and thus: • impersonation of someone else's identity (combined with someone else's portrait); • Creation of a fake identity; Transferability of the DTC-VC which may be bound to any other – uncontrolled – device (DTC-PC)	This risk always holds
Private key of the issuing authority (CSCA and DS/Seal Signer key)	If the compromise of a private key is made public, denial of service of the whole infrastructure by annihilating the trust of stakeholders. If the compromise of a private key is not made public, creation of forged LDS, barcode and DTC-VC and CRLs	This risk always holds

Conclusion: Passive authentication can be compromised by a sufficiently powerful CRQC, which would nullify any cryptographic protection of a travel document. The security of any travel document would fall back to their security properties that could be analysed by visual inspection and forensics.

For example, the security of an eMRTD would fall back to that of a MRTD (i.e. an eMRTD without an IC). Any fully electronic travel document like DTCs or VDSs on plain paper would no longer offer any protection against counterfeiting.

Furthermore, the private keys from the document issuing authorities can be revealed which would break any trust in the whole PKI infrastructure.

Threat severity: The loss of the protection offered by Passive Authentication constitutes the greatest risk among the threats analyzed in this paper.

4.2 Prevention against cloning (Chip/Active Authentication)

Chip Authentication (CA), PACE-CAM and Active Authentication (AA) are mechanisms that allow an Inspection System to validate that the data read electronically from a chip actually originates from a genuine chip that has been produced, personalized and issued on behalf of the issuing State or organization. The mechanisms offer protection against duplication of chips or chip substitution within eMRTDs or the physical component of a DTC (DTC-PC).

Cryptographically, CA and PACE-CAM rely on the DH/ECDH algorithm while AA is based on RSA or ECDSA. Quantum computing will put at risk in the future the anti-cloning mechanisms thanks to Shor's algorithms, which can break all of the mentioned cryptographic algorithms with a sufficiently powerful CRQC. These algorithms allow retrieving the private key from the public key or the certificate which is made public.
This will jeopardize the following assets and security properties which are instrumental for the trust in eMRTDs:

Table 10: Assets for CA/AA

Assets	Security properties at risks
AA private key of the eMRTD	Confidentiality, authenticity and integrity
CA private key of the eMRTD	Confidentiality, authenticity and integrity
Private key of the DTC-PC	Confidentiality, authenticity and integrity
PACE-CAM private key of the eMRTD	Confidentiality, authenticity and integrity

This will result in the following risks in the following conditions:

Table 11: Risk Analysis for CA/AA Assets

Assets	Risk	Conditions
AA, CA or PACE-CAM private key of the eMRTD	The authenticity of the eMRTD cannot be verified electronically	This may be countered by a visual or forensic inspection of the eMRTD (shape and physical security features)
AA, CA or PACE-CAM private key of the DTC-PC	The authenticity of the DTC-PC cannot be verified electronically	This risk always holds

Conclusion: CA, PACE-CAM and AA can be compromised by a sufficiently powerful CRQC which would nullify the protection against cloning of a travel document. The security of any travel document would fall back to their equivalent providing only Passive Authentication.

Threat severity: After a compromise of CA, PACE-CAM or AA, chips would no longer be protected against duplication and their authenticity could no more be validated. Still, the authenticity and integrity of the data protected by Passive Authentication would remain intact (if measures against CRQC attacks on Passive Authentication are already in place).

Since Chip Authentication or PACE-CAM present some advantages over Active Authentication (e.g. prevention of challenge semantics, establishment of strong session keys), these properties should also be supported by a future CRQC-proof protocol.

4.3 Access Control (PACE)

The PACE protocol provides password-based authentication between the eMRTD chip and the inspection system and derives strong session keys for Secure Messaging. Hereby, PACE offers protection against offline attacks against a transcript of the session independent of the strength of the password.

PACE strongly relies on the DH or ECDH key agreement algorithms, which can be broken by a sufficiently powerful CRQC using Shor's algorithms and thus allow recovering the session keys derived by the chip and the IS. If the keys are known, also the PACE password could potentially be recovered by simple brute-force since PACE is designed to support short passwords. This will jeopardize the following assets and security properties:

Table 12: Assets for PACE

Assets	Security properties at risks
PACE password	Confidentiality
Session keys	Confidentiality

Conclusion: PACE can by compromised by a sufficiently powerful CRQC. Consequently, an eMRTD could be read-out without knowledge of the MRZ or CAN. Also, any recorded transcript of the communication between an eMRTD and an Inspection System could be deciphered in the aftermath.

Threat severity: The loss of access protection by the PACE protocol would primarily impair the privacy of eMRTD holders since data from the integrated circuit could be read out without visually reading the MRZ or CAN from the data page, e.g. a closed passport booklet could be read-out by anyone in close proximity (Skimming attack).

The genuineness and authenticity of eMRTDs could still be verified if measures against quantum-algorithm based attacks on Passive Authentication or CA/AA are already in place.

4.4 Access Control (Terminal Authentication)

Terminal Authentication (TA) may be used to protect secondary biometrics in the eMRTD Application, like fingerprints or iris templates, which are considered highly sensitive data. This protocol enables the chip in the eMRTD to verify that the Inspection System is entitled to access that sensitive data. Furthermore, also access rights for data stored in the LDS2 application are granted by TA.

If TA is supported, issuing States or organizations may operate a CVCA Public Key Infrastructure (PKI) that issues certificates for authorized Document Verifiers (DV) which in turn issue Terminal Authentication certificates to authorized Inspection Systems. These Inspection Systems can now provide that chain of Terminal Authentication certificates to the eMRTD's IC starting at a trust-point stored on the IC which is the recent public key of the issuing State's or organization's CVCA. Afterwards, the IS can authenticate itself towards the IC by proving possession of the corresponding Terminal authentication private key

TA relies on the RSA or ECDSA signature algorithm for validation of the certificate chain and authentication of the terminal, whereby both algorithms can be broken by a sufficiently powerful CRQC using Shor's algorithms.
Hereby, the algorithm allows retrieving the private key from the public key or the Terminal Authentication certificate.

Table 13: Assets for Terminal Authentication

Assets	Security properties at risks
Terminal private key	Authenticity and integrity
CVCA-root and DV private key within the CVCA PKI	Authenticity and integrity

This will result in the following risks in the following conditions:

Table 14: Risk Analysis for Terminal Authentication Assets

Assets	Risk	Conditions
Terminal private key	An attacker could access sensitive data on the chip without authorization.	The Terminal Authentication certificate must match the CVCA public key stored on the eMRTD to be read out.
CVCA-root and DV private key	Compromise of a DV or CVCA private key could enable actors to • extend their access rights to data within the eMRTD (to access more data/perform more operations); • extend the validity period (generate new certificate with a new expiration date); • attacks over eMRTD by updating eMRTD effective date far in the future so that any rightful CVC is perceived by the eMRTD as expired; • update the CVCA public key within the eMRTD and thus take control of the eMRTD; • impersonate the DV or IS when sending a CSR request.	This risk always holds

Conclusion: TA can by compromised by a sufficiently powerful CRQC, which would enable any Inspection System to readout protected data, even without permission.

Threat severity: The loss of access protection by the TA would also primarily impair the privacy of eMRTD holders since Inspection Systems can now access sensitive biometric data or any data stored in the LDS2 application without proper authorization.

4.5 Session encryption (Secure Messaging)

After execution of the PACE protocol all communication between an eMRTD and an Inspection System is protected by Secure Messaging, which provides protection against eavesdropping. Secure Messaging relies on the 3DES or AES symmetric encryption cipher using session keys derived during PACE.

Quantum computing will put at risk in the future the session keys used to protect the Secure Messaging channel thanks to Grover's Algorithm and brute force attacks. This will jeopardize the following assets and security properties which are instrumental for the trust in eMRTDs.

Table 15: Assets for Secure Messaging

Assets	Security properties at risks
Session keys	Confidentiality

This will result in the following risks in the following conditions:

Table 16: Risk Analysis for Secure Messaging Assets

Assets	Risk	Conditions
Session keys	Session keys may be recovered during a transaction or after (by exploiting the logs). This would allow the attacker to • Recover data which are exchanged with the eMRTD in plain text, including the identity information; • Impersonate a rightful eMRTD in the course of a communication; • Impersonate a rightful Inspection system in the course of a communication; • Impersonate someone else identity in the course of a communication;	The session keys are small enough so that the attacker can apply Grover algorithm.

Conclusion: Grover's algorithm can lower the security level of the AES algorithm employed for Secure Messaging.

Threat severity: The attack can be mitigated by using the AES256 algorithm which is already defined in the current specifications. Even if Grover's algorithm could be used to its full extent, AES256 would still achieve a security level of 128 bits, which is considered secure.

The security provided by Passive Authentication, AA or TA would not be affected by Grover's algorithm. Also, CA and the PACE-protocol are not affected if they are used to derive sufficiently long symmetric keys (e.g. AES256).

5. Specific requirements for cryptographic mechanisms in eMRTDs

The last chapter provided a comprehensive summary of the threats posed by quantum computers to cryptographic security mechanisms currently implemented in eMRTDs and the eMRTD PKI. Building hereupon, this section compiles necessary requirements that cryptographic algorithms must meet to achieve the security objectives for the security mechanisms available in eMRTDs. These requirements are intended to systematically define comprehensible and plausible criteria for evaluating the suitability and maturity of potential quantum-safe algorithms, ensuring they effectively address the previously stipulated security

objectives. Furthermore, also functional requirements like computational performance, the message sizes, or necessary storage capacity are taken into account.

To achieve this, the chapter will identify the relevant cryptographic constructs for each use case to meet the security objectives and then specify the necessary requirements for those constructs.

5.1 Passive Authentication

In order to guarantee the authenticity and integrity of data stored on an eMRTD Passive Authentication relies on **Digital Signatures**.

R.PA.1: The Digital Signature algorithm must be able to generate a quantum-safe signature that allows to verify the authenticity and integrity of the signed data.

R.PA.2: The assurance of the signature must be sustained over a period of at least 10 years.

These Digital Signatures are generated on behalf of the issuing State or organization of an eMRTD using key material under sole control of that State or organization. That key material is highly sensitive and must be maintained securely by a Certificate Authority, which implements organizational and technical security measures to achieve an appropriate security level. General provisions on secure Certificate Authority operation are not provided here. However, algorithms currently being standardized lead to specific requirements that may not yet be implemented by a CA and are therefore explicitly addressed below.

Currently, the secure implementation of quantum-safe algorithms in soft- and hardware is still under heavy investigation. For secure operation these algorithms must achieve the same resistance against classical attacks.

R.PA.3: The Digital Signature algorithm should be securely implementable on a Hardware Security Module (HSM). This covers at least key generation, key storage and key usage. The security functionality of the HSM MUST be evaluated at a comparable evaluation depth to that of HSMs implementing current Digital Signature algorithms. The evaluation should in particular cover tamper resistance, emanation and side channel freeness, suitable access control, random number generation and security of cryptographic operations.

Certificate Authorities may support external generation of key material, key backup and import of keys, parallel operation of keys on multiple HSMs in order to ensure the resilience, redundancy and operational performance of the infrastructure. The procedures required for this may have to be adapted to the characteristics of new algorithms, i.e. some hash-based quantum-safe algorithm require a careful management of the private key’s internal state.

R.PA.4: The implementation of the Digital Signature algorithm should take technical and/or organizational management procedures into account which minimize the risk of operation errors that could result in key compromise.

Verification of data protected by Passive Authentication is done by an Inspection System and does not require any computation by the IC of an eMRTD. However, the certificates including its public-key to verify the Digital Signature, as well as the signature itself must be presented by the eMRTD towards the IS. These data are persistently stored in the non-volatile memory (NVM) of the IC in case of eMRTD, or the barcode itself in the case of VDS.

Therefore, size of the data to be persistently stored must not exceed the capacity of the underlying eMRTD.

R.PA.IC.1: The size of the signature generated by the Digital Signature algorithm and the certificate to verify the signature must not exceed the storage capacity of the IC of an eMRTD.

In case of VDS the storage capacity is limited by the size of the barcode. In order to reduce data to be stored in the barcode the certificate may be provided to the IS by some out-of-band mechanism. Yet, the VDS must store the actual signature.

R.PA.VDS.1: The size of the signature generated by the Digital Signature algorithm must not exceed the capacity of the barcode of a VDS.

5.2 Prevention against cloning (Chip/Active Authentication)

Prevention against cloning is implemented in the IC of an eMRTD by a **Proof-of-Possession mechanism** where the IC demonstrates its possession of a particular cryptographic key towards the inspection system.

R.CA.1: The Proof-of-Possession mechanism must be quantum-safe and be able to demonstrate possession of a cryptographic key.

The objectives of the mechanism can only be achieved if the key material on the IC is protected against duplication and extraction. Thus, the Proof-of-Possession mechanism algorithms executed on the chip must achieve the same resistance against classical attacks against the respective implementation.

R.CA.2: The Proof-of-Possession mechanism should be securely implementable on a secure cryptographic component (e.g. smart card). This covers at least key generation, key storage and key usage. The security functionality of the secure component MUST be evaluated at a comparable evaluation depth to that of secure components implementing current Digital Signature algorithms. The evaluation should in particular cover tamper resistance, emanation and side channel freeness, random number generation and security of cryptographic operations.

The Proof-of-Possession mechanism requires to perform calculations on the IC of the eMRTD. The runtime of these directly have a direct impact on the duration of the inspection procedure, which should be kept as low as possible in order to avoid congestions at the border. Since the mechanism must be implemented in each IC, the hardware costs scale linearly with the number of eMRTDs issued.

R.CA.3: The runtime of the Proof-of-Possession mechanism executed by the secure component should not delay the inspection procedure and should be as fast as possible.

R.CA.4: The Proof-of-Possession mechanism must be cost-effectively implementable on a secure component.

5.3 Protection against skimming and eavesdropping

Protection against skimming and eavesdropping is enforced by the IC of an eMRTD by a Chip Access Control mechanism that denies access to its contents unless the inspection

system can prove that it is authorized to access the IC. This proof is given in a cryptographic protocol, where the inspection system proves knowledge of the information derived from the physical document, like the MRZ or CAN. That protocol relies on **Password based authentication and key-agreement algorithm**, which derives session keys for the subsequent communication between the IC and the inspections system.

