Post-Quantum Certificates for Electronic Travel Documents

Gaëtan Pradel[1,2] and Chris J. Mitchell[2]

INCERT, Luxembourg
gpradel@incert.lu
Information Security Group, Royal Holloway, University of London, UK
me@chrismitchell.net

Abstract. Public key cryptosystems play a crucial role in the security of widely used communication protocols and in the protection of data. However, the foreseen emergence of quantum computers will break the security of most of the asymmetric cryptographic techniques used today, including those used to verify the authenticity of electronic travel documents. The security of international borders would thus be jeopardised in a quantum scenario. To overcome the threat to current asymmetric cryptography, post-quantum cryptography aims to provide practical mechanisms which are resilient to attacks using quantum computers. In this paper, we investigate the practicality of employing post-quantum digital signatures to ensure the authenticity of an electronic travel document. We created a special-purpose public key infrastructure based on these techniques, and give performance results for both creation and verification of certificates. This is the first important step towards specifying the next generation of electronic travel documents, as well as providing a valuable test use case for post-quantum techniques.

Keywords: Post-Quantum Cryptography · Certificates · Electronic Travel Document · PKI

1 Introduction

Like many modern systems, the security of electronic passports and other electronic travel documents relies on public key cryptography. The idea of making travel documents *electronic*, i.e. by adding a chip, emerged in 1988 [14], although it wasn't until the late 1990s that electronic travel documents started to appear. A few years later, the International Civil Aviation Organization (ICAO) released design specifications to enable their authenticity to be verified worldwide [23], and shortly after, in 2004, the first ICAO compliant electronic travel document was issued [4]. Initiatives such as the US Visa Waiver Program (VWP) [45] helped their adoption by forcing member states to implement these specifications for their citizens' travel documents.

Starting with the work of Juels et al. [31], since 2005 a range of security analyses of the ICAO standards have been performed [21,39]. The feature in the

ICAO specifications which has gone through the most changes because of security issues covers access control to the chip see Avoine et al. [4] and Chaabouni and Vaudenay [10]. Versions of the ICAO access control protocol include the 2005 Basic Access Control (BAC) and the 2009 Extended Access Control (EAC) versions 1 and 2 [4].

Quite separately from the known issues with the ICAO protocols, the potential advent of large-scale, general-purpose, quantum computing will radically change the situation. Quantum computers can solve mathematical problems that classical computers cannot. Over the past few years, much effort has been devoted to building such a device, although experts in the field suggest that it will be one or two decades before large scale quantum computers are a reality [12]. In the post-quantum era, the currently used asymmetric cryptographic techniques, i.e. integer factorization-based schemes (such as `RSA` [42]) and discrete logarithm-based schemes [15]), will become breakable [40,43]. This threatens the security of a wide range of systems, including the authenticity of electronic travel documents (the main focus of this paper).

In order to address this issue, as summarised by Bernstein and Lange [7], much recent effort has been devoted to developing post-quantum cryptographic schemes, i.e. schemes secure against attack using both quantum and classical computers. In parallel with this research effort, a number of major standardisation bodies have inaugurated projects to develop standards for post-quantum algorithms. Perhaps the most important of these is the standardisation process led by the *National Institute of Standards and Technology* (*NIST*) [12]. So far, from an initial 82 submissions, after Round 3 of this process only 15[1] schemes remain in the running for adoption[2].

Besides having a portfolio of cryptographic algorithms resilient to cryptanalysis using quantum computers, it is also necessary to ensure that they are practical and can interoperate with current applications and protocols based on asymmetric cryptography. For example, Kampanakis et al. [32] showed that post-quantum X.509 certificates are viable for TLS-like communication protocols for use in a "post-quantum Internet". X.509 certificates are also commonly used to protect the authenticity and integrity of data inside electronic travel documents, namely the owner's data.

The focus of this paper is on a practical trial designed to test the feasibility of using currently available post-quantum cryptographic techniques in electronic travel documents. We have implemented a post-quantum *Public Key Infrastructure* (*PKI*) for electronic travel documents, and have also obtained results on its performance. Since this PKI is fundamental to the operation of security for electronic travel documents, the work described here can be seen as both preliminary research for the next generation of travel documents and also a testbed for evaluating post-quantum cryptographic techniques.

[1] More precisely, 7 schemes are finalists and the other 8 are kept as alternatives.

[2] The results of Round 3 of the process were published on July 22, 2020, https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions

In Section 2 we describe how security is implemented for electronic travel documents. We then explain the development of the prototype post-quantum PKI in Section 3 and present the challenges we encountered in Section 4. Finally, we discuss our results in Section 5 and draw conclusions in Section 6.

2 Security for electronic travel documents

2.1 Electronic travel documents

For the last couple of decades, digital signatures have been used to protect electronic travel and national identity documents. The ICAO started work on *Machine Readable Travel Documents* (*MRTDs*) as long ago as the late 1960s [23]. More recently, in 1998, work commenced on *electronic MRTDs* (*e-MRTDs*), resulting in a set of specifications covering the issue and border verification of such documents [23].

The specifications include protocols and mechanisms designed to protect the data inside the contactless chips embedded in the documents and allow border controllers to securely authenticate an issued e-MRTD. In order to verify an e-MRTD, the *inspection system* (*IS*) used by border controllers for validating the authenticity of an e-MRTD, must:

1. access the contactless chip (see §2.4), where the IS proves to the chip that it is authorised to access it;
2. authenticate the card data (see §2.5), where the IS verifies that the data inside the chip (including the information in the data page[3]) is digitally signed by an appropriate authority;
3. (optionally) authenticate the contactless chip (see §2.6), where the chip proves to the IS that it is a genuine chip belonging to a genuine e-MRTD;
4. (optionally) perform extended security protocols, e.g. to gain access to specific biometric data such as fingerprint or iris information.

