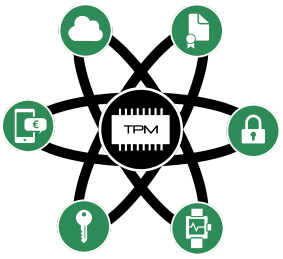

FutureTPM

D2.2

Second Report on New QR Cryptographic Primitives

Project number:	779391
Project acronym:	**FutureTPM**
Project title:	Future Proofing the Connected World: A Quantum-Resistant Trusted Platform Module
Project Start Date:	1st January, 2018
Duration:	36 months
Programme:	H2020-DS-LEIT-2017
Deliverable Type:	Report
Reference Number:	DS-LEIT-779391 / D2.2 / v1.0
Workpackage:	WP 2
Due Date:	December 31, 2019
Actual Submission Date:	December 31, 2019
Responsible Organisation:	IBM
Editor:	Bertram Poettering
Dissemination Level:	PU
Revision:	v1.0
Abstract:	This document describes the choices (and justification) of the public, symmetric and privacy-enhancing primitives for the TPM constructions.
Keywords:	Foundational primitives, basic protocols

DRAFT

The project FutureTPM has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 779391.

Editor

Bertram Poettering (IBM)

Contributors (ordered according to beneficiary numbers)

TEC, SURREY, RHUL, IBM, UB, IFAG, UL, INESC-ID

DRAFT

Disclaimer

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The content of this document reflects only the author's view – the European Commission is not responsible for any use that may be made of the information it contains. The users use the information at their sole risk and liability. This document has gone through the consortium's internal review process and is still subject to the review of the European Commission. Updates to the content may be made at a later stage.

Contents

DRAFT

1 Introduction

1.1 Structure of Document

The goal of WP2 of the FutureTPM project is the identification of cryptographic primitives and schemes that can serve in the security foundation of quantum-resistant TPMs. The predecessor of the current document, Deliverable D2.1 [30] of the same project, lays out the general scenery, including descriptions of various attack strategies that quantum adversaries might pursue, and methods to overcome these. The current document builds on the results of [30], and distills from them concrete recommendations regarding which schemes to implement in the FutureTPM prototypes (see WP5).

This document describes schemes that we recommend to implement in software or hardware: In Sections 2 and 3 we describe the hash functions and block ciphers, in Section 4 we describe modes of operations (of hash functions and block ciphers), in Sections 5 and 6 we describe the public key encryption primitives and signature schemes. Our evaluations of Sections 5 and 6 involved studying the progress made in the currently ongoing NIST PQC Standardization process.[1] For reference we give a full overview of all (remaining) NIST candidate schemes in Appendix A. Finally, in Appendix B, we describe our progress with the construction of a quantum-resistant DAA scheme. A detailed description and analysis of this construction will appear in Deliverable D2.3.

1.2 Assessment Approach

The security analysis of the symmetric components (i.e., of hash functions, block ciphers, and modes of operation) is fairly straight-forward, also in a setting involving quantum adversaries. The security of our lattice-based signature and DAA schemes is analyzed in the ROM, or in the QROM under slightly stronger (or less tight) assumptions. The reason for our stronger assumptions for signature and DAA schemes is that the most efficient constructions have an underlying protocol whose classical security proof requires the reprogramming of the random oracle that is accessed by the adversary. Since reprogramming (or rewinding), in the strict sense, is generally not possible if the adversary is quantum, the security proof is no longer applicable.

There have been recent works that give evidence that (lattice-based signature and DAA) schemes proven in the ROM are still fundamentally secure in the QROM. Unruh [130] and Kiltz et al. [74] showed that the zero-knowledge property of these protocols is still retained and that the security of the scheme can be tightly based on a stronger quantum assumption involving the random oracle and the underlying hard mathematical problem. Don et al. [39] showed that if the underlying mathematical problem satisfies an additional natural property, then the scheme is again secure in the quantum random oracle model. And Qin and Zhandry [79] showed that one can prove security of the protocol under the usual assumptions in the QROM under a rather loose reduction. These results give further more affirmation that there is nothing fundamentally insecure about the construction of any natural scheme built via the Fiat-Shamir framework whose security can be proven in the ROM. In our opinion, evidence is mounting that the distinction between schemes secure in the ROM and QROM will soon become treated in the same way as the distinction between schemes secure in the standard model and ROM – there will be some theoretical differences, but security in practice will be similar.