R.CAC.1: The Password based authentication and key-agreement algorithm must be quantum-safe and allow to derive cryptographic strong session keys from passwords with low entropy (e.g. the CAN or MRZ).

Since the Password based authentication and key-agreement algorithm must be implemented on an IC of an eMRTD equally as the Proof-of-Possession mechanism, the respective requirements apply also here.

R.CAC.2: The requirements R.CA.2, R.CA.3, and R.CA.4 apply mutatis-mutandis for the Password based authentication and key-agreement algorithm.

5.4 Access control for sensitive biometric data

Inspection Systems that want to readout protected sensitive biometric data must prove their authorization by means of a certificate. The authorization is validated by the eMRTDs IC by firstly, verifying the **Digital Signature** of the certificate and secondly, verifying that the inspection system is in possession of a secret cryptographic key (**Proof-of-Possession)** linked to the certificate. Therefore, the IC must implement the respective verification mechanisms.

R.TA.1 The Digital Signature algorithm must be able to generate a quantum-safe signature that allows to verify the authenticity and integrity of the signed data.

R.TA.2 The mechanism to verify the Digital Signature should be implementable on a secure cryptographic component (e.g. smart card).

R.TA.3 The Proof-of-Possession mechanism must be quantum-safe and be able to demonstrate possession of a cryptographic key.

R.TA.4 The mechanism to verify the Proof-of-Possession should be securely implementable on a secure cryptographic component.

R.TA.5 Both the Digital Signature and Proof-of-Possession verification mechanisms must be cost-effectively implementable on a secure component.

6. Status quo of Post-Quantum Cryptography (PQC)

Disclaimer: *The following chapters are still at an early draft stage and may be subject to significant changes in future versions of this document.*

6.1 Standardization of PQC primitives

End of 2016, NIST initiated a process to evaluate and standardize one or more quantum safe public-key cryptographic algorithms. This process was carried out in several rounds, whereby after the third round a total of four candidate algorithms have been selected for standardization, and four additional algorithms will undergo evaluation in a fourth round.

Hereby, three algorithms have been identified as candidates for as a scheme for digital signatures (CRYSTALS-Dilithium, FALCON and SPHINCS+) and a fourth algorithm will be standardized for implementation into key establishment schemes (CRYSTALS-KYBER).

Another class of cryptographic algorithms that must be considered separately are stateful hash-based signature schemes, which are constructed using hash trees and one-time signatures and are also called Merkle signatures. The security properties of Merkle signatures are very well understood and are considered quantum safe since a security reduction of Merkle signatures to the security of the underlying hash function has been proven. Thus, these signatures were standardized by the IETF some years ago (HSS/LMS, XMSS/XMSS^MT).

Open Topics:
- Intellectual Property

6.1.1 Digital Signatures

The following table provides a comparison of quantum safe signature algorithms with respect to their security strength, required key size and computational performance. The key size has an impact on the required storage capacity of a chip or a barcode and on the amount of data to be transferred. The data transfer together with the computational performance significantly determine the time required for the complete execution of a cryptographic protocol based on the respective primitive.

Table 17: Comparison of Post-Quantum Signature Algorithms

PQC Algorithm	Security level / strength [bits]	Private / Public-key size [bytes]	Signature size [bytes]	Implementability on IC			Implementability on HSM		Standard / Expected Finalization
				Key generation	Signature generation	Signature verification	Signature generation	Signature verification	
LMS[7]	2 / 128	/ 56	2828						IETF RFC 8554 and NIST SP 800-208 / final
HSS (LMS with multiple Merckle tree)[7]	2 / 128	/ 60	5076						
XMSS[7]	2 / 128	/ 68	2820	2, 3	2, 4	2, 4			IETF RFC 8391 and NIST SP 800-208 / final
XMSS^MT[7]	2 / 128	/ 68	4963	2, 3	2, 5, 6	2, 5, 6			
CRYSTALS Dilithium (ML-DSA)	1 / 128	2528 / 1312	2420	2	2, 5	2			NIST FIPS 204 / 2024
	3 / 192	4000 / 1952	3293	2	2, 5	2			
	5 / 256	4864 / 2592	4595	2	2, 5	2			
SPHINCS+ (SLH-DSA)	1 / 128	32 / 64	17088	2, 3	2, 3	2,5,7			NIST FIPS 205 / 2024
	3 / 192	48 / 96	35664	2, 3	2, 3	2,5,7			

	5 / 256	64 / 128	49856	2, 3	2, 3	2,5,7			
FALCON	1 / 128	1281 / 897	690	2, 7, 8	2, 7, 8	2			no draft yet available
	5 / 256	2305 / 1793	1330	2, 7, 8	2, 7, 8	2			
…									

Remarks:

1. LMS/HMS/XMSS/XMSS^MT key sizes for 2^20 (approx. 1 million) total messages per key using SHA-2 as underlying hash-function.
2. Security evaluation/certification of implementation on hardware not yet available.
3. Not implementable in the mid-term at least on IC as it requires a huge number of operations which are not possible for the type of IC used for eMRTD.
4. It can be implemented on today's IC with good performances when configured at most for a maximum number of 1 million signatures.
5. ICs are already available on which it may be implemented. Yet, the performance may be poor as dedicated hardware accelerators are necessary to streamline the execution and ensure a smooth usage and user experience. Next generation of ICs could offer dedicated hardware accelerators (within 3 to 5 years).
6. This operation is at least twice as slow as XMSS version
7. The availability of IC offering dedicated hardware accelerators depends on the market traction.
8. There are currently no ICs available on which it could be implemented as this operation requires a large amount of RAM which can't be found on today's IC. Next generation of ICs could offer sufficient RAM (within 3 to 5 years).

6.1.2 KEMs

Table 18: Comparison of Post-Quantum KEMs

PQC Algorithm	Security level / strength [bits]	Private / Public-key size [bytes]	Cipher-text size [bytes]	Implementability on IC			Implementability on HSM		Standard / Expected Finalization
				Key generation	Signature generation	Signature verification	Signature generation	Signature verification	
CRYSTALS KYBER (ML-KEM)	1 / 128	1632 / 736	800	1	1	1			NIST FIPS 203 / 2024
	3 / 192	2400 / 1088	1152	1	1	1			
	5 / 256	3168 / 1440	1504	1	1	1			
Classic McEliece	1 / 128	6492 / 261120	96	1	1	1			ISO/IEC 18033-2 Amd 2
	3 / 192	13608 / 524160	156	1	1	1			
	5 / 256	13932 / 1044992	208	1	1	1			
FrodoKEM	1 / 128	19888 / 9616	9720	1	1	1			ISO/IEC 18033-2 Amd 2
	3 / 192	31296 / 15632	15744	1	1	1			

	5 / 256	43088 / 21520	21520	1	1	1			
BIKE									Only specification available
HQC									Only specification available

Remarks:
1. Security evaluation/certification of implementation on hardware not yet available.

6.2 Standardization of hybrid schemes

The ISO/IEC 7816-8:Amd1 has been published (https://www.iso.org/standard/84543.html).

This standard defines interfaces, methods and data structure (including Card Verifiable Certificate format relevant for quantum safe version of EAC) for using quantum safe cryptography on secure hardware. It provides a generic framework for using any quantum safe algorithms on secure hardware, and also takes into consideration all finalists from the NIST contest, as well as HMS/LMS and XMSS/XMSS^MT. As such it is instrumental for the implementation of quantum safe protocols on eMRTDs.

Other works is also on going within SC17/WG4 on that topic.

6.3 Hardware implementation of PQC (PQC-only and hybrid schemes)

Open Topics:
- Overview of new chip prototypes that support PQC implementation
- Overview of existing chip types that are relevant for PQC implementation
- Status quo of research on side channel attacks for the relevant chips
- Runtime evaluation for the relevant chips
- Further results from deployment tests
- …

6.4 Hardware implementation of symmetric cryptography

Currently available secure cryptographic processors and HSMs both already support AES with key-lengths up to 256 bit. Even if a CRQC could perform Grover's algorithm, a security level of 128 bits could still be achieved, which is considered secure.

6.5 Implementation of PQC into protocols

Topics:
- Message syntax
- Signature schemes
- Asymmetric encryption
- Key-Exchange / Diffie-Hellman
- X.509
- …

The Doc 9303 specifications for eMRTDs build up on data structures that are specified by other standards development organizations; this clause provides information on the status of the integration of PQC algorithms in these data structures.

With respect to Passive Authentication Doc 9303 relies on

- Internet X.509v3 certificates and Certificate Revocation Lists (CRL) as specified in RFC 5280, see in particular Doc 9303 part 11
- Cryptographic Message Syntax (CMS) as specified in RFC 3369 / RFC 3852 / RFC 5652
    - for the encoding of the Document Security Object (SOD), see Doc 9303 part 10
    - for the encoding of CSCA Master Lists, see Doc 9303 part 12
    - for the encoding of Deviation Lists, see Doc 9303 part 12

The Internet Engineering Task Force (IETF) has started to specify provisions for using PQC algorithms in Internet X.509 certificates and CRLs as well as in CMS. In the IETF scope are the algorithms under standardization by the US NIST and the IETF aims to publish specifications in conformance with the final US NIST standards. In addition, further PQC algorithms are in the scope of the IETF.

Table 19 provides an overview on the IETF activities with respect to the support of PQC signature algorithms in Internet X.509 certificates, CRLs and CMS.

Table 19: Overview of IETF activities to support PQC signature algorithms X.509 certificates, CRLs and CMS.

PQC Algorithm	Internet X.509v3 / CRL	CMS
ML-DSA [NIST FIPS 204 ipd] (Dilithium)	Draft available [ML-DSA-certs]	No draft available/found so far
SLH-DSA [NIST FIPS 205 ipd] (SPHINCS+)	Draft available [HBS-certs-slhdsa]	Draft available [SLH-DSA-CMS]
FALCON [FALCON]	No draft available/found so far	No draft available/found so far
Hierarchical Signature System (HSS)/Leighton-Micali Signature (LMS) [RFC 8554]	Draft available [HBS-certs]	[RFC 8708] published
eXtended Merkle Signature Scheme XMSS and XMSS^MT [RFC 8391]	Draft available [HBS-certs]	No draft available/found so far

Furthermore, IETF is currently drafting a specification that defines conventions for the use of KEM algorithms by the originator and recipients to encrypt and decrypt CMS content [KEM-CMS].

In addition, there is an Internet-Draft submitted as a proposed work item to the IETF that specifies composite signatures and composite public keys for use in the Internet PKI including Internet X.509v3 certificates, CRLs, and CMS [Composite]. These composite signatures are combinations of multiple cryptographic algorithms, such as a traditional algorithm (RSA, DSA, or ECDSA) and a PQC algorithm, and "can be treated as a single atomic algorithm at the protocol level". These composite signatures are a specific type of a hybrid scheme.

6.6 Research projects (and other results) in eMRTD context

Open Topics:

- Suitability for eMRTDs
- Runtime results and limitations
- Side channels

The research project PoQuID [PoQuID] has developed and investigated possible PQ-resistant protocols (PQC-only, hybrid, KEM-only, KEM+Signature) that could be considered as a substitution for CA/AA and AA. Besides the protocol definitions, the project provides a security proof of the protocols and demonstrated their implementation on real-world smart card chip.

7. Cost analysis / Prerequisites for introduction of new cryptographic mechanisms

7.1 Approaches for PKI transition

This section gives a rough overview of various approaches for the transition from classic PKI to a quantum secure PKI. A similar overview was done by Vogt and Funke [VogFun21] in the context of electronic identity documents.

Parallel Infrastructure
One approach would be to set up a parallel infrastructure for the creation of digital signatures over eMRTD data. This would require to replicate the whole PKI infrastructure consisting of CSCA and Document /Seal Signer on the part of the issuing authority, whereby the duplicated infrastructure would implement quantum safe PQC instead of classic cryptography. eMRTDs would have to be provided with two certificates for this (e.g. two Document Security Objects/EF.SOD on an electronic passport), which could be validated individually during an inspection procedure at the discretion of the receiver.

A major advantage of this approach would be the strong backward compatibility to currently deployed Inspection Systems. Any IS not yet supporting PQC could just ignore the second document security object and still perform the inspection of an eMRTD using the primary object which provides signatures based on classic cryptography. Thus, any state could introduce a quantum safe PKI at a time of its choosing without fearing incompatibilities of the eMRTDs it issues at the border of other states.

A disadvantage would be the potentially long term operation of two parallel infrastructures with the resulting operating costs. Also, this approach would probably have limited applicability for digital seals, as integrating two signatures would increase the size of the barcode too much.

Hybridization of PKI
Hybridization is a possible approach for PKI transition, where regular cryptography is combined with quantum safe cryptography for which hindsight is still needed.
A technique for hybridization of PKI is standardized in ITU T-REC X.509:2019 [ITU-X.509] and ISO/IEC 9594-8:2020 [ISO/IEC 9594-8] via use of the certificate extensions containing an alternate public key and signature. This technique is covered by patents from Isara, but Isara has decided to dedicate these patents to the public:
https://www.isara.com/company/newsroom/isara-dedicates-four-hybrid-certificate-patents-to-the-public.html
IETF is currently drafting a specification that offers comparable properties [Chameleon-certs].

Hybridization of PKI enables to maintain a single PKI, while leaving flexibility on the side of entities processing the certificates (Inspection system eMRTD...) to validate or ignore the supplemental quantum safe signature, depending on whether their implementation recognizes the appropriate certificate extensions and has the policy to interpret them. Therefore, this approach could ensure a smooth migration by allowing coexistence of different versions of Inspection system and eMRTD on the field while using a single PKI. It must be stressed that verifying only the classic signature within hybrid certificates does not provide any security against quantum algorithms.

The specifications provided in ITU T-REC X.509 and ISO/IEC 9594-8 define a format for certificates that can be processed with backwards compatibility to systems that currently do not support the verification of quantum safe certificate extension. IS for eMRTDs should ignore non-critical extensions, but it must be stressed that states need to check if any IS does implement this correctly. If this is not the case, these inspection systems would need to be updated to at least be able to interpret the new hybrid certificate format as otherwise eMRTD implementing the certificates will not be recognized.
Furthermore, the current specification for VDS do not allow certificate extensions.