2.2 Public Key Infrastructures

The security of e-MRTDs rests on an underlying PKI, the operation of which is the main focus of this paper. For our purposes a PKI (see, for example, Barak [5]) is a means of distributing trusted copies of public keys for asymmetric cryptographic techniques, and relies on the use of digital signatures. It involves a collection of *public key certificates*, digitally signed by *Certification Authorities* (*CAs*), where each certificate contains a public key and associated information including the name of the owner, who is usually assumed to have the private key corresponding to the public key in the certificate. Certificates which are no longer trusted are called revoked certificates, and are listed in a *Certificate Revocation List* (*CRL*) digitally signed by a CA.

[3] The document data page is the page containing personal information of the document owner, such as photograph, name, date of birth, etc.

The entities participating in a PKI can be arranged as the vertices in a directed graph, where an edge goes from A to B if the certificate for B (Cert_B) was signed using A's private signature key, i.e. so that the public key of A can be used to verify Cert_B. Typically, a PKI will be arranged hierarchically, so that there is always a direct path (a *certificate chain*) from the *Root CA* to every *end-entity*.

That is, if an entity has a trusted copy of the Root CA public key (typically distributed as a self-signed *Root CA certificate*), then a trusted copy of every end-entity's public key can be obtained in the following way. First construct a certificate chain from the Root CA to the end-entity, and then verify all the certificates in the chain in turn, at each stage verifying a certificate using the public key obtained by verifying the previous certificate and the status of the certificate using the corresponding CRL.

2.3 PKI for electronic travel documents

The PKI for e-MRTDs, including e-passports, typically has three levels. The Root CAs are known as *Country Signing Certification Authorities* (*CSCAs*), and, as the name suggests, are typically operated on behalf of a government department such as the Ministry of Foreign Affairs. Each country will operate a Root CSCA, and each such Root CSCA will have a digital signature key pair and a (self-signed) certificate for its public key, i.e. a public key certificate signed using the corresponding private key. A Root CSCA uses its private signing key to sign *Document Signer* (DS) *Certificates* (DSCs), containing public keys of e-MRTD manufacturers. The corresponding private signature keys are used by the manufacturers to sign information held inside an e-MRTD.

In order to prove the authenticity and integrity of an e-MRTD at a border control, the self-signed root CSCA certificates are shared among states by bilateral exchanges, through states' *Master Lists*[4] or soon using the ICAO *Public Key Directory*[5].

In a typical PKI the Root CA is kept offline in order to diminish the risks of a potential security breach that might lead to the leakage of its private signing key, whereas the Intermediate CAs are kept online. The reason is operational, as requests are sent on a daily basis to the CAs for issuing and revoking end-entity certificates. If and when an Intermediate CA certificate needs to be revoked/renewed because of a potential compromise or expiry of its private signing key, the Root CA is activated and its private signing key is used locally to issue a new Intermediate CA certificate or revoke the current one. In a PKI for e-MRTDs, the Root CSCA is also kept offline, in line with a recommendation by ICAO [26]. There are no Intermediate CSCAs because, from an operational point of view, this role is managed by the DSs, who receive all requests for signing e-MRTD data sets.

[4] For example the German Master List: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/ElekAusweise/CSCA/GermanMasterList.html.

[5] See https://pkddownloadsg.icao.int/.

2.4 Access to the contactless chip

The first step for an IS is to gain authorised access to the e-MRTD's chip. It proves to the chip that it has the necessary authorisation using one of the following two ICAO-specified protocols[6] [26]. To perform them, the IS shall have access to the *Machine Readable Zone* (*MRZ*) of the e-MRTD and be equipped to acquire it from the data page, requiring the passport to be physically opened to be read optically.

We only very briefly sketch the two protocols here, since they are not the main focus of this paper. As noted above in Section 1, the shortcomings of the first scheme have been widely discussed.

Basic Access Control Basic Access Control (BAC) is based on symmetric cryptography, and consists of a three-pass challenge-response protocol in accordance with Key Establishment Mechanism 6 of ISO/IEC 11770-2 [30] using two-key Triple-DES (see ISO/IEC 18033-3 [28]). A Message Authentication Code (MAC) is appended to the ciphertexts, computed using MAC algorithm 6 of ISO/IEC 9797-1 [29].

Password Authenticated Connexion Establishment Password Authenticated Connexion Establishment (PACE)[27] is based on asymmetric cryptography, and consists of a password-authenticated Diffie-Hellman key agreement protocol (see [9]) which supplements and enhances BAC. The chip verifies that the IS is authorised to access its data and a secure communications channel is established.

2.5 Authentication of the data

This step, i.e. authentication of the chip-resident data, forms the main focus of this paper; verifying the validity of the e-MRTD data is probably the most important security function. This step includes a single protocol called Passive Authentication (PA) [25], so called because it does not require any computational capabilities (such as performing cryptographic operations) from the chip.

However, the storage capacity of the chip is of paramount importance for executing PA, since the chip needs to keep certain key data. In particular, all the data related to the owner, such as the data page information, owner's photo and fingerprints, etc., are stored in *Data Groups* (*DGs*) on the chip [24]. Also stored is the *Document Security Object* (SO_D), which contains the hash values of the DGs and is digitally signed with the private key of a manufacturer. The corresponding manufacturer public key is in a DSC signed with the private key associated with a root CSCA certificate (belonging to the government agency on whose behalf the manufacturer is acting). The DSC must be placed in the SO_D

[6] Since January 2018, states have been permitted to implement PACE but not BAC, given the known security issues with BAC; previously both protocols had to be implemented for interoperability reasons.

so that the IS can retrieve it and use it to help verify this digital signature in order to verify the integrity and authenticity of the chip data [25].

In this paper, a key focus is the size of these data elements (see Section 5). Post-Quantum Digital Signature Algorithms (PQDSAs) usually involve longer keys and signatures than currently used techniques [7], and thus we need to investigate the limited storage capacity of current chips to discover if they are adequate for the post-quantum era.

The PKI described in §2.3 is used in the following way to support data authentication. The IS retrieves the signed data and the DSC from the chip. The IS determines which CSCA signed the DSC, and constructs a certificate chain from the appropriate Root CSCA certificate and the DSC. Verifying this chain (using the appropriate stored trusted Root CSCA public key) enables the appropriate DSC public key to be authenticated. Finally, this public key can be used to verify the signature on the chip data.