[1] https://csrc.nist.gov/Projects/post-quantum-cryptography/

DRAFT

1.3 Selection Criteria

This document recommends a number of cryptographic primitives for the use in the FutureTPM project. In the upcoming sections we explain our choice of algorithms for hash functions, block ciphers, modes of operation, public key encryption, and signature schemes, one by one. The symmetric primitives were chosen such that they are simultaneously quantum resistant, as compatible with prior TPM standards as possible, and meeting the following general set of additional properties:

- support from academia: we focus on mature, well-known algorithms which lack known vulnerabilities that may pose a cryptographic risk in a post-quantum setting;
- support from industry: we focus on schemes that are widely supported by software cryptographic libraries and vendors;
- support from standardization bodies: we focus on schemes that are recognized and recommended by international organizations and standardization bodies;
- performance: we focus on schemes that provide good performance in both software and hardware;
- intellectual property: we focus on schemes that are not covered by patents.

Only some of these metrics apply to the asymmetric primitives that we are recommending, simply because research in this area is still ongoing and has not yet settled. However, as our choice is based on the (intermediate) results of the NIST PQC standardization effort, support by at least one standardization body, good performance, and the freeness from intellectual property claims can be expected to be given.

Our selection was also based on diversity: In each category we propose schemes and algorithms from different families, for two main reasons: (1) In general, diversity is good for security, as advances in cryptanalysis allow for a quick informed decision towards an alternative scheme. This aspect seems particulary relevant for the schemes considered in Sections 4–6. (2) Choosing algorithms from different families further allows us to conduct an in-depth comparison of implementational efficiency (in collaboration with WP5 of the project).

DRAFT

2 Hash Functions

Deliverable D2.1 [30] reviews the state of the art of the algorithms under consideration, and presents a brief review of known attacks against them, mainly targeting round-reduced versions. We recall that from a post-quantum perspective, the QS1 and QS2 notions of security coincide for hash functions. In general terms, for a family of hash functions with output size n bits, Grover's algorithm [56] finds a preimage using $\mathcal{O}\left(2^{n/2}\right)$ quantum queries, whereas the Brassard-Høyer-Tapp (BHT) algorithm [27] requires only $\mathcal{O}\left(2^{n/3}\right)$ quantum queries to produce a collision with overwhelming probability.[2] Hence, targeting a minimum security level of 128 bits implies a minimum output size of either 256 or 384 bits, depending on what security property is relevant in a given post-quantum context.
When selecting candidates for our recommendation, we ensured that all hash functions currently listed in the TCG Algorithm Registry [126] that we assess to be quantum-secure are also recommended in this report. This is to maximize compatibility with existing implementations, and to enable a smooth transition from classic to post-quantum primitives. We acknowledge that legacy requirements of specific applications might require the employment of other hash functions from the TCG Algorithm Registry. While these will not be quantum secure *in general*, in specific cases in which the results of Grover and BHT are not applicable they might be. The assessment of such cases requires a very careful analysis and is beyond the scope of this report.
Note that Deliverable D2.1 [30] lists the hash functions PHOTON and Lesamnta-LW for consideration in FutureTPM which do not appear in the current report D2.2. This is so because we believe that our current list of recommendations, which includes 13 candidate hash functions (six of which are mandatory), is already large enough to provide sufficiently many choices in practice. Further, PHOTON and Lesamnta-LW —despite being standardized by ISO— do not seem to have gained much popularity in industry.

DRAFT

SHA-2. The SHA-2 family of hash functions was designed by the US National Security Agency, and subsequently published as a NIST standard in 2001 (last updated in 2015 [102]). Although no public competition was held to select this family of algorithms, public review and comments were accepted when the first draft of the standard [95] was made available. Currently, the best known attacks break the preimage resistance of SHA-256 when reduced to 52 rounds (out of 64) using 2^{255} operations, or of SHA-512 when reduced to 57 rounds (out of 80) using 2^{511} operations [73]. For collision resistance, a second-order differential attack breaks SHA-256 when reduced to 46 rounds (out of 64) in 2^{46} operations [23].

Among the members of the SHA-2 family, we recommend the following hash functions for implementation in FutureTPM:

- SHA-256 [except for collision resistance],
- SHA-384,
- SHA-512.