Quantum safe only cryptography

Migration to a PKI that only relies on quantum safe cryptographic algorithms may take decades, as time is needed to have hindsight enough on these new algorithms. The quantum safe cryptographic algorithms are relatively new and less researched, particularly in terms of implementation security. Therefore, several governmental agencies (e.g. BSI, …) currently do not endorse the exclusive implementation of new quantum safe algorithms into cryptographic protocols, but instead recommend to always implement them together with a classic algorithms in a hybrid scheme. One exception are stateful hash-based signature schemes which are also resistant against quantum algorithms, but only rely on classic well understood cryptographic primitives and are therefore already considered suitable for exclusive implementation as described in the following.

Quantum safe only cryptography based on Merkle signature schemes

Migration to quantum safe cryptography could be envisioned for CSCA, DS and SOD in the very short term by using HMS/LMS or XMSS/XMSS^MT (stateful hash-based signature also called Merkle signature) which are proven to be secure against the quantum computer.
Nevertheless, this approach should be carefully assessed as it has some consequences:

- Stateful hash-based signature schemes can only create a limited number of messages with one key-pair. As the number of eMRTDs to be issued is predictable, this is likely not an issue for a document issuing PKI since key rollovers can be easily scheduled. Still a careful analysis of the utilization of signature keys is required since also auxiliary signature which are required for the creation of CRLs or internal management operations, must be taken into account.
- Stateful hash-based signature schemes require that each key-pair used for the underlying one-time signature scheme is never reused or the scheme loses it security. Thus, the state must be managed securely to ensure that the scheme remains secure. This is especially important if a key export functionality is considered since the state must remain consistent even during key export/import procedures (e.g. a key must not be imported or used if it is not known which one-time key-pairs have already been used.)
- The size of the private key depends on the number of signatures to be generated. While it may not be an issue for the CSCA, it will be relevant for the DS key. Therefore, it may be necessary to reduce the number of SOD a DS could create, in order to reduce the size of the signature key;

- The memory size of the eMRTD should be large enough to store the DS and SOD which will be much larger. The signature size of HMS/LMS and XMSS/XMSS^MT depends on the number of signatures the private key is able to generate, which can result in relatively large signatures.

Regarding the usage of HMS/LMS and XMSS/XMSS^MT, it may be much more challenging for signed barcode because of the very large size of the signature and the size constraint for barcode (e.g. Schengen Visa).

Note: PKIs offer the possibility to use different algorithms at the Root and End Entity levels. In particular, ICAO Doc 9303 Part 12 allows that issuing State or organization may support different algorithm for use in their CSCA and Signing Certificate keys.

7.2 Approaches for document rollout

As of now, and until all the security mechanisms used within eMRTD are updated to be quantum safe, the validity of eMRTD could be reduced to limit risks (e.g. from 10 to 5 years).

The rationale for limiting the validity of eMRTD is that it will reduce the time needed to renew all eMRTDs on the field once (1) the security mechanisms are available and (2) the decisions for migration are taken. It will bring higher agility and flexibility to issuing authority to react in the case of a major breakthrough regarding the advent of quantum computers.

7.3 Approaches for already issued documents

Update of eMRTD on the field may be a possibility to address the risk of quantum computer. For instance, (1) new algorithms may be configured, or (2) cryptographic protocols or security measures may be added.
However, addition of brand new cryptographic algorithms on an eMRTD through update entails a number of challenges. First the capacities of the underlying chip may be too limited to implement a brand new algorithm (e.g. NVM and/or RAM is insufficient). In addition, tamper resistant implementation of cryptographic algorithms requires hardware implementation of cryptographic primitives which are tamper proof. If such primitives are not present in the chip, it is challenging to implement a tamper resistant implementation of cryptographic algorithm.

7.4 Dependencies

- Standardisation
- Prototyping
- Evaluation of (side channel) attacks
- Testing
- Certification
- Chip production and document production
- Document rollout
- …

8. References

[HE2024] Houston-Edwards, Kelsey, Tomorrow's Quantum Computers Threaten Today's Secrets. Here's How to Protect Them, Scientific American, February 1, 2024, https://www.scientificamerican.com/article/tomorrows-quantum-computers-threaten-todays-secrets-heres-how-to-protect-them-2/

[NIST SP 800-131A] National Institute of Standards and Technology (2019), Transitioning the Use of Cryptographic Algorithms and Key Lengths, NIST SP 800-131A Revision 2, https://doi.org/10.6028/NIST.SP.800-131Ar2

[NIST FIPS 204 ipd] National Institute of Standards and Technology (2023), Module-Lattice Based Digital Signature Standard. (Department of Commerce, Washington, D.C.), Federal Information Processing Standards Publication (FIPS) NIST FIPS 204 ipd. https://doi.org/10.6028/NIST.FIPS.204.ipd (Draft)

[NIST FIPS 205 ipd] National Institute of Standards and Technology (2023), Stateless Hash-Based Digital Signature Standard. (Department of Commerce, Washington, D.C.), Federal Information Processing Standards Publication (FIPS) NIST FIPS 205 ipd. https://doi.org/10.6028/NIST.FIPS.205.ipd (Draft)

[NIST SP 800-208] National Institute of Standards and Technology (2020), Recommendation for Stateful Hash-Based Signature Schemes, NIST SP 800-208 (Final), https://doi.org/10.6028/NIST.SP.800-208

[FALCON] Fast-Fourier Lattice-based Compact Signatures over NTRU, https://falcon-sign.info

[RFC 8391] XMSS: eXtended Merkle Signature Scheme, https://www.rfc-editor.org/rfc/rfc8391

[RFC 8554] Leighton-Micali Hash-Based Signatures, https://www.rfc-editor.org/info/rfc8554

[RFC 8708] Use of the HSS/LMS Hash-Based Signature Algorithm in the Cryptographic Message Syntax (CMS), see https://www.rfc-editor.org/info/rfc8708

[ML-DSA-certs] IETF draft, Internet X.509 Public Key Infrastructure: Algorithm Identifiers for Dilithium, IETF draft, https://datatracker.ietf.org/doc/draft-ietf-lamps-dilithium-certificates/

[SLH-DSA-CMS] IETF draft, Use of the SLH-DSA Signature Algorithm in the Cryptographic Message Syntax (CMS), https://datatracker.ietf.org/doc/draft-ietf-lamps-cms-sphincs-plus/

[HBS-certs] IETF draft, Internet X.509 Public Key Infrastructure: Algorithm Identifiers for HSS and XMSS, https://datatracker.ietf.org/doc/draft-ietf-lamps-x509-shbs

[HBS-certs-slhdsa] IETF draft, Internet X.509 Public Key Infrastructure: Algorithm Identifiers for SLH-DSA, https://datatracker.ietf.org/doc/draft-ietf-lamps-x509-slhdsa

[KEM-CMS] IETF draft, Using Key Encapsulation Mechanism (KEM) Algorithms in the Cryptographic Message Syntax (CMS), https://datatracker.ietf.org/doc/draft-ietf-lamps-cms-kemri/

[Composite] IETF draft, Composite ML-DSA for use in Internet PKI, https://datatracker.ietf.org/doc/draft-ietf-lamps-pq-composite-sigs/

[Chameleon-certs] IETF draft, A Mechanism for Encoding Differences in Paired Certificates, https://datatracker.ietf.org/doc/draft-bonnell-lamps-chameleon-certs/

[PoQuID] Fischlin, M., von der Heyden, J., Margraf, M., Morgner, F., Wallner, A., Bock, H. (2023), Post-quantum Security for the Extended Access Control Protocol, https://eprint.iacr.org/2023/352.

[ISO/IEC 9594-8] ISO/IEC 9594-8:2020 (12/2020), Information technology - Open systems interconnection - Part 8: The Directory: Public-key and attribute certificate frameworks, https://www.iso.org/standard/80325.html

[ITU-X.509] ITU-T X.509 (10/2019), https://handle.itu.int/11.1002/1000/14033

[NC20] Michael. A. Nielsen and Ike. L. Chuang. Quantum Computation and Quantum Information. Cambridge University Press, Cambridge, UK, 2000.

[FMMC12] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. https://arxiv.org/abs/1208.0928

[TTK10] Yasuhiro Takahashi, Seiichiro Tani, and Noboru Kunihiro. Quantum addition circuits and unbounded fan-out. Quantum Information & Computation, 10(9&10):872-890, 2010.

[Mon85] Peter L. Montgomery. Modular multiplication without trial division. Mathematics of Computation, 44(170):519{521, 1985.

[AMG+16] M. Amy, O. D. Matteo, V. Gheorghiu, M. Mosca, A. Parent, and J. Schanck. Estimating the Cost of Generic Quantum Pre-image Attacks on SHA-2 and SHA-3. In Selected Areas in Cryptography - SAC 2016, Springer LNCS vol. 10532, pp. 317-337, 2016.

[NIST15a] NIST. Secure Hash Standard (SHS). Federal Information Processing Standards Publication 180-4, 2015.

[NIST15b] NIST. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. Federal Information Processing Standards Publication 202, 2015.

[RNSL17] M. Roetteler, M. Naehrig, K. M. Svore, and K. Lauter. Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms. In Advances in Cryptology - ASIACRYPT 2017 (Part 2), Springer LNCS 10625, pp. 241-270, 2017.

[GQAI23] Google Quantum AI. Suppressing quantum errors by scaling a surface code logical qubit. Nature. 2023 Feb;614(7949):676-681. doi: 10.1038/s41586-022-05434-1. Epub 2023 Feb 22. PMID: 36813892; PMCID: PMC9946823.

[BCG+24] S. Bravyi, A. W. Cross, J.M. Gambetta, D. Maslov, P. Rall, T.J. Yoder. High-threshold and low-overhead fault-tolerant quantum memory. Nature. 2024 Mar; 627(8005):778-782. doi: 10.1038/s41586-024-07107-7. Epub 2024 Mar 27. PMID: 38538939; PMCID: PMC10972743.

[NIST SP800-186] National Institute of Standards and Technology (2023), SP.800-186, Recommendation for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters. https://doi.org/10.6028/NIST.SP.800-186

[SRB+24] M. P. da Silva, C. Ryan-Anderson, J. M. Bello-Rivas, A. Chernoguzov, J. M. Dreiling, C. Foltz, F. Frachon, J. P. Gaebler, T. M. Gatterman, L. Grans-Samuelsson, D. Hayes, N. Hewitt, J. Johansen, D. Lucchetti, M. Mills, S. A. Moses, B. Neyenhuis, A. Paz, J. Pino, P. Siegfried, J. Strabley, A. Sundaram, D. Tom, S. J. Wernli, M. Zanner, R. P. Stutz, K. M. Svore. Demonstration of logical qubits and repeated error correction with better-than-physical error rates. 2024. https://arxiv.org/abs/2404.02280

[Mos15] M. Mosca. Cybersecurity in an era with quantum computers: will we be ready? 2015. https://eprint.iacr.org/2015/1075

[IBMR24] IBM Quantum Computing Roadmap. https://www.ibm.com/roadmaps/quantum/

[IonQR20] IonQ Quantum Computing Roadmap. : https://ionq.com/posts/december-09-2020-scaling-quantum-computer-roadmap

[FMMC12] A. G. Fowler, M. Mariantoni, J. M. Martinis, A. N. Cleland. Surface codes: Towards practical large-scale quantum computation. Physical Review A—Atomic, Molecular, and Optical Physics, 86(3), 032324. (2012).

[GE21] C. Gidney, M. Ekerå. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. Quantum 5, 433 (2021).

[UtimacoHSM] Utimaco. https://utimaco.com/products/firmware/quantum-protect

[MTGModule] MTG. https://www.mtg.de/en/post-quantum-cryptography/pqc-hsm

[Crypt4AHSM] Crypt4A. https://crypto4a.com/hardware-security-modules/

[LunaHSM] Thales. https://www.thalestct.com/luna-t-series-hardware-security-module-7-13-0-release-announcement/

[AHH+18] M. R., Albrecht, C., Hanser, A. Hoeller, T. Pöppelmann, F. Virdia, A. Wallner. Implementing RLWE-based schemes using an RSA co-processor. IACR TCHES, 2019(1):169–208, 2018.

[WGY20] B. Wang, X. Gu, Y. Yang. Saber on ESP32. In Mauro Conti, Jianying Zhou, Emiliano Casalicchio, and Angelo Spognardi, editors, Applied Cryptography and Network Security - ACNS 2020, volume 12146 of LNCS, pages 421–440. Springer, 2020.

[BRvV21] J. W. Bos, J. Renes, C. van Vredendaal. Post-Quantum Cryptography with Contemporary Co-Processors: Beyond Kronecker, Schönhage-Strassen & Nussbaumer. In 31st USENIX Security Symposium (USENIX Security 22) (pp. 3683-3697).

[PML+21] J. Y. Park, Y. H. Moon, W. Lee, S. H. Kim, K. Sakurai. A survey of polynomial multiplication with RSA-ECC coprocessors and implementations of NIST PQC Round3 KEM algorithms in Exynos2100. 2021. IEEE Access, 10, 2546-2563.

[GMV20] A. Greuet, S. Montoya, G. Renault. Speeding-up Ideal Lattice-Based Key Exchange Using a RSA/ECC Coprocessor. Cryptology ePrint Archive. 2020.

[GMV23] A. Greuet, S. Montoya, C. Vermeersch. Modular Polynomial Multiplication Using RSA/ECC Coprocessor. 2023. In International Conference on Network and System Security (pp. 283-304). Cham: Springer Nature Switzerland.

[RED+08] S. Rohde, T. Eisenbarth, E. Dahmen, J. Buchmann, C. Paar. Fast hash-based signatures on constrained devices. In Smart Card Research and Advanced Applications, volume 5189 of Lecture Notes in Computer Science, pages 104–117. Springer Berlin / Heidelberg, 2008.