2.6 Authentication of the contactless chip

The third step for the IS is to authenticate the contactless chip, although this is not mandatory. This step enables the IS to verify that the chip is genuine, preventing copying and/or substitution; it uses one of the following three protocols.

As in §2.4, the protocols for this step are only briefly sketched, since they are not the main focus of this paper.

Active Authentication Active Authentication [25] is based on asymmetric cryptography and requires the chip to sign a challenge sent by the IS with a private key held by the chip. This means that the chip must have the computational power to perform a digital signature. The associated public key is accessible by the IS, and its authenticity has already been verified during PA (see §2.5). After verifying the signed challenge, the IS is assured of the authenticity of the chip. This technique can raise a privacy issue under specific conditions [31], as each generated signature could be logged. The owner of an e-MRTD (and thus the owner of the private key used to sign the challenges) could then be traced using the logged signatures. The Chip Authentication protocol (see below) has been devised as a replacement in order to mitigate this risk.

Chip Authentication Chip Authentication [25] is based on asymmetric cryptography, more precisely on a variant of the Diffie-Hellman key agreement protocol [15]. In addition to guaranteeing the authenticity of the chip, it also provides authentication of the data inside the chip and a secure communication channel between chip and IS. Moreover, as the exchanged keys are ephemeral, it prevents any tracing of the e-MRTD's owner [25]. The static key pair used in the protocol is stored inside the chip; the private key is held in secure memory whereas the public key is accessible to the IS. However, Chip Authentication is subject to reset and transferability attacks [8].

PACE with Chip Authentication Mapping PACE with Chip Authentication Mapping is a combination of PACE (§2.4) and Chip Authentication (§2.6), optimised for performance.

3 Building a post-quantum PKI for electronic travel documents

As discussed in §2.5, the authenticity of e-MRTD chip data is verified using the PA protocol. This protocol relies on the PKI established by states through their networks of CSCAs. Thus, to ensure that PA continues to provide security in the post-quantum world, a *post-quantum PKI* (*pqPKI*), i.e. a PKI based on the architecture presented in §2.3 but using post-quantum cryptography, is needed. To verify the practicality of building and operating such a PKI, we have built a proof-of-concept implementation which we next describe.

3.1 Design

For the purposes of this proof-of-concept, the PKI architecture for e-MRTDs as described in §2.3 can be simplified without loss of generality. The proof-of-concept PKI is composed of one CSCA certificate and one DSC.

Both types of certificate follow the standard structure for an X.509 certificate, signed using a PQDSA, e.g. as presented in [32], although the certificates must also be compliant with the relevant ICAO specification [26]. This latter specification defines the extensions and the associated values for each type of certificate in the e-MRTD PKI, with the details depending on their role in this structure, i.e. their *certificate profile*.

The CSCA certificate is self-signed and the associated private key is used to sign the private key associated with the DSC. The DSC is then normally used to sign an e-MRTD document; in our case this involves signing data of any type, ideally an $\mathrm{SO_D}$ (see Section 2).

3.2 Algorithm selection

Quantum computers pose a great threat to public key cryptography, including digital signature algorithms. Signature schemes usually use a hash function, which must remain secure in a post-quantum scenario [7]. We used the hash function that the designers included in their implementation, typically `SHA-3` [46].

To enable a comparison, we incorporated seven different PQDSAs into our prototype, all of which were candidates in Round 2[7] of the NIST standardisation process [12]. We chose these particular algorithms from the set of candidates for two main reasons: the cryptographic library we used (see §3.3) provides

[7] The experiments were run before the publication of the Round 3 which was announced on July 22, 2020.

implementations of these schemes, and as their security is not based on the same mathematical properties, they cover a broad range of the hard problems underlying post-quantum cryptography. The chosen algorithms[8] are as follows:

- `qTESLA` [3], which is a lattice-based digital signature scheme. The hardness assumption on which the security of `qTESLA` is based is the R-LWE problem [35,41].
- `CRYSTALS-Dilithium` [17], referred simply as `Dilithium` here, which is also a lattice-based digital signature scheme. The hardness assumption on which the security of `Dilithium` is based is the M-LWE problem [2,41].
- `Picnic` [11], whose security is not directly based on hardness assumptions, as is usually the case in public key cryptography. Its security is rather based on a zero-knowledge proof [20] and symmetric key primitives, which makes the scheme very different from the other examples.
- `FALCON` [18] is also a lattice-based digital signature scheme, based on the work of Gentry et al. [19]. Its hardness assumption is based on the Short-Integer-Solution (SIS) [1] problem over NTRU lattices [22].
- `MQDSS` [13] is based on the hardness of the multivariate quadratic problem.
- `Rainbow` [16] is also based on the hardness of the multivariate quadratic problem.
- `SPHINCS+` [6] is a set of three stateless hash-based signature schemes. The three schemes differ by the hash function used. We decided to use only two of the schemes, one instantiated with `SHAKE256` (which is part of the `SHA-3` family [46]) and the other with Haraka [34].

3.3 Implementation

To implement the prototype, we used a fork of OpenSSL combined with the library `liboqs` from the *Open Quantum Safe* (*OQS*) project [44]. OpenSSL is an open-source implementation of the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols, and incorporates a widely used cryptographic primitives library. It was not designed to establish PKIs, such as a PKI for e-MRTDs; however, despite this we decided to use this software because of its wide use and flexibility.

`liboqs` is an open-source library in `C` of post-quantum algorithms, which has been integrated into prototype forks of OpenSSL and OpenSSH. `liboqs` includes algorithms from the NIST Post Quantum Standardization Project. To generate the PKI for e-MRTDs described in §3.1, we implemented an OpenSSL configuration file that caused it to issue certificates with the appropriate extensions. The configuration file included all the certificate components and extensions needed by each certificate type, as defined in the relevant certificate profile, i.e. a CSCA certificate or a DSC, as specified in ICAO Doc 9303 Part 12 [26].

[8] Some of the chosen algorithms did not advance to Round 3 of the NIST competition. The results are published on the following website: https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

3.4 Overview of experiments performed

For each of the selected PQDSAs (together with two currently used schemes for comparison purposes), we performed the following steps.