We note that whereas the three schemes satisfy, at least, 128-bit security for preimage resistance, only SHA-384 and SHA-512 satisfy this requirement for collision resistance. This means that in scenarios where critical levels of security are required, SHA-256 can only be considered when

[2] This result is disputed by some in the cryptographic community, and more optimistic, that is, higher, bounds are claimed [16]. However, our recommendations in this deliverable will be guided by the most conservative result from [27].

the relevant security properties are preimage and second-preimage resistance, but not collision resistance. However, widespread adoption of SHA-256 in numerous environments suggests that we should also include this algorithm among the set of recommended schemes, despite offering only 85.3 bits of collision resistance security. (Recall from above that some actually dispute this bound, and estimate it to be higher in practice.) Several reports expect SHA-256 to remain secure for at least another 10 years [105, 44].

SHA-3. The SHA-3 family [103], originally coined as Keccak, is the finalist of the NIST hash function competition started in 2006, which initially had 51 submissions. The SHA-3 family consists of four closely related fixed-length hash functions and two extendable output functions, and was adopted as a NIST standard in 2015 [103]. Unlike SHA-2, this family of functions is based on a permutation-based sponge construction [20].
SHA-3 has been subject to extensive cryptanalysis by the academic community (see the detailed overview in [19]), and to date the best known pre-image attack is for SHA3-512 when reduced to 8 (out of 24) rounds [91], while the best known collision-finding attack is for SHA3-384 and SHA3-512 when reduced to 3 rounds, and for SHA3-256 when reduced to 8 rounds [38]. These attacks are far from practical in terms of space and time. In fact, the absence of practical attacks, despite the huge cryptanalytic effort invested, motivated the SHA-3 inventors to also consider faster versions, i.e., Keccak using only 12 (out of 24) rounds [21, 131]. These faster algorithms, which are not part of the standardized family, are conjectured to be as secure as SHA-3 [21], and directly benefit from more than ten years of public scrutiny, including from the cryptanalysis conducted during and after the SHA-3 competition [21, 19].
We propose some members of the SHA-3 family for the use in the FutureTPM project. The rationale behind our choice is based on the same general considerations of the quantum-security of hash functions and their output lengths as discussed above in the SHA-2 context.

Among the members of the SHA-3 family, we recommend the following hash functions for implementation in FutureTPM:

- SHA3-256 [except for collision resistance],
- SHA3-384,
- SHA3-512.

We further consider extendable-output function (XOF) variants of SHA-3, named SHAKE, but only for optional implementation. For a hash function and XOF with an n-bit output built out of a sponge of capacity c bits, the security for generic quantum preimage and second-preimage attacks is $\geq \min(n/2, c/2)$ bits, and for generic quantum collision attacks is $\min(n/3, c/3)$ [31, 116], taking the most conservative bound in terms of hash function calls [27]. We thus recommend the following XOF for optional implementation in the FutureTPM project:

- SHAKE128 (n-bit output, 256-bit capacity) [with $n \geq 256$ for preimage and second-preimage resistance; not for collision resistance],
- SHAKE256 (n-bit output, 512-bit capacity) [with $n \geq 256$ for preimage and second-preimage resistance, and $n \geq 384$ for collision resistance].

DRAFT

BLAKE2b. This family of hash functions [6, 5, 111] is an improved version of BLAKE, one of the finalists of the NIST hash function competition. Its structure follows the HAIFA design [22]. BLAKE has been subject of significant cryptanalysis as part of the SHA-3 competition, as well as the changes introduced by BLAKE2. BLAKE2 has been rapidly adopted as an independent alternative to SHA-2 and SHA-3, and numerous benchmarks show its superiority in terms of speed when compared to the NIST standards [132, 121]. The best known attacks for BLAKE and BLAKE2 work against versions reduced to 2.75 rounds (out of ≥ 10) [45, 57], which are way far from posing a threat from a cryptographic point of view.

Among the members of the BLAKE2b family, we recommend the following hash functions for optional implementation in FutureTPM (with the reasoning as above).

- BLAKE2b-256 [except for collision resistance],
- BLAKE2b-384,
- BLAKE2b-512.

Other non-mandatory candidates. In addition to the above, we consider the following candidates suitable for implementation in FutureTPM. The selection is based mostly on meeting international standards (other than NIST). The implementation in FutureTPM is not mandatory, and a decision for implementation will depend on the requirements that arise in the development of the project.