[PQSt23] Federal Office for Information Security. Status of quantum computer development. https://www.bsi.bund.de/EN/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Quantentechnologien-und-Post-Quanten-Kryptografie/Entwicklungsstand-Quantencomputer/entwicklungsstand-quantencomputer.html

[NIST-IR8547] National Institute of Standards and Technology (2024), Transition to Post-Quantum Cryptography Standards, NIST IR 8547, https://doi.org/10.6028/NIST.IR.8547.ipd

[IBMHSM] IBM. https://cloud.ibm.com/catalog/infrastructure/hardware-security-module

[EntrustHSM] Entrust nShield 5s HSM. https://www.entrust.com/products/hsm/nshield-5s

[SecurosysHSM] Securosys Primus E-Series . https://www.securosys.com/en/e-series

[NIST FIPS 203 ipd] National Institute of Standards and Technology (2023), Module-Lattice-Based Key-Encapsulation Mechanism Standard. (Department of Commerce, Washington, D.C.), Federal Information Processing Standards Publication (FIPS) NIST FIPS 203 ipd. https://doi.org/10.6028/NIST.FIPS.203

[PPM17] Robert Primas, Peter Pessl, and Stefan Mangard. Single-Trace Side-Channel Attacks on Masked Lattice-Based Encryption. TCHES 2017

[KPP20] Matthias Kannwischer, Peter Pessl, and Robert Primas. Single-trace attacks on keccak. CHES 2020

[BGR+21] Joppe W. Bos, Marc Gourjon, Joost Renes, Tobias Schneider, Christine van Vredendaal. Masking Kyber: First- and Higher-Order Implementations. CHES 2021

[DNGW23] Elena Dubrova, Kalle Ngo, Joel Gärtner, and Ruize Wang. Breaking a fifthorder masked implementation of crystals-kyber by copy-paste. Asiacry PKC 2023

[MGT+19] Vincent Migliore, Benoit Gérard, Mehdi Tibouchi, and Pierre-Alain Fouque. Masking Dilithium: Efficient Implementation and Side-Channel Evaluation. ACNS 2019

[IETFXMSS] Hülsing, A., Butin, D., Gazdag, S., Rijneveld, J., Mohaisen, A.: Internet-Draft: XMSS: Extended Hash-Based Signatures. https://datatracker.ietf.org/doc/draft-irtf-cfrg-xmss-hash-based-signatures/ (2018)

[KGB+18] Matthias J. Kannwischer, Aymeric Genêt, Denis Butin, Juliane Krämer, and Johannes Buchmann. Differential Power Analysis of XMSS and SPHINCS. COSADE 2018

[GKPM18] Aymeric Genêt, Matthias J. Kannwischer, Hervé Pelletier, and Andrew McLauchlan. Practical Fault Injection Attacks on SPHINCS. https://eprint.iacr.org/2018/674

[CMP18] Laurent Castelnovi, Ange Martinelli, and Thomas Prest. Grafting Trees: a Fault Attack against the SPHINCS framework. PQCrypto 2018

[Gen23] Aymeric Genêt. On Protecting SPHINCS+ Against Fault Attacks. TCHES 2023

[VogFun21] Vogt, Sebastian, and Funke, Holger, How quantum computers threat security of PKIs and thus eIDs. 2021.

Annex A: Abbreviations

Abbreviation	
3DES	Triple DES
AA	Active Authentication
AES	Advanced Encryption Standard
CA	Chip Authentication
CAN	Card Access Number
CAM	Chip Authentication Mapping
CMS	Cryptographic Message Syntax
CRL	Certificate Revocation List
CRQC	Cryptographically Relevant Quantum Computers
CSCA	Country Signing Certification Authority
DH	Diffie Hellmann
DS	Document Signer
DSA	Digital Signature Algorithm
DTC	Digital Travel Credential
DTC-PC	DTC Physical Component
DTC-VC	DTC Virtual Component
DV	Document Verifier
EAC	Extended Access Control
ECC	Elliptic-Curve Cryptography
ECDH	Elliptic Curve Diffie Hellmann
ECDSA	Elliptic Curve Digital Signature Algorithm
ETD	Emergency Travel Document
eMRTD	Electronic Machine Readable Travel Document
HSM	Hardware Security Module
IC	Integrated Circuit
ICAO	International Civil Aviation Organization
LDS	Logical Data Structure
MRTD	Machine Readable Travel Document
MRZ	Machine Readable Zone
OID	Object Identifier
PACE	Password Authenticated Connection Establishment
PKI	Public Key Infrastructure
PKD	Public Key Directory
PQC	Post Quantum Cryptography
RSA	Rivest, Shamir and Adleman
SM	Secure Messaging
SO_D	Document Security Object
TA	Terminal Authentication
VDS	Visible Digital Seal

Appendix B

Information Paper

Quantum safe mechanisms for the Document Issuing PKI and Passive Authentication

Version – 0.04

Date – Oct 20, 2025

MACHINE READABLE TRAVEL DOCUMENTS

Information Paper

Quantum safe mechanisms for the Document Issuing PKI and Passive Authentication

Version – 0.04
Date – Oct 20, 2025

ISO/IEC JTC1 SC17 WG3/TF5
FOR THE
NEW TECHNOLOGIES WORKING GROUP

File	: 2025-10-20_WG3TF5_N0XXX IP - PQC_PKI+PA_Options_v0.4_Annex_for_TRIP_WP (003).doc
Author	: ISO/IEC JTC1 SC17 WG3/TF5

Quantum safe mechanisms for the Document Issuing PKI and Passive Authentication

Release : **0.04**
Date : Oct 20, 2025

Table of contents

1. Scope

The information paper "Cryptographic Agility and Post-Quantum Cryptography" [CSAGPQC] examined the primary threats posed by quantum computers to the security mechanisms currently implemented in electronic machine readable travel documents (eMRTDs). Among these, the threat to Passive Authentication is widely recognized as the most systemic. For example, an attacker with access to a quantum computer could compute the private keys of an Issuing State or organization from their certificates, enabling the fraudulent issuance of eMRTDs that would appear cryptographically valid.

The risk of quantum computer is looming. Considering the validity period of eMRTDs (usually ten years), the inertia to upgrade the surrounding infrastructure and the forecasts regarding availability of a cryptographic relevant quantum computer, actions should be taken as of now to migrate eMRTD security mechanisms so that they are immune against the quantum computer when the risk arises.
When addressing the challenge to migrate the existing security mechanisms so that they become quantum resistant, issuing authorities should also consider (1) uncertainty regarding the security of these post quantum algorithm and their hardware implementation – "Security", (2) today's technical challenges stemming from their larger key and cryptogram sizes and need for larger computational resources (CPU, RAM) – "Readiness", and (3) backward compatibility with existing infrastructure, in particular Inspection System – "Backward compatibility".

The paper at hand seeks to identify and evaluate options for migrating the security mechanisms that ensure the authenticity and integrity of eMRTDs and their impacts on (1) security, (2) readiness, and (3) backward compatibility. These options include the Document Issuing Public Key Infrastructure (PKI), which secures data through digital signatures, the certificate formats used for these signatures, and the verification processes employed by Inspection Systems. Therefore, this paper first establishes a set of guiding objectives for the migration called aspirations, which serve as the basis for evaluating potential options in the subsequent analysis.

This document is intended to support the definition, development and implementation of a phased migration strategy to ensure a systematic and structured transition to quantum safe mechanisms. The following phases should be considered:

- Phase 1: Information Gathering (State of Play) – This phase is largely addressed by the content of this paper, which provides an overview of the current landscape and the available options for Passive Authentication.
- Phase 2: Selection of Digital Signature Algorithms – This paper provides an overview of quantum-safe algorithms currently undergoing standardization. In order to limit complexity, the selection should be narrowed to a small set of algorithms that best meet the requirements of Passive Authentication.
- Phase 3: Risk assessment – Based on a risk assessment, and assessment of the readiness and impact on backward compatibility, issuers should evaluate which approach is necessary for their eMRTDs. However, to ensure full interoperability, Inspection Systems must be capable of supporting classical, quantum-safe-only, and hybrid signature schemes — combining a classic and a quantum-safe algorithm.
- Phase 4: Implementation of migration – This phase involves defining the approach(es) for migration, i.e., hybrid or purely based on post quantum-safe algorithms, considering the impact on the security, readiness and backward

compatibility. Regarding implementation of hybrid schemes, this phase involves defining the necessary technologies to implement such a scheme for Passive Authentication that remains backward compatible with existing Inspection Systems.
- Phase 5: Migration Planning – The final phase, which lies beyond the scope of this paper, involves individual states and issuing authorities developing their own detailed migration plans, informed by the foundational insights provided here.

Regardless of their chosen strategy chosen for document issuance, states should upgrade their Inspections Systems to support quantum-safe algorithms as soon as possible. While final specifications for quantum-safe eMRTDs are not yet available, Inspection System vendors can already consider implementing cryptographic libraries that offer support for the already standardized digital signature algorithms.

This paper is still in the drafting stage and will be revised in future versions. To move closer to a common understanding necessary for the next steps in standardization of quantum safe mechanisms for Passive Authentication, States are requested to provide their opinions in particular on the following questions:

1. **Which digital signature algorithm(s) do you consider suitable for Passive Authentication and the Document Issuing PKI? Are there any reasons why you would not support a particular algorithm (c.f. chapter 3)?**

2. **What approach do you envision for operating the Document Issuing PKI during the transition to quantum-safe algorithms (c.f. chapter 5)?**

3. **Which options for storing and distributing digital signatures on eMRTDs do you consider suitable to achieve both future-proofness and backward compatibility with existing non-quantum-safe Inspection Systems (c.f. chapter 6)?**

4. **Do you see a need for backward compatibility of future eMRTDs with quantum-safe signatures with inspection systems that are not able to verify quantum-safe signatures? Or do you advocate a fixed transition date after which all inspection systems must support the verification of quantum-safe signatures (c.f. Table 1 and section 2.1)?**

1.1 Short Introduction of Passive Authentication

Passive Authentication (PA) is used to verify the authenticity and integrity of the data stored on an eMRTD. When an eMRTD is produced, the data stored in the chip is digitally signed. A so-called document signer certificate is used for this purpose, which in turn is signed with the CSCA certificate (Country Signer Certificate Authority) of the Issuing Authority and whose private signature key is only available to the officially authorized passport manufacturer. This certificate forms the root of the CSCA-PKI (Public Key Infrastructure), a hierarchy for certificates to prove the authenticity of data on eMRTDs.
When an eMRTD is read, the signature of the data which is stored in the file EF.SOD on the chip is checked using Passive Authentication and traced back to the CSCA certificate. This makes it possible to determine whether the data stored in the eMRTD's chip has actually been issued on behalf of the Issuing Authority and whether it has not been manipulated.
This means that the actual creation of the signature takes place outside the chip. The chip merely stores the signature and the document signer certificate used to create the signature. The signature is calculated over a Document Security Object, which contains hash values of all data groups stored on the chip. When the signature is checked, the signatures are also

verified outside the chip, e.g., by an inspection system at the border control. The signatures are therefore always created outside the chip, which means that the performance of the signature algorithm used is of secondary importance with regard to the chip. The size of public key and the signature size of the signature algorithm can however impact the current chip requirements and should be considered carefully.

2. Aspirations for cryptographic mechanisms

The following table specifies aspirations for potential quantum safe cryptographic mechanisms that can be used for the Document Issuing PKI and Passive Authentication:

Aspirations	Definition	Significance
Quantum safe DSA	The Digital Signature algorithm must be able to generate a quantum-safe signature that allows to verify the authenticity and integrity of the signed data. *Note: This document only considers algorithms which are also endorsed by national agencies.*	MUST
Long term assurance	The assurance of the signature must be sustained over a period of at least 10 years. *Note: This aspiration may not only be met by a single cryptographic primitive, but hybridization schemes with several primitives may also be considered.*	MUST
Cryptographic agility	The cryptographic mechanism must provide means to support the transition to future signature algorithms not yet standardized today.	
Backwards Compatibility	The specification must allow to issue eMRTDs with quantum safe certificates and Document Security Objects that are verifiable by Inspection Systems not yet supporting quantum safe cryptography. Passive: A non-quantum safe IS can parse quantum safe certificates and will automatically ignore the quantum-safe parts. Active: An eMRTD can respond differently depending on whether it detects a non-quantum safe or quantum safe ready IS) *Note: Issuing Authorities may opt for issuing eMRTDs implementing only quantum safe algorithms immediately, in which case the aspiration is not applicable.*	MUST
Downgrade Protection	The inspection procedure must allow an Inspection System to detect whether parts of the signature has been tampered or removed (e.g. the quantum safe part of the signature).	MUST
HSM Support	The Digital Signature algorithm should be securely implementable on a Hardware Security Module (HSM). This covers at least key generation, key storage and key usage. The security functionality of the HSM should be evaluated at a comparable evaluation depth to that of HSMs implementing current Digital Signature algorithms. The evaluation should in particular cover tamper resistance, emanation and side channel freeness,	SHOULD

	suitable access control, random number generator and security of cryptographic operations.	
Data Size	The size of the signature generated by the Digital Signature algorithm and the certificate to verify the signature must not exceed the storage capacity of the IC of an eMRTD. *Note: The storage capacity of barcodes (IDB) should also be taken into consideration.*	MUST
Performance	The execution time of the Inspection Procedure should not increase significantly. This covers both the reading of the data from the chip as well as the verification of the signatures on both conventional inspection systems already in the field and future quantum safe inspection systems.	SHOULD

Table 1: List of Aspirations for the Document Issuing PKI and PA

2.1 Implications regarding backwards compatibility

Waiving the requirement for backwards compatibility could allow to accelerate the process of creating a specification that ensures the authenticity and integrity of eMRTDs while using quantum-secure mechanisms. In this case the specification could assume that its requirements will be supported by all inspection systems from a specific date onwards and thus would not require to provide provisions for eMRTDs that need to be inspectable by current inspection systems. This would allow for a slightly simpler specification that creates less burden for the issuing authorities in the transition phase and would simplify certificate handling for receiving authorities in the long term. In particular, issuing authorities would not be required to operate multiple PKI branches in parallel or to provide a backwards compatible certificates and signature on their eMRTDs.