1. We generated a key pair to be associated with the root CSCA certificate, with the parameter sets corresponding to the highest NIST security level available [38].
2. We generated a *Certificate Signing Request* (*CSR*) with the previously generated key pair and the CSCA certificate profile in order to create a (self-signed) CSCA certificate.
3. We generated a key pair to be associated with the DSC, with the parameter sets corresponding to the lowest NIST security level available [38].
4. We generated another CSR with the key pair generated in step (2) and the DS certificate profile to create a DSC signed using the CSCA private key generated in step (1).
5. We hashed and signed a random data string using the private key associated with the DSC to complete the chain.

The two key pairs generated in step (1) and (3) do not have the same parameter sets because their role is quite different (see [26] for more details). The key pair from step (1) has a long lifespan (it can be as much as one to two decades) and is used from time to time to verify or renew DSCs; thus a high NIST security level is required. The key pair from step (3) however has a much shorter lifespan (it can be days, weeks or months depending on the configuration chosen by the relevant state) and is used to sign the chip data of many e-MRTDs (during their production) in order to ensure their authenticity. To optimize performance, a lower NIST security level seems adequate.

4 Challenges

OpenSSL is an implementation of SSL/TLS, and is not designed to generate and manage a PKI producing certificates for signing e-MRTDs according to the relevant ICAO specifications [25,26]. For example, extensions such as *Private Key Usage period*, which are required by ICAO, cannot be set up with OpenSSL, although they can be displayed. To overcome this difficulty, we took advantage of the fact that OpenSSL allows integration of ad hoc extensions created by the user via the Arbitrary Extension module[9]. This allows an implementer to encode arbitrary extensions in created certificates[10].

A problem was encountered when trying to create a certificate chain. Although the software produced chains using well-established digital signature schemes, it refused to produce them for the chosen post-quantum algorithms. We reported the problem to the authors of the `liboqs` library, and simultaneously

[9] https://www.openssl.org/docs/manmaster/man5/x509v3_config.html

[10] An example of such an ad hoc extension is given at: http://openssl.6102.n7.nabble.com/Private-Key-Usage-Period-td28401.html

worked on a resolution. The issue has been resolved and the documentation for the software has been updated[11].

5 Results

We generated certificates according to the two certificate profiles described in Section 3 (CSCA certificate and DSC) for ten algorithms and two parameter sets, and in each case measured the memory needed to store them and their generation time. To perform the operations we used an Ubuntu 18.04.2 LTS $x86\_64$ machine with 8GB of RAM and a four-core Intel(R) Core(TM) i5-3470 CPU @ 3.20GHz. Two of the ten algorithms used were long-established signature schemes (RSASSA-PSS [37] and ECDSA [33] with the Brainpool parameters [36]), which were included for comparison purposes. We chose these algorithms because they are currently used by governments in their CSCAs[12] (see [4] for more details). The eight other algorithms were the PQDSAs presented in Section 3. Note that in the results reported below, certificate generation included both key pair generation and signing of the certificate, apart from Figures 1a, 1b and 1c for which the different steps of the certificate issuance process have been clearly separated. Table 1 summarises the algorithms and key lengths used for the two certificate types.

Table 1: Algorithms and key lengths by certificate type

	CSCA certificate	DSC
`RSASSA-PSS`	4096 bits with `SHA-256`	2048 bits with `SHA-256`
`Brainpool`	384 bits with `SHA-256`	224 bits with `SHA-256`
`qTESLA`	`qTESLA-p-III` with `SHAKE256`	`qTESLA-p-I` with `SHAKE128`
`Dilithium`	`Dilithium-4` with `SHAKE256`	`Dilithium-2` with `SHAKE128`
`Picnic`	`Picnic2-L5-FS` with `SHAKE256`	`Picnic2-L1-FS` with `SHAKE128`
`FALCON`	`FALCON-1024` with `SHAKE256`	`FALCON-1024` with `SHAKE256`
`MQDSS`	`MQDSS-31-64` with `SHAKE256`	`MQDSS-31-48` with `SHAKE256`
`Rainbow`	`Rainbow-Vc` with `SHA-512`	`Rainbow-Ia` with `SHA-256`
`SPHINCS+`	`SPHINCS-Haraka-256f-robust`	`SPHI NCS-Haraka-128f-robust`
`SPHINCS+`	`SPHINCS-SHAKE256-256f-robust`	`SPHINCS-SHAKE256-128f-robust`

To construct a post-quantum PKI, we separated certificate generation into three steps, according to the process described in §3.1, as follows:

1. generation of the key pair;
2. generation of the CSR; and

[11] See resolution in https://github.com/open-quantum-safe/openssl/issues/68

[12] In particular, the CSCA certificate from Luxembourg is signed using `RSASSA-PSS`. See https://repository.incert.lu/ for more details.

3. generation of the certificate (including the digital signature of the CSR generated in step (2) using the key generated in step (1)).

To be consistent with the associated certificate profile, the CSCA certificates were all self-signed and the DSCs were signed with a CSCA private key from the same algorithm family, e.g. a DSC including a `qTESLA-p-I` public key was signed with a `qTESLA-p-III` private key. We measured the execution time for 1000 iterations. The results are shown in Figures 1a, 1b and 1c respectively for each generation step. Figure 1d shows the total execution time for the three steps.

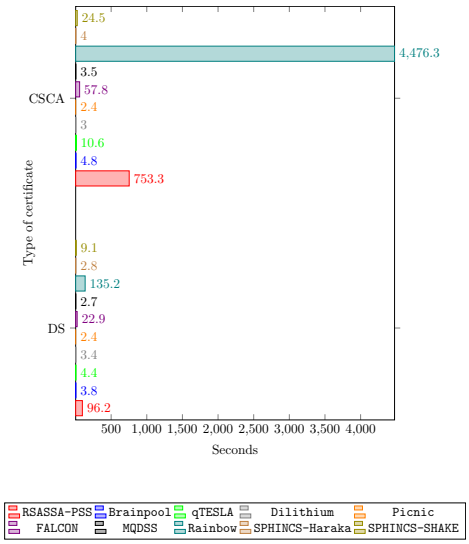
Overall, `Dilithium` is the best performing PQDSA, and even has a slightly better performance then `Brainpool`. The most secure version of `Rainbow` gave the worst performance, since key generation is very slow. Similar remarks apply to `RSASSA-PSS`. However, the less secure version of `Rainbow` performed better for the DSCs, as `SPHINCS-SHAKE` was slower. In particular, for `SPHINCS-SHAKE` and also `Picnic`, as expected their key generation is quite fast, although computing signatures is much slower than for their counterparts. All the other PQDSAs give results that are of the same order of magnitude, except `qTESLA` which gave similar results to `Dilithium` and `Brainpool`, but still performs much better than `RSASSA-PSS`.