- SM3 [133] [except for collision resistance]:
  This cryptographic hash function has a 256-bit output, and was approved as a Chinese National Standard by the Organization of State Commercial Administration of China (OSCCA) [118] in 2016. It may be the only hash function that can be used as foreign encryption technology allowed in China, because under the Chinese law, OSCCA requires that any company or individual selling encryption products in China to first obtain its approval.
- Ascon-XOF [4] (256-bit capacity, n-bit output, with $n \geq 256$ for preimage and second-preimage resistance; not for collision resistance). Ascon-XOF is based on the sponge construction and has the same capacity as SHAKE128. Therefore, the classical and post-quantum security is the same as for SHAKE128. The Ascon cipher suite is well analyzed and has been selected as the primary choice for lightweight authenticated encryption in the CAESAR competition and was submitted to the NIST lightweight cryptography competition.

DRAFT

	NIST [106]	ISO/IEC [66, 64]	ITU-T [70]	TCG [126]	IETF [43, 111, 129, 114]	ETSI [52, 51]	W3C [137, 136]	SOG-IS [113]	NSA-CSS [108]	ECRYPT-CSA [44]	PQCRYPTO [109]
SHA-256	✓	✓	✓	✓	✓	✓	✓	✓		✓	
SHA-384	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SHA-512	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
SHA3-256	✓	✓			D	✓		✓		✓	
SHA3-384	✓	✓			D	✓		✓		✓	✓
SHA3-512	✓	✓			D	✓		✓		✓	✓
SHAKE128*	✓	✓			D	✓		✓		✓	
SHAKE256*	✓	✓			D	✓		✓		✓	
BLAKE2b-256*					✓					✓	
BLAKE2b-384*					✓					✓	
BLAKE2b-512*					✓					✓	
SM3*		✓		✓	D	✓					
Ascon-XOF*											

Table 1: Recommendation of selected and optional* hash functions by standardization bodies, and by academic projects. D: draft in progress.

DRAFT

	AMCL [89]	Botan [80]	Bouncy Castle [77]	cryptlib [36]	Crypto++ [122]	GnuTLS [124]	IAIK-JCE [59]	Libgcrypt [123]	Libsodium [35]	mbed TLS [81]	NaCl [18]	Nettle [90]	NSS [92]	OpenSSL [125]	SJCL [120]	UniCrypt [15]	wolfCrypt [135]
SHA-256	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SHA-384	✓	✓	✓		✓	✓	✓	✓		✓		✓	✓	✓		✓	✓
SHA-512	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SHA3-256	✓	✓	✓	✓	✓	✓	✓	✓				✓		✓			✓
SHA3-384	✓	✓	✓	✓	✓	✓	✓	✓				✓		✓			✓
SHA3-512	✓	✓	✓	✓	✓	✓	✓	✓				✓		✓			✓
SHAKE128*	✓	✓	✓		✓		✓	✓				✓		✓			
SHAKE256*	✓	✓	✓		✓		✓	✓				✓		✓			
BLAKE2b-256*		✓	✓		✓												✓
BLAKE2b-384*		✓	✓		✓												✓
BLAKE2b-512*		✓	✓		✓				✓					✓			✓
SM3*		✓	✓		✓									✓			
Ascon-XOF*																	

Table 2: Support of selected and optional* hash functions by cryptographic libraries.

3 Block Ciphers

With respect to the security of block ciphers, Deliverable D2.1 [30] identified that in the QS1 and QS2 scenarios, Grover's algorithm can be used to achieve a quadratic speed-up in the brute force key search, effectively halving the key bit security when compared to the non-quantum setting QS0. Further, a quantum adversary in QS2 can execute a preimage attack which depends on the block size of the algorithm. In particular, in order to achieve a target 128-bit security level, we require key sizes of at least 256 bits both in QS1 and QS2, plus a block size of at least 256 bits in QS2. As in practice the QS2 attacks on block ciphers seem to be challenging to mount, our recommendations focus mostly on the QS1 setting. The comment made in Section 2 about implementing primitives proposed in the TCG Algorithm Registry [126] to maintain compatibility with current TPM implementations applies, analogously, to the current section as well.
Note that Deliverable D2.1 [30] lists the block ciphers Serpent and Twofish for consideration in FutureTPM, while they do not appear in the current report D2.2. This is so because we believe that they do not offer advantages over the remaining AES and Camellia, and only the latter two gained much popularity in industry.