On the other hand, dropping backwards compatibility would require to agree on specific transition date until which any inspection system must be updated to support the verification of quantum-safe signatures. Only after that date, issuing states and organizations would be able to issue eMRTDs protected by quantum-safe signatures that could be inspected electronically by any state. This could delay the introduction of quantum-secure signatures and consequently increase the time frame during which there is no protection against quantum algorithms. With regard to the expected timeframe for setting a deadline, similar previous transitions for the access protocol (from BAC to PACE) or for the encoding of biometric data (from ISO/IEC 19794 to 39794) should be noted.

Request for Information

States are invited to provide their opinion on the requirement of backwards compatibility:

Do you see a need for backward compatibility of future eMRTDs with quantum-safe signatures with inspection systems that are not able to verify quantum-safe signatures? Or do you advocate a fixed transition date after which all inspection systems must support the verification of quantum-safe signatures?

3. Quantum safe digital signature algorithms

Quantum safe cryptography encompasses a diverse set of cryptographic algorithms designed to remain secure against adversaries equipped with a cryptographic relevant quantum computer. Among the most prominent families of these algorithms are hash-based signatures and lattice-based signatures, each offering distinct advantages and trade-offs. Hash-based signatures rely on the security of cryptographic hash functions and provide well-established security guarantees along with practical implementations. Examples of hash-based signature schemes include LMS [RFC8554], XMSS [RFC8391], and SLH-DSA (SPHINCS+) [FIPS 205]. In contrast, lattice-based signatures, such as ML-DSA [FIPS 204] and Falcon [Falcon], leverage the hardness of (structured) lattice problems to achieve compact signatures and efficient performance, making them suitable for a wide range of applications. This chapter explores both families, examining their underlying principles, practical considerations, and applicability to passive authentication. Table 2 provides a high-level overview of the algorithms discussed in this chapter, while Sections 3.1 and 3.2 offer more detailed overviews of hash-based and lattice-based signature schemes, respectively. The next section discusses hash-based signatures (HBS) and lattice-based signature (LBS) schemes, and gives a more detailed summary of all algorithms.

Algorithm	Description
Lattice-Based	
ML-DSA	A digital signature scheme relying on the Fiat-Shamir with Aborts paradigm over Module-LWE (Learning with Errors) and Module-SIS (Shortest Integer Solution). Known for simplicity, speed, and relatively small signatures and keys.
Falcon	A digital signature scheme based on the NTRU assumption and the Hash-and-Sign design paradigm. Known for compact keys and signatures but with more complex signing/verification algorithms using Fast Fourier Transform (FFT).
Hash based	
LMS	Leighton-Micali Signature scheme: A stateful hash-based signature scheme that uses Merkle trees.
HSS	Hierarchical Signature System: An extension of LMS that uses multiple layers of Merkle trees.
XMSS	eXtended Merkle Signature Scheme: A stateful hash-based signature scheme with improved efficiency and formal security proofs.
XMSS^MT	Multi-Tree version of XMSS: A hierarchical variant of XMSS with multiple Merkle tree layers to improve scalability.
SLH-DSA (SPHINCS+)	Stateless hash-based signature scheme designed for quantum safe security with trade-offs in performance.

Table 2: Brief description of relevant quantum safe signature schemes

3.1 Comparison of Hash-Based Signatures

Hash-based signature schemes represent one of the most well-understood and mature classes of quantum safe cryptographic algorithms. Unlike many modern cryptosystems that rely on the presumed hardness of mathematical problems such as factoring, discrete logarithms, or lattice-based constructions, hash-based schemes derive their security directly from the properties of cryptographic hash functions. Since these hash functions can be instantiated with functions from the SHA family, the best-known quantum attack is Grover's algorithm, which asymptotically halves the bit security of a symmetric primitive. This makes hash-based schemes particularly attractive as a quantum safe alternative to RSA and discrete logarithm-based signatures.

Hash-based signatures are built upon the concept of one-time signatures (OTS), where each key pair is used only once to ensure security. In a hash-based signature scheme, the signer generates a set of one-time signature key pairs, each corresponding to a single signature operation. To efficiently manage multiple one-time signatures, a Merkle tree (or hash tree) is used. The Merkle tree enables the commitment to all the public keys generated from these one-time signatures. The commitment is represented by the root of the Merkle tree, which is simply a single hash. When signing a message, the signer signs it using one of the one-time signatures, and to prove that the corresponding public key is valid, they provide a proof consisting of the hash path from the leaf node (the one-time signature's public key) to the root of the Merkle tree. This structure allows for scalable, secure signing with a compact commitment to all public keys involved.
Among the prominent hash-based signature schemes, LMS (Leighton-Micali Signature) and HSS (Hierarchical Signature Scheme) offer scalable and efficient stateful solutions. Similarly, XMSS (eXtended Merkle Signature Scheme) and its multi-tree variant, XMSS^MT, provide standardized and highly secure mechanisms. SLH-DSA serves as a stateless alternative, eliminating the need for state management at the cost of larger key and signature sizes, making it suitable for scenarios where fault tolerance is a priority.

One of the critical trade-offs in hash-based schemes lies between efficiency and usability, particularly in terms of state management. Stateful schemes like LMS, HSS, XMSS, and XMSS^MT require careful tracking of the signing state, since each one-time key must only be used once and key-reuse immediately results in a compromised private-key. This introduces complexities in implementation but allows for more compact signatures and keys. In contrast, SLH-DSA circumvents state management entirely by introducing randomized signing, which ensures security without relying on a tracked state. However, this comes at the expense of computational overhead and larger cryptographic artifacts.

In the following sections, we will explore the properties and features of these algorithms in greater detail, focusing on their design principles, performance, and security trade-offs. Table 3 provides an overview of the standard instantiations of these algorithms.

Attribute	LMS	HSS	XMSS	XMSS^MT	SLH-DSA
Stateful	Yes	Yes	Yes	Yes	No
Security Level [bits]	192/256	192/256	192/256	192/256	128/192/256
Public Key Size [byte]	48/56	52/60	52/68	52/68	32/48/64
Secret Key Size [byte]	Impl. Dep.	Impl. Dep.	1296-2240	1296-2240	32/48/64
Signature Size [byte]	1296 - 9328	2644 - 9300	1492 - 9732	4963-27688	7856/16224/29792s 17088/35664/49856f
Performance (Key Gen)	6s – 1.5h	6s – 1.5h*	6s – 3min	3min-3h	3ms – 0.3s
Performance (Sign)					0.1s-5s
Performance (Verify)					1.5ms-12ms
Max. Sigs.	2^5 – 2^25	2^5 – 2^25*	2^10-2^20	2^20-2^60	2^63-2^68

Table 3: Overview of hash-based signature schemes

Note that Table 3 provides only a range for signature size, the maximum number of signatures per public key, and the performance of LMS, HSS, XMSS, and XMSS^MT signatures. The performance of the algorithms is taken from the respective RFCs or [NIST SP 800-208] describing them, or from [SPHINCS+v31] for SLH-DSA. This is because [NIST SP 800-208] and [RFC8554, RFC8391] specify a large number of parameter sets. For example, [NIST SP 800-227] defines 15 different parameter sets for the LMS signature scheme, 12 sets for XMSS, and 32 sets for XMSS^MT. Interestingly, neither [NIST SP 800-208] nor [RFC8554] provide concrete parameter sets for HSS. Furthermore, it is worth noting that since no standard specifies a limit on the size of the multitree for HSS, one may choose the size to accommodate at least 2^60 signatures.
SLH-DSA [FIPS 205] defines 12 different parameter sets. Specifically, there are three groups, each targeting a different security level: 128, 192, and 256 bits. Within each group, there are two subgroups: one prioritizes fast signature generation, while the other emphasizes shorter signatures. All parameter sets within these subgroups share the same parameters but can be instantiated with either the SHA2 or SHAKE family of hash functions, depending on the desired hash output size.

On the Security Level:
NIST, in Special Publication 800-57 Part 1, defines *x*-security in terms of equivalence to breaking a generic block cipher with *x*-bit input/output rather than the equivalence of the best cryptanalytic attack to generic brute force search. Consequently, one should also consider the actual quantum security these primitives offer. According to [AMG+16] and as discussed in Section 2 of [CSAGPQC], Grover's algorithm may asymptotically reduce the search space for breaking second preimages from 128 bits to 64 bits, 192 bits to 96 bits, and 256 bits to 128 bits.

On Efficiency:
In terms of efficiency, HSS, XMSS, and XMSS^MT share similar characteristics. Namely, all these algorithms require hours of computation to generate the public key when parameterized to support a similar number of signatures as SLH-DSA. In terms of public key size, all public keys are very small, but SLH-DSA offers the smallest. SLH-DSA can be configured for fast signing or small signatures. When configured for small signatures, the most secure parameter set has a signature size comparable to XMSS^MT. In fact, XMSS^MT signatures are only 7% smaller than those of SLH-DSA, or roughly 2 KB in absolute terms. However, for the same parameter set, SLH-DSA supports 256 times more signatures and offers faster public key generation. In contrast, XMSS^MT requires approximately 3 hours to generate the public key.
Crucially, these public key generation times are estimated using high-performance processors running on systems equipped with gigabytes of random access memory. In practice, hardware security modules are often implemented on slower embedded platforms like the ARM Cortex M4. Finally, the most time-consuming part is generating the 2^60 secret keys and public keys, and then computing the Merkle tree over all public keys. While it could be argued that the Merkle tree generation might be performed outside of the secure environment, the existing standards [RFC8554, RFC8391] and recommendations [NIST SP 800-227] do not explicitly specify such a delegation procedure.

On Statefulness:
Statefulness in the context of hash-based signature schemes refers to the requirement for the signer to maintain and update a record of the cryptographic keys or indices used during the signing process. This is essential to ensure that each key or index is used only once, as hash-based schemes rely on one-time signatures that are perfectly secure if used once but can be completely broken if used twice. The signer must securely store this state and update

it after every signature operation to avoid reusing keys, which would compromise the security of the scheme.
The primary advantage of statefulness is its efficiency. Stateful schemes like LMS, HSS, XMSS, and XMSS^MT are typically more compact in terms of signature and key sizes compared to stateless alternatives like SLH-DSA. Since the state is carefully managed and reused keys are avoided, the scheme does not require as much overhead for redundancy or additional security measures, making it potentially more efficient in practice.
One of the most critical issues of statefulness is the potential for state management failures. If the signer fails to update the state correctly (e.g., due to a system crash, rollback, or concurrent signing operations), the same key or index might be reused. An attacker could exploit this key reuse to recover the private signing key, rendering the scheme insecure. Additionally, securely storing and updating the state requires robust mechanisms to prevent tampering, rollbacks, or accidental corruption. This complicates implementation, especially in environments where reliable storage and synchronization are not guaranteed.
Another drawback of statefulness is its operational complexity. In environments where multiple devices or systems need to perform signing operations, synchronizing the state between these devices becomes a significant challenge. Without careful coordination, there is a risk of state desynchronization, which could again result in key reuse and compromise security. This requirement limits the flexibility of stateful schemes, particularly in scenarios where multiple devices or instances of the signer operate independently. An example of such a scenario may be hardware security modules, which introduce redundancy in the form of additional signing modules to handle module failures.
In summary, statefulness in hash-based signature schemes is a double-edged sword. It offers efficiency advantages but comes with the burden of securely managing and updating the state. While suitable for scenarios where reliable storage and controlled environments are available, the risks and complexities of statefulness make it less appealing for distributed systems. In contrast, stateless schemes like SLH-DSA, though slightly less efficient, eliminate these risks by design, making them better suited for critical applications where robustness against implementation errors is a priority.
As mentioned earlier, SLH-DSA differs significantly from LMS, HSS, and XMSS by being a stateless scheme. This design eliminates the need for state management, thereby avoiding the risk of key reuse due to improper state handling. SLH-DSA relies on a combination of hash-based primitives, including one-time signature schemes like WOTS+ (Winternitz One-Time Signature), and a tree-based structure to aggregate signatures. The stateless design inherently simplifies certain aspects of implementation but introduces other challenges. SLH-DSA requires the generation of many pseudorandom values during signing, which increases computational overhead.
Another characteristic of SLH-DSA is its significant overlap with the stateful hash-based signature scheme XMSS. In particular, the code for signature validation of XMSS is largely incorporated into the SLH-DSA code.

On implementation issues, side-channel and fault attacks.
Side-channel attacks are less prominent in HBS, since HBS are built from well-established symmetric primitives like hash functions, and many computational platforms already implement side-channel-hardened versions of these functions. Nevertheless, implementations must ensure constant-time implementation of the OTS signing and key generation procedures.

On the other hand, HBS have been shown to be vulnerable to fault injection attacks. LMS and XMSS, as well as their hierarchical variants, are stateful. Hence, injecting faults into the current state counter may lead to reusing the state. The first fault attacks that do not exploit the state were proposed in [CMP18]. These attacks target the signature generation of multi-tree schemes and affect multi-tree signatures like HSS, XMSS^MT, and SLH-DSA. Practical

evaluations of these attacks were shown in [GKPM18, ALCZ20]. [WWO+23] targets the checksum computation in HBS and is therefore applicable to LMS, XMSS, and SLH-DSA. Interestingly, [WWO+23] shows attacks on both the signature generation and verification processes. The authors demonstrate that a single bit flip in the signing procedure can compromise the security of hash-based algorithms. Attacking signature verification is particularly relevant as it allows bypassing the secure boot process in certain systems. Furthermore, the attack on the verification procedure has been evaluated as practical in [BSIFault]. In particular, [BSIFault] uses laser fault injection on a microcontroller implementing the verification procedure from the XMSS reference library [XMSSRef].
On the positive side, in the context of passive authentication, fault attacks on HBS are relevant for key generation and signature creation, which take place during the HSM setup and personalization phase, and for verification of the data groups. All of these processes usually occur in somewhat controlled environments where physically executing elaborate fault injection attacks may be difficult. Personalization is performed at issuing authorities' secure facilities. In the case of stolen HSMs, the keys can be revoked. Data group verification is typically done at airports, border checks or online, where the hardware chip implementing the verification procedure is physically isolated from the document holder.