In addition, we generated as many certificates as possible during a five-second period for each certificate profile, algorithm and key length. The results are exactly as expected based on Figure 1d. Again, `Dilithium` shows on average better performance than all the other schemes, with `Brainpool` and `qTESLA` almost as good. Because of slow key pair generation or signature computation, we managed to generate on average only a few CSCA certificates and DSCs with the four worst performing algorithms: `Rainbow`, `Picnic`, `SPHINCS-SHAKE` and `RSASSA-PSS`.

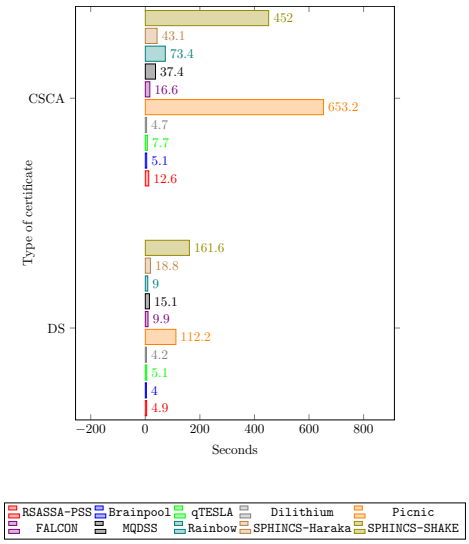
With respect to computation, the performance of several post-quantum schemes was actually superior to that of the existing algorithms. However, we also examined the memory space necessary to store the various certificates. This is a crucial point to consider in practice because of the limited memory capacity of contactless chips.

The certificates based on the two classical algorithms were significantly smaller than all of those based on post-quantum algorithms (see Figure 3a), although `Rainbow` and `Dilithium` yielded much shorter certificates than their peers. When considering the sizes of the generated key pairs (see Figure 3b), we obtained heterogeneous results. PQDSAs based on symmetric cryptography such as `Picnic` and `SPHINCS+` have extremely short keys. `Brainpool` has similar key sizes. As for `Rainbow`, both certificate and key sizes are much greater than any of the other algorithms. Figures 3a and 3b suggest that, as far as storage is concerned, `Falcon` is the best post-quantum candidate, with `Dilithium` not far behind.

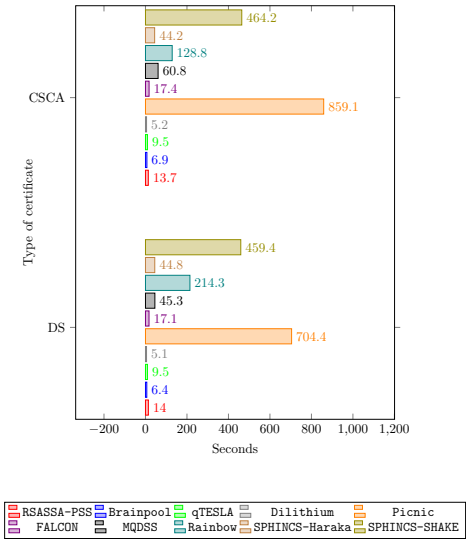
Over and above these somewhat abstract performance results, we wanted to consider how a switch to post-quantum algorithms would affect the "real world". That is, we wanted to assess the impact of a move to the post-quantum world on



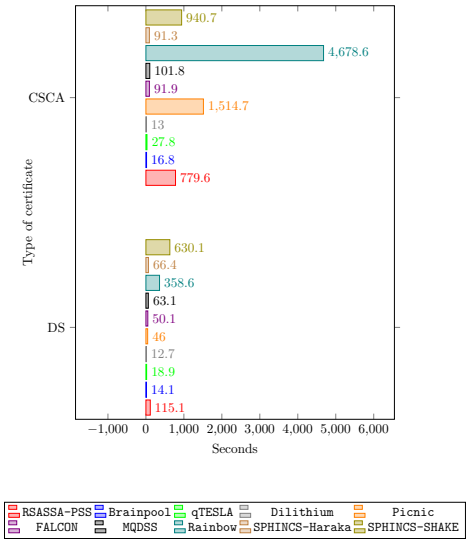
(a) Time in seconds to generate 1000 key pairs