AES/Rijndael. The AES block cipher is the winner of the 1997 NIST AES competition, and was announced as a US standard in 2001 [93]. Since then it has undergone a large amount of review and analysis by numerous experts. We note that the original submission, coined Rijndael [33], was designed to support any combination of key and block lengths among 128, 192 and 256 bits. However, despite preserving the three different key sizes, the AES standard is defined only for a fixed block length of 128 bits, with the rationale that in 2001 this block size seemed sufficient. We refer the reader for instance to [30, 44] for surveys on the best currently known attacks on AES and Rijndael.

Among the members of the AES/Rijndael family, we recommend the following block ciphers for implementation in FutureTPM:

- AES-256 [with 128-bit block size, thus only for QS1].

For optional implementation we further recommend:

- Rijndael-256 [192-bit block size for QS1, and 256-bit block size for QS1 and QS2].

Camellia. The Camellia block cipher [86] was jointly developed by Mitsubishi Electric and NTT of Japan. It has been approved for use by the ISO/IEC [61]. The cipher is claimed to have security levels and processing abilities comparable to the NIST AES. It has key sizes of 128, 192 and 256 bits, and a block size of 128 bits. Therefore, the same considerations as made for the case of AES apply for this block cipher.

Among the members of the Camellia family, we recommend the following block ciphers for optional implementation:

- Camellia-256 [with 128-bit block size, thus only for QS1].

Complementing our note on the SM3 hash function in Section 2, we conclude with a remark on the SM4 block cipher [128] included in the current TCG Algorithm Registry [126]. SM4 is an algorithm approved by the Chinese standardization organization (OSCCA) [119], and may be the only block cipher that currently can be allowed as encryption technology in China. However, as SM4 has only 128-bit key and block sizes, it will not satisfy the QS1 (leave alone QS2) security requirements. Based on our analyses we thus cannot recommend it for use in FutureTPM.

DRAFT

	NIST [93]	ISO/IEC [61]	ITU-T [68]	TCG [126]	IETF [55, 86, 112, 7, 115, 87, 110]	ETSI [47, 48, 49, 53]	W3C [137, 136]	SOG-IS [113]	NSA-CSS [108]	ECRYPT-CSA [44]	PQCRYPTO [109]
AES	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Rijndael*											
Camellia*		✓	✓	✓	✓	✓				✓	

Table 3: Recommendation of selected and optional* block ciphers by standardization bodies, and by academic projects. The entry “Rijndael” refers to the 192- and 256-bit block versions.

DRAFT

	AMCL [89]	Botan [80]	Bouncy Castle [77]	cryptlib [36]	Crypto++ [122]	GnuTLS [124]	IAIK-JCE [59]	Libgcrypt [123]	Libsodium [35]	mbed TLS [81]	NaCl [18]	Nettle [90]	NSS [92]	OpenSSL [125]	SJCL [120]	UniCrypt [15]	wolfCrypt [135]
AES	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Rijndael*			✓				✓	✓									
Camellia*		✓	✓		✓	✓	✓	✓		✓		✓		✓			✓

Table 4: Support of selected and optional* block ciphers by cryptographic libraries. The entry “Rijndael” refers to the 192- and 256-bit block versions.

4 Modes of Operation

Hash functions and block ciphers are typically operated in *modes* (of operation) that provide specific services to applications. Services that we found relevant in the context of TPMs are message authentication, encryption, and authenticated encryption (AE). In this section we suggest modes that securely provide these services for the use in the FutureTPM project.

The TCG Algorithm Registry [126] currently does not mandate the implementation of any integrated AE scheme. (However, it allows the implicit emulation of such a mode via the Encrypt-then-MAC paradigm.) As research in the past two decades has shown that the use of non-authenticated encryption often leads to vulnerable overall constructions, while the use of AE generally promises a fair level of robustness, we decided to explicitly include integrated AE modes in our set of recommendations.

The modes we propose below either require a Merkle–Darmgard hash function, a sponge-based hash function, or a block cipher as an underlying primitive. The formal security analyses of the modes rely on a cryptographic property of the primitive in combination with an information-theoretic argument. While the cryptographic property can be affected by quantum attackers (e.g. by Grover's algorithm), the information-theoretic argument cannot (information-theoretic arguments assume computationally unbounded attackers in the first place). As a consequence, all the modes of operation that we propose for block ciphers are deemed secure for use in FutureTPM if the block cipher is instantiated with one from the set recommended in Section 3. Similarly, all the hash functions recommended in Section 2 can be securely used with the modes for hash functions proposed here (if they are of the right type).