Forward Security.
A desirable feature in the context of personal identity documents distributed at scale is forward security of signature schemes [And97]. Forward security, in the context of signature schemes, guarantees that breaking the signature scheme, even recovering the secret key at some point, does not invalidate the authenticity of signatures produced with prior states. Therefore, in the case of a compromised HSM, there is no need to revoke eMRTDs issued before the compromise. However, this assumption is based on the premise that it is infeasible to recover prior states from the HSM. The validity of this assumption may be implementation-dependent. All HBS considered in this document have some form of forward security. However, the statefulness of an HBS may introduce vulnerabilities due to potential rewind attacks and fault injection into the state counter. Physical storage of all secret keys in stateful HBS schemes may be another attack vector on signatures issued before the HSM compromise. In contrast to stateful HBS, SLH-DSA appears to be more robust against adversaries attempting to break forward security, due to its stateless design.

3.2 Comparison of Lattice Signatures

Lattice-based signature schemes offer strong security guarantees based on the hardness of well-studied mathematical problems on lattices, such as the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP). Prominent lattice-based signatures include ML-DSA [FIPS 204], which is based on the CRYSTALS-Dilithium [Dilithium], and Falcon, which is based on the NTRU-based signature scheme [Falcon] and is supposed to be standardised under the name FN-DSA. The final version of Falcon to be prepared by NIST is still unknown (to be released as FIPS 206). ML-DSA is known for its simplicity and ease of implementation. The algorithm leverages efficient polynomial arithmetic over structured lattices for performance. Falcon is designed for compactness in the signatures and public key. It utilizes the NTRU lattice and Gaussian sampling to produce exceptionally small signatures. These differences highlight the trade-offs between these schemes, such as implementation complexity, signature size, and computational efficiency.
Beyond their theoretical foundations, lattice-based schemes face practical challenges, including ensuring side-channel resistance, managing numerical stability, and optimizing implementations for various platforms. This section gives an overview of ML-DSA and Falcon, comparing their performance, security, and resistance to implementation vulnerabilities. The key attributes of these algorithms are provided in Table 4.

Attribute	ML-DSA (Dilithium)	Falcon
Security Assumption	Module-LWE, Module-SIS	NTRU
Statefulness	Stateless	Stateless
Public Key Size [byte]	1312/1952/2592	897/…/1793
Secret Key Size [byte]	2560/4032/4896	1281/…/2305
Signature Size [byte]	2420/3309/4627	666/…/1280
Performance (KeyGen)	?	?
Performance (Sign) [s]	1.1/1.8/2.2	8.8/…/13
Performance (Verify) [s]	0.2/0.4/0.7	0.1/…/0.3
Complexity of Implementation	Relatively Simple to Implement	Requires to implement the fast-Fourier (FFT) transform over floating point numbers
Side-Channel Resistance	Lack of heavy operations on the secret key	More complex operations on the secret key
Standard Security Levels [bits]	128/192/256	128/192/256
Design paradigm	Fiat-Shamir with Aborts	Hash-and-Sign paradigm with Gaussian Sampling
Memory Usage	Low	Higher (due to FFT and Gaussian Sampling)

Table 4: Overview of lattice-based signature schemes

On the Underlying Cryptographic Assumptions.
ML-DSA derives its security from the hardness of the Module Learning with Errors (Module-LWE) and Module Short Integer Solution (Module-SIS) problems. These problems generalize the widely studied LWE and SIS problems by introducing additional structure through modules, which allow for more efficient cryptographic operations without sacrificing security. The reduction of Module-LWE and Module-SIS to their standard lattice counterparts, LWE and SIS, provides strong evidence of the scheme's robustness against both classical and quantum adversaries.
Falcon, on the other hand, relies on the security of the NTRU lattice. NTRU lattices are known for their efficiency and compact representation. Decades of scrutiny and the lack of practical attacks against NTRU-based schemes provide confidence in Falcon's security. Gaussian sampling in Falcon further ensures that signatures are randomized and short. To use fast Gaussian sampling that results in small signatures, Falcon requires the use of a special power-of-two modulus. The downside of using such a modulus is the necessity to implement polynomial multiplication via a floating-point fast Fourier transform (FFT) which is problematic for platforms without support for the particular floating point arithmetic standard used in Falcon.

On Performance.
ML-DSA and Falcon differ significantly in performance and resource requirements, reflecting their underlying design goals. The timings in Table 4 are taken from the respective submission packages [Dilithium] and [Falcon]. ML-DSA’s public keys typically range in size from kilobytes, while signatures are around 2-3 kilobytes, depending on the parameter set. This makes ML-DSA suitable for applications where computational efficiency and simplicity are prioritized over minimizing data size. It should be highlighted that the signing time for ML-DSA is randomly variable. Depending on the case, the signing process may have to be repeated several times (c.f. [FIPS 204]). Falcon, in contrast, emphasizes compactness. Its public keys range between 0,9 to 1,7 kilobytes, and its signatures, measuring from 0,7 to 1,3 kilobytes. This makes Falcon highly appealing for bandwidth-constrained environments or scenarios where signature size is critical. However, Falcon’s reliance on computationally intensive operations, such as floating-point arithmetic, can lead to higher computational costs and increased complexity in implementation. Nevertheless, efficient implementation on

embedded devices is not only possible, but is shown to be very competitive [Pornin25]. While the standard has not been published yet, the small signature size and fast verification may compensate for more complex personalization devices.
Any signature algorithm includes a first step during which the message is hashed. Carrying out the hashing of the message in a HSM could impede the signature generation time due to the hashing time. Therefore, implementers should investigate whether hash computation inside the HSM is sufficient for their use case. On the other hand, in case of ML-DSA the recommended pure mode includes the hashing step as part of the signature creation algorithm and thus can only be implemented on the same device.

On Implementation Complexity and Side Channel Resistance.
ML-DSA operates using the Fiat-Shamir with Aborts paradigm, transforming a lattice-based identification scheme into a signature scheme. It employs rejection sampling to ensure valid signatures with the desired statistical properties. Randomness for signing can optionally be derived deterministically by hashing the private key, message, and a counter using a cryptographic hash function. This deterministic approach avoids reliance on true random number generators (RNGs), simplifying implementation and ensuring reproducibility. ML-DSA relies on modular arithmetic over finite fields, which is computationally efficient and well-supported by software and hardware libraries. However, the optional deterministic nature of its signing process introduces potential vulnerabilities to side-channel attacks, as attackers can observe patterns across multiple signing instances. To mitigate these risks, implementations must ensure constant-time operations for hash computations and modular arithmetic, employ masking techniques to protect intermediate values, and verify signatures to safeguard against fault injection attacks. While these countermeasures strengthen the scheme, they also introduce additional implementation complexity. Masking schemes, in particular, usually introduce high redundancy.

Falcon, in contrast, generates signatures using a randomized approach based on Discrete Gaussian Sampling. This randomness ensures high entropy and provides better resistance to side-channel attacks, as it is harder to correlate observations across multiple signing instances. While randomized signing improves side-channel resistance, the Discrete Gaussian Sampling process itself is a potential source of vulnerability and may be challenging to implement. Falcon's design produces much smaller signature sizes compared to ML-DSA. However, the security of the scheme relies on high-quality RNGs, and poor randomness can lead to vulnerabilities. Additionally, precision errors and numerical stability in FFT-based Gaussian sampling pose significant challenges, especially when ensuring constant-time execution to mitigate timing attacks. Furthermore, implementers must ensure that the proper standard for floating-point arithmetic is used in the implementation. Yet, the floating-point arithmetic is complex to implement and thus very prone to implementation errors which could ruin the overall security of key generation and signature computation. The experience and hindsight regarding the security against side channel and fault injection attacks of floating-point arithmetic is currently limited. The computation of the floating-point arithmetic used by Falcon requires a FPU (Floating Point Unit) when executed on hardware devices. However, many hardware devices (HSM or IC) do not have such FPU at the moment, depriving them from the possibility to implement key generation and signature computation with Falcon.

Note: There is currently little information available on whether the implementation of one of the quantum-safe signature algorithms on a HSM has undergone a security assessment. Issuing authorities and document manufacturers are encouraged to request their HSM suppliers to evaluate their products, whereby the evaluation of the quantum-safe algorithms should demonstrate a security strength comparable to that of HSMs with classic digital signature algorithms.

3.3 Impact on storage capacity of the chip

Signatures generated using quantum-safe algorithms are significantly larger than those created using classic cryptography. The same generally applies to public keys of quantum-safe algorithms.

In today's eMRTDs, the chip stores a signature over the Document Security Object (SOD) and the Document Signer, which includes the Document Signer's public key and a signature from the CSCA. Moving to a quantum-safe signature thus requires storing two quantum-safe signatures and one quantum-safe public key on the chip. The total size of these objects is summarized in the following table for each algorithm.

To give context, these values are compared against the typical size of Data Group 2 (DG2), which holds the facial image and is assumed with a size of 15 kilobytes. This comparison allows the reader to gauge the additional storage requirements and the potential increase in readout time introduced by a quantum-safe signature algorithm.

	ML-DSA	**Falcon**	**SLH-DSA**	**LMS**	**HSS**	**XMSS**	**XMSS^MT**
Quantum safe	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Security	Lattice	Lattice	Hash	Hash	Hash	Hash	Hash
Signature Size [byte]	3.309 / 4.627	N.A. / 1280	16.224 / 29.792	- / 2.832	- / 5.076	1.732 / 2.820	- / 4.963
Public Key Size [byte]	1.952 / 2.592	N.A. / 1793	48 / 64	48 / 56	52 / 60	52 / 68	52 / 68
Size of 2 Signatures & 1 Public Key [byte]	8.570 / 11.846	N.A. / 4.353	32.496 / 59.648	- / 5.720	- / 10.212	3.516 / 5.708	- / 9.994
Fraction of DG2 (á 15 kB)	0.56 / 0.77	N.A. / 0.28	2.11 / 3.88	- / 0.37	- / 0.66	0.22 / 0.37	- / 0.65
Standard	FIPS 204	TBS	FIPS 205	IETF RFC 8554, NIST SP 800-208			

Table 5: Comparison of quantum safe signature schemes

Note: In Table 5 a cipher suite is chosen for each algorithm matching a security strength of 192 and 256 bit:
ML-DSA: ML-DSA-65, ML-DSA-87 / Falcon: -, Falcon-1024 /
SLH-DSA : SLH-DSA-SHAKE-192s, SLH-DSA-SHAKE-256s /
LMS: LMS_SHA256_M32_H20, LMS_SHA256_M24_H20 /
HSS: LMS_SHA256_H10_W4_H10_W4 /
XMSS: XMSS-SHA2_20_192, XMSS-SHA2_20_256 /
XMSS^MT: XMSSMT-SHA2_20/2_256

The cipher suites shown in Table 5 are only a representative selection from the full range of possible configurations. Also, different signature algorithms may be used at the CSCA level and at the Document Signer level.
It is also important to highlight that EF.CardSecurity currently also contains the Document Signer certificate as well as a signature over its content. Thus, the storage capacity for EF.CardSecurity must also be taken into consideration.

3.4 Request for Information

States are invited to provide their opinion on quantum-safe signature algorithms.

Which digital signature algorithm(s) do you consider suitable for Passive Authentication and the Document Issuing PKI? Are there any reasons why you would not support a particular algorithm?

4. Options for PKI

In the eMRTD PKI each issuing State or organization operates a CSCA which serves as trust-anchor for any certificate issued by that state or organization within the PKI. Receiving States or organizations establish trust directly into these CSCA certificates. According to Doc 9303, there must be precisely one CSCA per issuing State or organization.

However, for the migration to quantum-safe cryptography, the PKI and its root CA must also use cryptographic keys suitable for the new quantum safe cryptographic algorithms. At the same time, the existing PKI together with the keys suitable for the classic algorithms must be maintained to ensure backward compatibility with existing Inspection Systems. Consequently, it is necessary to discuss how the term CSCA should be defined in the future and under which conditions it should be operated. In particular, it must be agreed on how CSCA can simultaneously treat cryptographic keys for multiple cryptographic algorithms.

It is obvious that the a CSCA issuing certificates using more than one signature algorithm can no longer be tied to a single key. Instead, the term CSCA should be understood in the sense of an authority under whose sovereignty several keys and several PKI branches are operated.

In order to allow for a backwards compatible transition, the CSCA of an issuing State or organization may issue separate certificates for classic and quantum-safe cryptography during the transition period. The number of certificates actively used to sign Document Signer should be limited to two by the CSCA, i.e., one for each signature algorithm. The same limitation should apply to any entity operating a Document Signer.

Disclaimer: This chapter is still at an early draft stage and may be subject to significant changes in future versions of this document, In particular, information on the following topics will likely be amended:

PKI operation method: Two separate PKIs		
	Definition	Two separate PKI branches are used: - The classic PKI which aims at supporting backwards compatibility - A second PKI which uses quantum safe cryptography Both branches issue their own certificates using only their supported signature algorithm. The certificates from both branches are not linked to each other. Optional: Each PKI branch will issue a cross-signing certificate that signs the root certificate of the other branch to provide a

		cryptographic link between the branches.
	Backwards Compatibility	Yes, any non-quantum safe Inspection System can use the classic PKI only.
	Non-separability	No
	Downgrade Protection	Limited: Inspection Systems must be aware of any Root-CA key operated under the authority of an CSCA. However, Inspection Systems have no means of determining whether an CSCA operates multiple keys and must rely solely on operational procedures.
	Security Properties	The certificate from the classic and the quantum safe branch are separable.
PKI operation method: Two linked PKIs issuing hybrid certificates.		
	Definition	Same as 'Two separate PKIs', but the two branches together issue a single hydrid certificate. Several implementation options for these certificates are currently available, which are discussed in chapter 6:
	Backwards Compatibility	Depends on the implementation of the hybrid certificates, c.f. chapter 6.
	Downgrade Protection	Depends on the implementation of the hybrid certificates, c.f. chapter 6.
	Security Properties	Depends on the implementation of the hybrid certificates, c.f. chapter 6.
PKI operation method: Single PKI with quantum safe cryptography support only		
	Definition	Only a single PKI branch is used, and it only supports quantum-safe cryptography. The existing PKI, which supports classic cryptography, will be discontinued following a direct transition to the quantum safe PKI.
	Backwards Compatibility	No, any IS not supporting quantum-safe signatures cannot verify them.
	Downgrade Protection	Yes
	Security Properties	The overall security of the signature depends entirely on the security properties of the quantum safe signature algorithm.