(b) Time in seconds to generate 1000 CSRs



(c) Time in seconds to generate 1000 certificates



(d) Total time in seconds

Fig. 1: Performance results

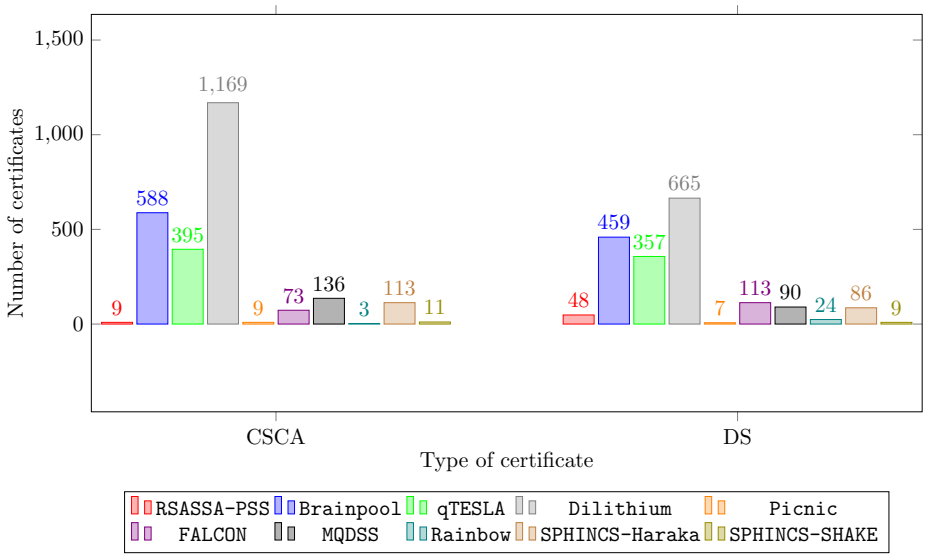


Fig. 2: Throughput of certificates in a five-second period

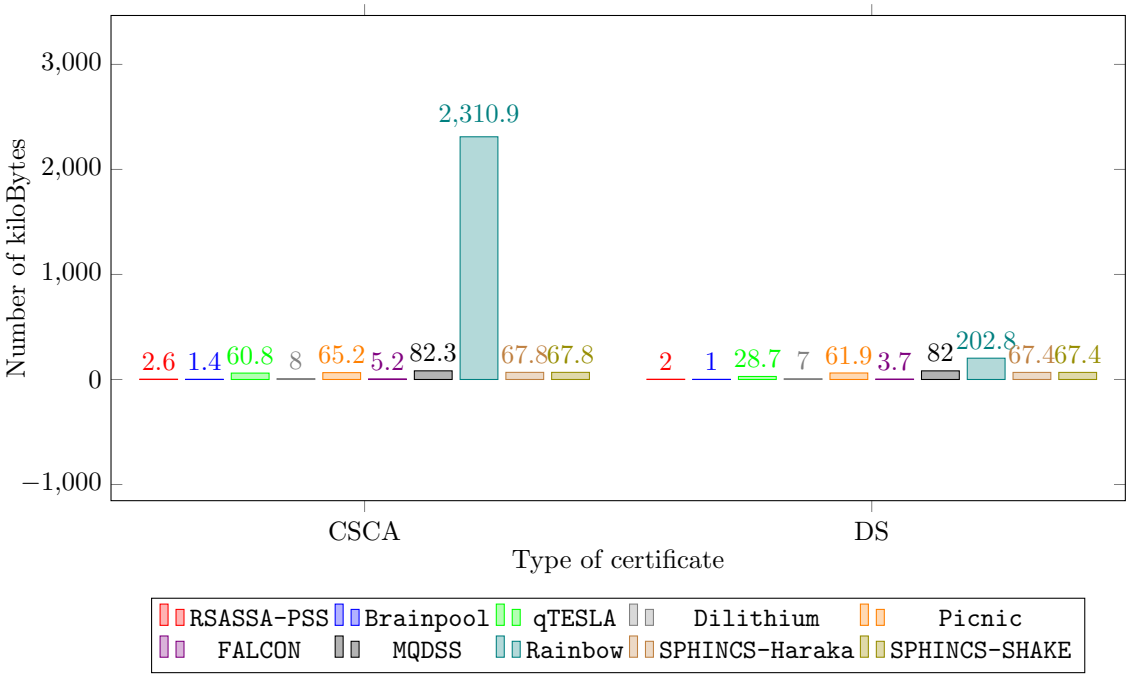
the generation and management of CSCA certificates and DSCs for government authorities.

We used as an example Luxembourg, in which the management of the PKI for generating the digital signatures of e-MRTDs has been assigned to a public agency[13] under the authority of the Ministry of Economy and the Ministry of Foreign Affairs.

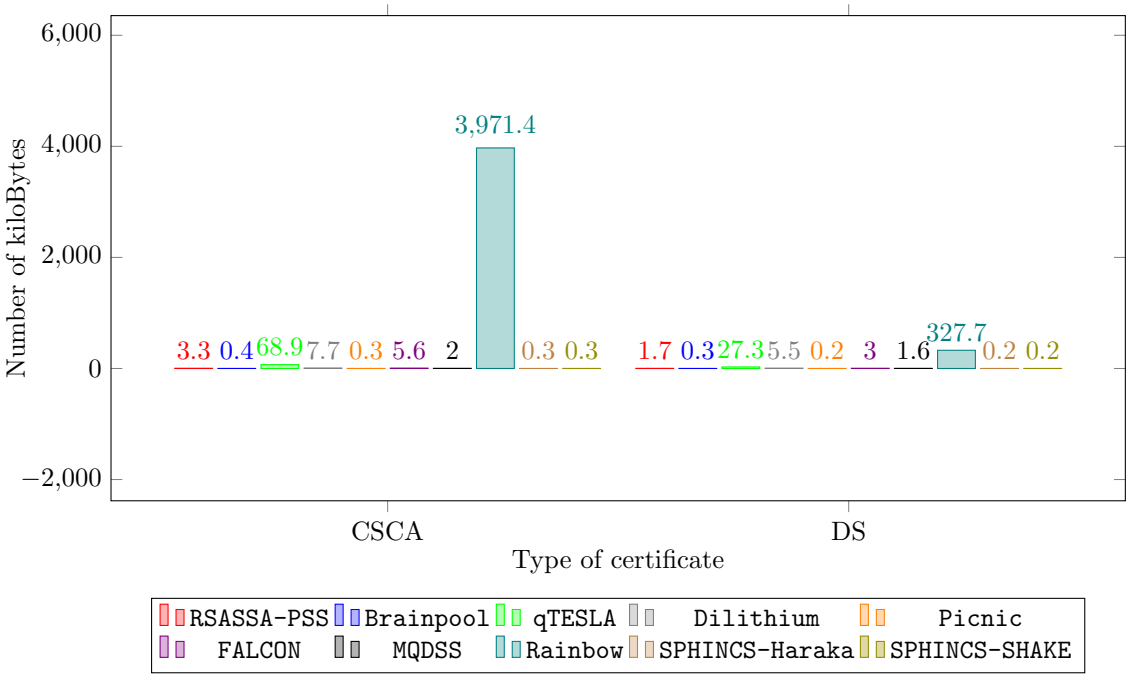
We did not consider the computational power of the contactless chips inside the e-MRTDs, but rather the memory space available in those chips (see Section 2). As explained above, the computational power of the contactless chips is not the main focus here. Such issues are the responsibility of the e-MRTD and chip manufacturers, and a CSCA, as a purchaser, has little say over such detailed design matters.