When keys for hybrid certificate are introduced in the PKI (CSCA and/or DS), it is recommended to apply the following security rules:

- The cryptographic keys linked to a hybrid-certificate (CSCA or DS-certificate) shall never be used alone. E.g., if a hybrid-certificate is associated with a ECDSA and a

ML-DSA private key, the ECDSA key must not be used alone create a create an ECDSA-only signature.

- If an object is to be signed using a hybrid signing algorithm, all higher-level entities in the PKI must also hold a hybrid certificate including the corresponding key. E.g., if an issuing authority wishes to issue a hybrid document security object (EF.SOD) the respective Document Signer and CSCA must both be provided with a hybrid certificate including the corresponding keys.

The option “Two separate PKIs” could also be combined with the option “Two linked PKIs issuing hybrid certificates”. Hereby, the second separate PKI-branch would consist of the two-linked PKI branch issuing hybrid certificates. This combination may be necessary if a genuine classic branch is required and at the same time the requirement for hybrid cryptographic algorithms must be met for the quantum-secure branch.

4.1 Request for Information

States are invited to provide their opinion on the operation of a PKI that supports quantum-safe signature algorithms.

What approach do you envision for operating the Document Issuing PKI during the transition to quantum-safe algorithms?

5. Options for hybridization of signatures and certificates

Security properties for cryptographic signature

As per ETSI TS 103 966 Annex B, the main security notion for digital signature algorithms is Existential Unforgeability under Chosen-Message Attack (EUF-CMA), which should be considered to set up hybridization strategies.

- **EUF-CMA security:** Given a public key and access to a signing oracle that will return a signature value for any message, the adversary tries to construct a valid signature value for a message that was not included in a request to the signing oracle.

Security properties for hybrid signature scheme

- **Separable:** Given a hybrid signature value, an adversary can produce a signature value that will be accepted by the verification algorithm for one of the component algorithms and contains no evidence that it was derived from a hybrid signature value.
- **Weakly non-separable:** Given a hybrid signature value, an adversary cannot produce a signature value that will be accepted by the verification algorithm for one of the component algorithms without it containing some evidence that it was derived from a hybrid signature value.
- **Strongly non-separable:** Given a hybrid signature value, an adversary cannot produce a signature value that will be accepted by the verification algorithm for one of the component algorithms.

Weak non-separability allows the verifier to detect when they are presented with part of a hybrid signature value. Strong non-separability prevents the verifier from accepting any partial signature value as valid.

5.1 Analysis of hybridization methods for cryptographic signatures stored in the Document Security Object

Hybridization should be implemented to securely bind the two cryptographic signatures stored in the Document Security Object: the classic ECDSA or RSA based signature and the quantum safe signature.

The purpose is to avoid an attacker to remove the stronger cryptographic signature to only present the weak or flawed cryptographic signature to the Inspection System, or at least ensure the Inspection System/Verifier is able to detect this manipulation.

Several cryptographic hybridization strategies of signatures are possible.

Hybridization Method: Concatenated		
	Definition	Two independent signatures are computed over the same message: One classic signature and quantum safe signature. The resulting signature objects are provided as concatenation in the signed data structure (e.g. as two subsequent SignerInfo objects).
	Backwards Compatibility	Yes, if Inspection System is able to ignore quantum safe signature.
	Downgrade Protection	No, each signature could be removed from Document Security Object.
	Security Properties	Signatures are separable.
	Options for Implementation	Concatenated signature objects do not require to specify a hybrid certificate format, but can be included in the existing SignedData Type structures already defined in Doc 9303. This option can only be used in conjunction with the “Two separate PKI” option.
Hybridization Method: Concatenated with identifier		
	Definition	Two independent signatures are computed over the same message: One classic signature and quantum safe signature. The messages signed by each signature algorithm includes an identifier to denote that each signature comes with another one.
	Backwards Compatibility	Yes, if Inspection System is able to ignore identifier[1] and quantum safe signature.
	Downgrade Protection	
	Security Properties	Signatures are weakly non-separable.
	Options for Implementation	Concatenated signature objects do not require to specify a hybrid certificate format. Yet, the existing specification from Doc 9303 need to be extended to specify the place and format to store the extra identifier (e.g. by supporting Certificate Discovery using RelatedCertificateDescriptor as described in 5.3.5) This option can only be used in conjunction with the “Two separate PKI” option.
Hybridization Method: Weak nesting (of classic signature)		
	Definition	A first signature is computed over the message.

[1] The identifier could be stored as part of the data in a data group that inspection systems ignore when it contains unknown information.

		A second signature is computed over the first signature.
	Backwards Compatibility	Yes, if Inspection System is able to ignore the outer quantum safe signature. In the context of eMRTD, to support backward compatibility, the first signature shall be the classic signature.
	Downgrade Protection	
	Security Properties	Classic signature is separable. Quantum safe signature is strongly non-separable. Weak nesting critically depends on the unforgeability of only the inner signature and not the outer signature algorithm.
	Options for Implementation	
Hybridization Method: Weak nesting (of classic signature) + classic signature includes an identifier		
	Definition	A first signature is computed over the message. A second signature is computed over the first signature. In addition, the first (classic) signature is also computed over an identifier which denotes it comes with a post quantum signature.
	Backwards Compatibility	Yes, if Inspection System is able to ignore identifier[1] and the outer quantum safe signature. In the context of eMRTD, to support backward compatibility, the first signature shall be the classic signature.
	Downgrade Protection	
	Security Properties	Classic signature is separable. Quantum safe signature is strongly non-separable. Weak nesting critically depends on the unforgeability of only the inner signature and not the outer signature algorithm.
Hybridization Method: Strong nesting (of classic signature)		
	Definition	A first signature is computed over the message. A second signature is computed over the message and the first signature.
	Backwards Compatibility	Yes, if Inspection System is able to ignore the outer quantum safe signature. In the context of eMRTD, to support backward compatibility, the first signature shall be the classic signature.
	Downgrade Protection	
	Security Properties	Classic signature is separable. Quantum safe signature is strongly non-separable. This technique produces a scheme where if either the first or second signature scheme is unforgeable, then the hybrid scheme is also unforgeable.[2]
Hybridization Method: Strong nesting (of classic signature) + classic signature includes an identifier		
	Definition	A first signature is computed over the message. A second signature is computed over the message and the first signature. In addition, the first (classic) signature is also computed over an

[2] N. Bindel, U. Herath, M. McKague, and D. Stebila, “Transitioning to a quantumresistant public key infrastructure,” Cryptology ePrint Archive, Tech. Rep. 460, 2017. [Online]. Available: http://eprint.iacr.org/2017/460

		identifier which denotes it comes with a post quantum signature.
	Backwards Compatibility	Yes, if Inspection System is able to ignore identifier[1] and the outer quantum safe signature. In the context of eMRTD, to support backward compatibility, the first signature shall be the classic signature.
	Downgrade Protection	
	Security Properties	Classic signature is weakly non-separable. Quantum safe signature is strongly non-separable. This technique produces a scheme where if either the first or second signature scheme is unforgeable, then the hybrid scheme is also unforgeable.[2]
Hybridization Method: True Hybrid		
	Definition	Both signature algorithms are intertwined so that they can't be separated. This has direct impact on the cryptographic algorithm implementation and does not support backward compatibility of classic signature. In addition, standards for intertwining such algorithms do not exist.
	Backwards Compatibility	None
	Downgrade Protection	
	Security Properties	Both signatures are strongly non-separable.
Hybridization Method: Two different data groups		
	Definition	The eMRTD provides two different Document Security Objects: One for the classic signature in the data group EF.SOD and another one for the quantum safe signature in a newly defined data group.
	Backwards Compatibility	Yes. Both signatures are independently encoded in two different files/data groups.
	Downgrade Protection	
	Security Properties	Inspection system/Verifier must be aware of additional data group to prefer the post-quantum signature.

5.2 Request for Information

States are invited to provide their opinion on the hybridization of signatures and their storage on eMRTD chips.

Which options for storing and distributing digital signatures on eMRTDs do you consider suitable to achieve both future-proofness and backward compatibility with existing non-quantum-safe Inspection Systems?

5.3 Certificate Formats for hybrid PKI

5.3.1 X.509 Certificate Structure

The certificates issued in the eMRTD Public key infrastructures (PKIs) rely on the **X.509 certificate format** according to [RFC 5280] as the fundamental data structure for

representing and authenticating public keys. X.509 provides a flexible mechanism to bind an entity's identity to a cryptographic key, together with metadata and policy information. X.509 certificate is defined in ASN.1 notation and encoded in **Distinguished Encoding Rules (DER)**. eMRTD PKI certificates include the following objects:

Certificate: ASN.1 SEQUENCE of

- **tbsCertificate:** (To Be Signed Certificate) This is the core payload of the certificate. It containing all of the fields that are signed by the certificate issuer.-
  - **version**: must be 3.
  - **serialNumber**: a unique identifier assigned by the issuing CA.
  - **signature**: Contains an *AlgorithmIdentifier* object specifying the algorithm (e.g., RSA, ECDSA) used by the CA to sign the certificate.
  - **issuer**: the distinguished name (DN) of the CA that issued the certificate.
  - **validity**: two dates (*notBefore* and *notAfter*) that define the certificate's validity period.
  - **subject**: the distinguished name of the entity to which the certificate is issued.
  - **subjectPublicKeyInfo**: a structured field containing:
    - an *AlgorithmIdentifier* (OID specifying the key type, e.g., rsaEncryption, id-ecPublicKey), and
    - the raw public key bits.
  - **extensions**: an extensible set of mandatory and optional fields according to the certificate profiles provided by Doc 9303-12. Common examples include:
    - *KeyUsage:* Defines the allowed purposes (e.g., encipherment, signature, certificate signing) of the key contained in the certificate.
    - *PrivateKeyUsagePeriod:* two dates that define the validity period for the private key.
    - …
- **signatureAlgorithm** This field repeats the *AlgorithmIdentifier* object specifying the algorithm (e.g., RSA, ECDSA) used by the CA to sign the certificate. It ensures consistency between what was declared in *tbsCertificate* and the actual signature method.
- **signatureValue** The actual digital signature value generated by the CA over the entire *tbsCertificate* structure. This binds all of the fields together and ensures authenticity and integrity.

Relevance for Hybrid Certificates:
The extensibility of the X.509 format — particularly through X.509 v3 extensions such as *Subject Alternative Name* or *Subject Information Access* — makes it possible to introduce post-quantum migration strategies without breaking compatibility with existing systems. By placing PQC-related material (alternative keys, signatures, or discovery metadata) inside *non-critical extensions*, legacy clients can safely ignore them, while PQC-aware clients can make use of the additional fields.
This design flexibility underlies the hybrid certificate approaches described in the following sections (e.g., ISARA Hybrid, Chameleon, Composite, and Certificate Discovery mechanisms).

5.3.2 subjectAltPublicKeyInfo / ISARA ("Hybrid Certificates" / Catalyst / AltPubKey)

The ISARA Catalyst Hybrid Certificate approach extends the X.509 v3 certificate format by embedding an alternative public key and alternative signature in *non-critical* extensions, alongside the classical cryptographic elements. This enables a single certificate to support

both a traditional (pre-quantum) algorithm and a post-quantum cryptographic (PQC) algorithm, while preserving backward compatibility.

Mechanism: Three dedicated non-critical extensions are added to the list of extension of the *tbsCertificate* portion:

- **subjectAltPublicKeyInfo:** contains the alternative (e.g., PQC) public key.
- **altSignatureAlgorithm:** specifies the algorithm identifier for the alternative signature.
- **altSignatureValue:** stores the signature produced using the alternative private key.

The classical signature covers nearly the entire certificate, including these new fields, while the alternative signature covers almost everything except the legacy signature.

Backwards Compatibility	Yes. When validating X.509 certificates, a verifier must ignore all non-critical extensions that it cannot process. Hereby, existing Inspection Systems would process only the SubjectPublicKeyInfo, signatureAlgorithm, and signatureValue fields holding the classic signature.
Security Properties for quantum-safe inspection systems	Allows an inspection system to learn that the issuer supports PQ. Classic certificate is weakly non-separable from PQ cert. A quantum-safe inspection system is able parse and verify both signatures from the certificate.
References	[ISARA] [X.509]

5.3.3 Chameleon Certificates

The Chameleon approach generalizes hybrid certificates by issuing two related certificates simultaneously:

- **Base certificate:** contains a classical public key and includes a non-critical *Descriptor* extension.
- **Delta certificate:** contains the PQC public key and is *reconstructable* from the Base certificate plus the Descriptor data.

Mechanism: The Descriptor is stored in the non-critical *DeltaCertificateDescriptor* extension of the base certificate. It encodes sufficient information for a quantum-safe capable Inspection System to reconstruct the Delta certificate.

Both certificates are generated in a single issuance process what ensures that identity and metadata are consistent, but also allows different keys, validity periods, or algorithms to be used.

Backwards Compatibility	Yes. When validating X.509 certificates, a verifier must ignore all non-critical extensions that it cannot process. For backwards compatability the embedded inner certificate must be the quantum-safe certificate. Existing Inspection Systems validating the certificate must ignore the non-critical DeltaCertificateDescriptor extension and only process the base certificate.
Security Properties for quantum-safe	Allows an inspection system to learn that the issuer supports PQ. Classic certificate is weakly non-separable from PQ cert. A quantum-safe inspection system is able to reconstruct the quantum-

inspection systems	safe certificate from the delta certificate stored in the DeltaCertificateDescriptor extension. Afterwards, it can verify both signatures from the certificate.
References	[ChameleonCert]

5.3.4 Composite Certificates / Composite Signatures

A composite signature scheme embeds multiple signatures (e.g., ECDSA and ML-DSA) into a single logical signature, usually requiring all to verify correctly.

Mechanism: Instead of adding extensions, composite certificates modify the core fields of X.509:

- **SubjectPublicKeyInfo**: contains an ASN.1 structure with multiple public keys.
- **signatureAlgorithm**: uses a composite algorithm OID.
- **signatureValue**: contains concatenated or structured component signatures.

Backwards Compatibility	No, Inspection Systems requires support for composite algorithm object identifiers and the respective signature algorithms.
Security Properties for quantum-safe inspection systems	Strongly non-separable certificate: Inspection Systems must always verify both signatures provided by the certificate, can detect wether one of the signatures has been removed and the certificate can only be compromised if both signature algorithms are broken. Existing certificate formats do not need to be modified and no reliance on custom extensions is required.
References	[CompositeCert]

5.3.5 Certificate Discovery with RelatedCertificateDescriptor

This method addresses certificate discovery rather than hybrid signing itself. It is especially useful when operating parallel certificate hierarchies (e.g., classical and PQC).

Mechanism: Uses the Subject Information Access (SIA) extension in X.509 to store a certDiscovery entry pointing to a related certificate. The RelatedCertificateDescriptor includes:

- A URI where the related certificate can be retrieved.
- Optional metadata: public key algorithm, signature algorithm, and a cryptographic hash for integrity binding.
-

Backwards Compatibility	Yes. When validating X.509 certificates, a verifier must ignore all non-critical extensions that it cannot process, i.e. also the RelatedCertificateDescriptor extension.
Security Properties for quantum-safe inspection systems	Allows an inspection system to learn that the issuer supports PQ. Weak non-separability to the related certificate by embedding data which announces the existence of a related certificate with a different algorithm. (At least) Two certificates required, that include the classic and the quantum-safe signature respectively. Both certificates (legacy and quantum-safe) contain evidence of each other.
References	[RelateCertDescr]

5.4 Options for storing quantum-safe signature on eMRTDs

The authenticity and integrity of the data stored on an eMRTD is protected by the Document Security Object, which is stored on the chip in the elementary file EF.SOD. The file contains hash values of all data groups stored on the chip, and is digitally signed by the issuing State or organization using their Document Signer. The file is specified in Doc 9303-10 and on comprises of a SignedData Type ASN.1 structure, as specified in [RFC 3369], Cryptographic Message Syntax (CMS).

- **SignedData**: ASN.1 Sequence
  - **version**: Must always be 3 for EF.SOD
  - **digestAlgorithms**: ASN.1 Set of all hashing algorithms used by any digital signature algorithm. These must by of type DigestAlgorithmIdentifier
  - **encapContentInfo**: The data to be signed. Includes the hash values over all data groups.
  - **certificates**: ASN.1 Set of all certificates required to validate the signature. Must include the Document Signer certificate.
  - **singerInfos**: ASN.1 Sequence of all SignerInfos objects (see below). Currently, it is recommended to provide only one SignerInfo object.
- **SignerInfo**:
  - **version**:
  - **sid**: Reference to the certificate used to create the signature stored in this SignerInfo object.
  - **digestAlgorithm**: hashing algorithm used by the digital signature algorithm used to create the signature stored in this SignerInfo object.
  - **signedAttrs** (optional):
  - **signatureAlgorithm**: digital signature algorithm used to create the signature value stored in this SignerInfo object
  - **signature**: the actual signature value created by the digital signature algorithm
  - **unsignedAttrs** (optional):

5.4.1 Option for composite and quantum-safe only certificates/siganture

For both signature types, composite signatures and quantum-safe only signatures the existing *SignedData* structure does not need to modified and does only require a single *SignerInfo* object inside the *singerInfos* field of *SignedData*, as recommended by Doc 9303. They also both rely on a single certificate that gets stored the certificates list of the *SignerData* object.

The composite or quantum-safe only signature algorithms are both indicated by an explicit object identifier, that is stored in the *signatureAlgorithm* value of the *SignerInfo* object. Both algorithms produce an opaque octet-string as signature value, that gets stored in the *signature* field of the *SignerInfos* structure and can be directly verified by a verifier supporting the respective algorithm.

Compatible PKI operation methods:

- Two PKIs issuing hybrid certificates
- Single PKI with quantum safe cryptography support only

Compatible certificates:
- X.509 certificates without extra non-critical extension

5.4.2 Option for ISARA hybrid certificates

If the CMS structure from EF.SOD gets signed by an ISARA X.509 hybrid certificate, the additional quantum-safe signature gets included into as additional attributes into the *SignerInfo* structure. The following two attribute are added to the *unsignedAttrs* of the *SignerInfo* structure:
- **altSignatureAlgorithm:** specifies the algorithm identifier for the quantum-safe signature.
- **altSignatureValue:** stores the signature value produced using the quantum-safe private key linked to the hybrid certificate.

The ISARA X.509 hybrid certificate gets stored in the *certificates* list of the *SignedData* structure.

Compatible PKI operation methods:
- Two PKIs issuing hybrid certificates

Compatible certificates:
- ISARA X.509 hybrid certificate

5.4.3 Option for multiple SignerInfo objects

According to Doc 9303-10 it is recommended to provide only one *SignerInfo* object in the *SignedData* structure of the Document Security Object (EF.SOD). Still, it is allowed to include multiple *SignerInfo* objects.

Hereby, each of these *SignerInfo* objects could provide the signature from a different certificate which use different signature algorithms. I.e., one *SignerInfo* could provide a classic signature while another *SignerInfo* provides a quantum-safe one. The respective Document Signer certificates used for signing must be stored on the *certificates* list of the *SignedData* structure.

One challenge of that option is that the behavior of Inspection Systems in the presence of multiple signatures where some of them are not verifiable (e.g., due to lacking algorithm support) is currently undefined.

Compatible PKI operation methods:
- Two PKIs issuing hybrid certificates
- Two separate PKIs
- Combination of options above

Compatible certificates:
- X.509 certificates without extra non-critical extension
- X.509 certificate with RelatedCertificateDescriptor extension
- Composite Certificates (for one the certificates)
- Combination of options above

5.4.4 Option for additional Document Security Object

The currently specified Document Security Object (EF.SOD) does only include the allowed classic signature algorithms as specified by the currently valid specifications in Doc 9303. Any signature produced by a quantum-safe or hybrid signing key gets stored in an additional document security object (e.g. EF.SOD2), that is stored in a different file on the eMRTDs chip. That file must be addressed by a file identifier currently not used by Doc 9303.
Full backwards compatibility is achieved since any inspection system not capable of processing quantum safe signature would simply not try to read the additional document security object EF.SOD2. Instead, it would just read the classic EF.SOD which only includes classic signatures as currently specified in Doc 9303.

Compatible PKI operation methods:

- Two PKIs issuing hybrid certificates
- Two separate PKIs
- Combination of options above

Compatible certificates (for additional document security object):

- X.509 certificates without extra non-critical extension
- X.509 certificate with RelatedCertificateDescriptor extension
- Composite Certificates (for one the certificates)
- Combination of options above

6. References

[HE2024] Houston-Edwards, Kelsey, Tomorrow's Quantum Computers Threaten Today's Secrets. Here's How to Protect Them, Scientific American, February 1, 2024, https://www.scientificamerican.com/article/tomorrows-quantum-computers-threaten-todays-secrets-heres-how-to-protect-them-2/

[CSAGPQC] ISO/IEC JTC 1/SC 17/WG 3 N 147. Developments regarding Cryptographic Agility and Post Quantum Cryptography for eMRTDs. Version 0.07

[AMG+16] M. Amy, O. D. Matteo, V. Gheorghiu, M. Mosca, A. Parent, and J. Schanck. Estimating the Cost of Generic Quantum Pre-image Attacks on SHA-2 and SHA-3. In Selected Areas in Cryptography - SAC 2016, Springer LNCS vol. 10532, pp. 317-337, 2016

[FIPS 204] NIST. Module-Lattice-Based Digital Signature Standard. https://doi.org/10.6028/NIST.FIPS.204

[FIPS 205] NIST. Stateless Hash-Based Digital Signature Standard. https://doi.org/10.6028/NIST.FIPS.205

[Dilithium] CRYSTALS-Dilithium – Algorithm Specifications and Supporting Documentation (Version 3.1) Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. https://pq-crystals.org/dilithium/index.shtml

[Falcon] Pierre-Alain Fouque Jeffrey Hoffstein Paul Kirchner Vadim Lyubashevsky Thomas Pornin Thomas Prest Thomas Ricosset Gregor Seiler William Whyte Zhenfei Zhang. Fast-Fourier Lattice-based Compact Signatures over NTRU. https://falcon-sign.info/ Specification document (update from February 2021). 2021-02-08

[NIST SP 800-208] NIST. Recommendation for Stateful Hash-Based Signature Schemes https://csrc.nist.gov/pubs/sp/800/208/final

[NIST SP 800-227] NIST. Recommendations for Key-Encapsulation Mechanisms. https://csrc.nist.gov/pubs/sp/800/227/ipd

[RFC3369] Internet Research Task Force. Cryptographic Message Syntax (CMS). https://datatracker.ietf.org/doc/rfc3369/

[RFC8554] Internet Research Task Force. Leighton-Micali Hash-Based Signatures. https://datatracker.ietf.org/doc/html/rfc8554

[RFC8391] Internet Research Task Force. XMSS: eXtended Merkle Signature Scheme. https://datatracker.ietf.org/doc/html/rfc8391

[CMP18] Laurent Castelnovi, Ange Martinelli, and Thomas Prest. Grafting Trees: A Fault Attack Against the SPHINCS Framework. In Tanja Lange and

Rainer Steinwandt, editors, Post-Quantum Cryptography, volume 10786, pages 165–184. Springer International Publishing, Cham, 2018. Series Title: Lecture Notes in Computer Science

[Gen23] Aymeric Genêt. On Protecting SPHINCS+ Against Fault Attacks. IACR Transactions on Cryptographic Hardware and Embedded Systems, pages 80–114, March 2023.

[GKPM18] Aymeric Genêt, Matthias J. Kannwischer, Herv´e Pelletier, and Andrew McLauchlan. Practical Fault Injection Attacks on SPHINCS. 2018. https://eprint.iacr.org/2018/674.

[ALCZ20] Dorian Amiet, Lukas Leuenberger, Andreas Curiger, and Paul Zbinden. FPGA-based SPHINCS+ Implementations: Mind the Glitch. In 2020 23rd Euromicro Conference on Digital System Design (DSD), pages 229–237, 2020

[BSIFault] BSI: Laser Fault Injection Attack on the eXtended Merkle Signature Scheme (XMSS). https://www.bsi.bund.de/EN/Service-Navi/Publikationen/Studien/LFI_Attack_XMSS/LFI_Attack_XMSS_node.html

[XMSSRef] https://github.com/XMSS/xmss-reference

[And97] Ross Anderson. Two remarks on public key cryptology. In Manuscript. Relevant material presented by the author in an invited lecture at the 4th ACM Conference on Computer and Communications Security, CCS, pages 1–4. Citeseer, 1997

[Pornin25] Falcon on ARM Cortex-M4: an Update. https://eprint.iacr.org/2025/123

[NIST-CSOR] https://csrc.nist.gov/projects/computer-security-objects-register/algorithm-registration

[WWO+23] Alexander Wagner and Vera Wesselkamp and Felix Oberhansl and Marc Schink and Emanuele Strieder. Faulting Winternitz One-Time Signatures to forge LMS, XMSS, or SPHINCS+ signatures. https://eprint.iacr.org/2023/1572

[SPHINCS+v31] Jean-Philippe Aumasson, Daniel J. Bernstein, Ward Beullens, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Andreas Hülsing, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, Bas Westerbaan. SPHINCS+ – Submission to the 3rd round of the NIST post-quantum project. v3.1. https://sphincs.org/resources.html

[X.509] X.509: Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks. 2019. https://www.itu.int/rec/T-REC-X.509

[ISARA] ISARA Catalyst Technical Overview. https://www.isara.com/catalyst

[ChameleonCert] A Mechanism for Encoding Differences in Paired Certificates. https://datatracker.ietf.org/doc/draft-bonnell-lamps-chameleon-certs/

[CompositeCert] Composite ML-DSA for use in Internet PKI. https://datatracker.ietf.org/doc/draft-ounsworth-pq-composite-sigs/

[RelateCertDescr] A Mechanism for X.509 Certificate Discovery. https://datatracker.ietf.org/doc/draft-ietf-lamps-certdiscovery/

[Siebren2024] Siebren Lepstra, Preparing Passports for the Post Quantum Era - Securing Travel Documents with Post Quantum Cryptography, 2024 https://cs.ou.nl/members/hugo/supervision/2024-siebren.lepstra-msc-thesis.pdf

Annex A: Performance of quantum safe signature algorithms

Performance measurements on ARM Cortex A72 (ARM v8). Implementation is based on library pqcrypto v0.18.1 (https://crates.io/crates/pqcrypto). Size of message to be signed: 90 bytes. Number of cycles: 10. Time in ms.

	ML-DSA-44	**ML-DSA-67**	**ML-DSA-87**
Key Gen	7	10	6
Sign	30	42	20
Verify	7	9	6

	Sphincs+-SHA2-128f	**Sphincs+-SHA2-128s**	**Sphincs+-SHA2-192f**	**Sphincs+-SHA2-192s**	**Sphincs+-SHA2-256f**	**Sphincs+-SHA2-256s**
Key Gen	25	1647	38	2415	99	1587
Sign	604	12528	1003	22184	2030	19629
Verify	36	12	53	18	54	27

	Sphincs+-SHAKE-128f	**Sphincs+-SHAKE-128s**	**Sphincs+-SHAKE-192f**	**Sphincs+-SHAKE-192s**	**Sphincs+-SHAKE-256f**	**Sphincs+-SHAKE-256s**
Key Gen	50	3249	74	4765	195	3135
Sign	1189	24691	1927	42852	3943	37360
Verify	71	24	103	35	105	51

	Falcon-512	**Falcon-512-padded**	**Falcon-1024**	**Falcon-1024-padded**
Key Gen	190	202	506	533
Sign	6	6	12	12
Verify	0	0	1	1