Typically, the infrastructure of a PKI is based on servers and hardware security modules, and can be arbitrarily expanded. CSCA certificates are re-issued every 3 to 5 years [26], and thus any cost change for CSCA certificate generation in terms of performance and memory will not be an issue. Two criteria are used to determine when DSCs must be renewed: their lifespan and the number of signatures performed. As best practice, both should be kept low. Typical limits might be a lifespan of at most one month and a limit of 100 000 e-MRTDs. In the case of Luxembourg, with only around 600 000 inhabitants of which only half are citizens, we can assume that the production of e-MRTDs is much less than many other countries. With this number of citizens, each DSC is most unlikely to reach

[13] https://www.incert.lu



(a) Size of certificates in kiloBytes



(b) Size of key pairs in kiloBytes

Fig. 3: Memory space results

the threshold of 100 000 digital signatures. For the algorithms we examined, we have disparate results. For some PQDSAs, both key generation and signing are faster than for the classical schemes, although this was not universally true.

When considering the memory space capacities of the contactless chip inside e-MRTDs, we need only to check that the chip provides enough memory space to store the post-quantum certificates and signatures necessary to perform PA, a protocol which does not require any computational power from the chip (see Section 2.5). Current chips[14] for e-MRTDs can have as much as 160 Kbytes of EEPROM memory and 280 Kbytes of User ROM. This would be large enough to store a post-quantum certificate and digital signatures for all of the PQDSAs we studied except `Rainbow`.

If we consider both computational and storage requirements, `Dilithium` offers the best performance overall. `Falcon` has very low storage requirements, but performs worse than `Dilithium` computationally. In particular, for the CSCA certificates `Dilithium` performed 7 times faster than `Falcon`, and for the DSCs, 4 times (see Figure 1d)[15]. Also, `Dilithium` offers computational performance comparable to the elliptic curve scheme.

6 Conclusions and future work

As in the work of Kampanakis et al. [32], the results of this paper show that post-quantum X.509 certificates can be used in current applications such as e-MRTDs. We used the seven different post-quantum digital signature algorithms as examples in our proof-of-concept, and showed that the performance for key generation and digital signature is clearly good enough (for most of the PQDSAs) to replace classical cryptographic asymmetric techniques (namely `RSASSA-PSS` and `Brainpool`), a result that is particularly clear in the case of `Dilithium`. At the same time, whilst memory requirements increase, the change is not sufficiently large to make the algorithms impractical. Of course, e-MRTDs produced with a post-quantum digital signature algorithm such as those used in our experiments would not be compliant with ICAO Doc 9303 Part 12 [26] which defines the algorithms to be used. Moreover, ICAO defines the minimum chip size as 32 Kbytes [24]. Based on our results, none of the post-quantum certificate would fit in a chip which provides this memory space. ICAO will have to update their specifications for the post-quantum era in order to ensure the security of e-MRTDs.

For this feasibility test of post-quantum PKI for e-MRTDs, we decided to use OpenSSL for implementation flexibility and ease of use, but this tool is not optimised or even designed for such a specific PKI. Possible future work includes use of JMRTD[16], an open source Java implementation for MRTD standards. This

[14] See for example these contactless cryptocontrollers: https://www.infineon.com/cms/en/product/security-smart-card-solutions/security-controllers/sle-78/.

[15] These results are in line with the NIST Round 3 statement: https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf.

[16] https://jmrtd.org/

tool uses *The Legion of the Bouncy Castle*, a cryptographic techniques library which has included `qTESLA` since 2019[17].

The next generation of e-MRTDs will be based on post-quantum cryptographic techniques, but no such documents have yet been issued, as far as we are aware. This paper focuses only on one of the three steps verifying the authenticity of an e-MRTD, but the other two steps also require cryptographic asymmetric techniques that will need to be quantum-resistant.

Finally, governmental authorities managing a CSCA usually manage another type of CA, known as the *Country Verifying Certification Authority* (*CVCA*). A CVCA is used to issue *Card Verifiable Certificates* (*CVCs*) to control authorities (such as the national police) so they can access, using the EAC protocol, fingerprint and/or iris data (if present) held in the controlled area of an e-MRTD. Further experiments will also be required to check possible post-quantum migration paths for this class of lightweight certificates.

7 Acknowledgements

Supported by the Luxembourg National Research Fund (FNR) (12602667).

References

1. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: STOC. pp. 99–108. ACM (1996)
2. Albrecht, M.R., Deo, A.: Large modulus ring-lwe $\geq$ module-lwe. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 10624, pp. 267–296. Springer (2017)
3. Alkim, E., Barreto, P.S.L.M., Bindel, N., Longa, P., Ricardini, J.E.: The lattice-based digital signature scheme qtesla. Cryptology ePrint Archive, Report 2019/085 (2019)
4. Avoine, G., Beaujeant, A., Hernandez-Castro, J., Demay, L., Teuwen, P.: A survey of security and privacy issues in epassport protocols. ACM Comput. Surv. **48**(3), 47:1–47:37 (2016)
5. Barak, B.: The complexity of public-key cryptography. In: Tutorials on the Foundations of Cryptography, pp. 45–77. Springer International Publishing (2017)
6. Bernstein, D.J., Hülsing, A., Kölbl, S., Niederhagen, R., Rijneveld, J., Schwabe, P.: The sphincs$^+$ signature framework. In: ACM Conference on Computer and Communications Security. pp. 2129–2146. ACM (2019)
7. Bernstein, D.J., Lange, T.: Post-quantum cryptography — dealing with the fallout of physics success. Cryptology ePrint Archive, Report 2017/314 (2017)
8. Blundo, C., Persiano, G., Sadeghi1, A.R., Visconti, I.: Resettable and non-transferable chip authentication for e-passports. In: Workshop on RFID Security (RFIDSec 2008) (2008)
9. BSI: Elliptic curve cryptography. Technical guideline, Federal Office for Information Security, Bonn, Germany (2018)
10. Chaabouni, R., Vaudenay, S.: The extended access control for machine readable travel documents. In: BIOSIG. LNI, vol. P-155, pp. 93–103. GI (2009)

[17] Please see https://www.bouncycastle.org/releasenotes.html.

11. Chase, M., Derler, D., Goldfeder, S., Orlandi, C., Ramacher, S., Rechberger, C., Slamanig, D., Zaverucha, G.: Post-quantum zero-knowledge and signatures from symmetric-key primitives. In: ACM Conference on Computer and Communications Security. pp. 1825–1842. ACM (2017)
12. Chen, L., Jordan, S., Liu, Y.K., Moody, D., Peralta, R., Perlner, R., Smith-Tone, D.: Report on post-quantum cryptography. Report, US Department of Commerce, National Institute of Standards and Technology (2016)
13. Chen, M., Hülsing, A., Rijneveld, J., Samardjiska, S., Schwabe, P.: From 5-pass MQ -based identification to MQ -based signatures. In: ASIACRYPT (2). Lecture Notes in Computer Science, vol. 10032, pp. 135–165 (2016)
14. Davida, G.I., Desmedt, Y.: Passports and visas versus IDS (extended abstract). In: EUROCRYPT. Lecture Notes in Computer Science, vol. 330, pp. 183–188. Springer (1988)
15. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Information Theory **22**(6), 644–654 (1976)
16. Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: ACNS. Lecture Notes in Computer Science, vol. 3531, pp. 164–175 (2005)
17. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-dilithium: A lattice-based digital signature scheme. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2018**(1), 238–268 (2018)
18. Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: Falcon: Fast-fourier lattice-based compact signatures over ntru (2017)
19. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC. pp. 197–206. ACM (2008)
20. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems (extended abstract). In: STOC. pp. 291–304. ACM (1985)
21. Hoepman, J., Hubbers, E., Jacobs, B., Oostdijk, M., Schreur, R.W.: Crossing borders: Security and privacy issues of the european e-passport. In: IWSEC. Lecture Notes in Computer Science, vol. 4266, pp. 152–167. Springer (2006)
22. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: ANTS. Lecture Notes in Computer Science, vol. 1423, pp. 267–288. Springer (1998)
23. International Civil Aviation Organization (ICAO): Doc 9303 — Machine Readable Travel Documents — Part 1: Introduction. Tech. rep., ICAO, Montréal, CA (2015), Seventh Edition
24. International Civil Aviation Organization (ICAO): Doc 9303 — Machine Readable Travel Documents — Part 10: Logical Data Structure (LDS) for Storage of Biometrics and Other Data in the Contactless Integrated Circuit (IC). Tech. rep., ICAO, Montréal, CA (2015), Seventh Edition
25. International Civil Aviation Organization (ICAO): Doc 9303 — Machine Readable Travel Documents — Part 11: Security Mechanisms for MRTDs. Tech. rep., ICAO, Montréal, CA (2015), Seventh Edition
26. International Civil Aviation Organization (ICAO): Doc 9303 — Machine Readable Travel Documents — Part 12: Public Key Infrastructure for MRTDs. Tech. rep., ICAO, Montréal, CA (2015), Seventh Edition
27. International Civil Aviation Organization (ICAO): Supplemental Access Control for Machine Readable Travel Documents. Tech. rep., ICAO, Montréal, CA (2015), Version 1.01

28. ISO Central Secretary: Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers. Standard ISO/IEC 18033-3:2010, International Organization for Standardization, Geneva, CH (2010)
29. ISO Central Secretary: Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher. Standard ISO/IEC 9797-1:2011, International Organization for Standardization, Geneva, CH (2011)
30. ISO Central Secretary: IT Security techniques — Key management — Part 2: Mechanisms using symmetric techniques. Standard ISO/IEC 11770-2:2018, International Organization for Standardization, Geneva, CH (2018)
31. Juels, A., Molnar, D., Wagner, D.A.: Security and privacy issues in e-passports. In: First International Conference on Security and Privacy for Emerging Areas in Communications Networks, SecureComm 2005, Athens, Greece, 5-9 September, 2005. pp. 74–88. IEEE (2005)
32. Kampanakis, P., Panburana, P., Daw, E., Geest, D.V.: The viability of post-quantum x.509 certificates. Cryptology ePrint Archive, Report 2018/063 (2018)
33. Kerry, C.F., Secretary, A., Director, C.R.: FIPS PUB 186-4 Digital Signature Standard (DSS) (2013)
34. Kölbl, S., Lauridsen, M.M., Mendel, F., Rechberger, C.: Haraka v2 - efficient short-input hashing for post-quantum applications. IACR Trans. Symmetric Cryptol. **2016**(2), 1–29 (2016)
35. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 6110, pp. 1–23. Springer (2010)
36. Merkle, J., Lochter, M.: Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation. RFC 5639 (Mar 2010)
37. Moriarty, K.M., Kaliski, B., Jonsson, J., Rusch, A.: PKCS#1: RSA Cryptography Specifications Version 2.2. RFC 8017 (Nov 2016)
38. National Institute of Standards and Technology: Submission requirements and evaluation criteria for the post-quantum cryptography standardization process. Report, US Department of Commerce (December 2016)
39. Pasupathinathan, V., Pieprzyk, J., Wang, H.: Security analysis of australian and E.U. e-passport implementation. Journal of Research and Practice in Information Technology **40**(3), 187–206 (2008)
40. Proos, J., Zalka, C.: Shor's discrete logarithm quantum algorithm for elliptic curves. Quantum Information & Computation **3**(4), 317–344 (2003)
41. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC. pp. 84–93. ACM (2005)
42. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
43. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**(5), 1484–1509 (1997)
44. Stebila, D., Mosca, M.: Post-quantum key exchange for the internet and the open quantum safe project. In: SAC. Lecture Notes in Computer Science, vol. 10532, pp. 14–37. Springer (2016)
45. United States Department of Homeland Security: United states customs and border protection: Visage waiver passport requirements (October 2006)
46. U.S. DoC/NIST: Sha-3 standard: Permutation-based hash and extendable-output functions. Standard, National Institute for Standards and Technology (2015)