DRAFT

4.1 Message Authentication

In symmetric cryptography, message authentication is typically implemented with a message authentication code (MAC). A MAC is a keyed function that takes a message and outputs a tag. The idea is that the sender of a message computes the tag and appends it to the message before sending it, and the receiver verifies the authenticity of the message by re-computing the tag and comparing.

For the use in FutureTPM we recommend the MAC instantiations HMAC, KMAC, and CMAC. The three schemes require different building blocks: HMAC uses a Merkle–Damgard hash function, KMAC uses a sponge-based hash function, CMAC uses a block cipher. While HMAC and CMAC appear in the TCG Algorithm Registry [126], KMAC was only recently introduced and is not included in the TCG's algorithm portfolio. Thus, our recommendation to make HMAC mandatory, and to leave KMAC and CMAC optional for implementations, is aligned with [126]. We note that all three schemes are provably secure, not only as MAC schemes but also as pseudorandom functions (PRF).

HMAC. The HMAC algorithm uses a Merkle–Damgard (MD) hash function as a building block, which itself is constructed by iteratively evaluating an internal compression function. The security argument of HMAC requires less strong properties than the standard goals of hash functions (e.g. collision resistance), and indeed HMAC was proven to be a secure PRF under the sole assumption that the internal compression function is a PRF [11, 12]. That is, even when instantiated with a hash function with compromised collision resistance (e.g. MD5, SHA1), the HMAC algorithm is considered resistant to forgery attacks. The best known specifically quantum attack on HMAC is an exhaustive key search using Grover's algorithm, where the key space coincides with the

output space of the compression function. Among our main recommendations in Section 2 only the members of the SHA-2 family are MD constructions (all of which exhibit a large enough output of the compression function). We therefore recommend the HMAC scheme for use in FutureTPM for any of SHA-256, SHA-384, and SHA-512.

KMAC. The KMAC scheme was proposed in NIST SP-800-185 [104] as a MAC derived from the Keccak permutation of SHA3 (more precisely: from the closely related SHAKE128 and SHAKE256 XOFs). We note that the KMAC principle can also be applied to the random permutation of any other sponge-based hash function, including that of Ascon. The following references to KMAC shall be understood in this general sense. The best known specifically quantum attack on KMAC is an exhaustive key search using Grover's algorithm. Among our main recommendations in Section 2 only the members of the SHA-3 family and the related SHAKE128 and SHAKE256 XOFs are sponge based. We recommend the KMAC scheme for use in FutureTPM in conjunction with any of these primitives.

CMAC. The CMAC algorithm is a mode of operation of a block cipher. It is a version of CBC-MAC (which would CBC-encrypt the message with fixed IV and output the last ciphertext block as the tag), but is strengthened against truncation and length extension attacks by processing the last block in a special, key-dependent way. The best quantum attacks on CMAC are therefore quantum attacks on the underlying block cipher (both in QS1 and QS2). Thus, as an optional recommendation for use in FutureTPM, we consider CMAC instantiated with any block cipher proposed in Section 3.

DRAFT

4.2 Symmetric Encryption

Messages are encrypted to preserve their confidentiality when sent over an insecure channel. It is important to note that encryption itself does not ensure any type of authentication. In particular, messages that are received by a party, even if the transmission was encrypted, cannot be assumed to originate from the notional sender. It is further a standard result that pure confidentiality schemes offer very limited security against active attackers that can modify ciphertexts in transmission. The modes recommended in this subsection should thus be used with extreme caution. However, as ways to operate them securely are known (e.g., in the Encrypt-then-MAC construction), we consider such modes in our recommendations nevertheless.

CFB, CBC, CTR. The modes of operation of a block cipher CFB, CBC, CTR, and OFB are classic pure encryption modes. All four appear in the TCG Algorithm Registry [126], but only CFB is marked as mandatory. As the best quantum attacks against these modes are attacks against the underlying block ciphers, in principle all four modes can be securely used in the FutureTPM context if they are instantiated with a block cipher recommended in Section 3. However, as OFB is rather of historic relevance and little used in current practice, we drop it from the list. Overall, in alignment with [126], we recommend CFB for mandatory implementation, and CBC and CTR for optional implementation.

XOR obfuscation. The TPM 2.0 specification [127] has the concept of session-based parameter XOR obfuscation with mask. XOR obfuscation resembles CTR encryption (see above), but it uses a KDF as the pseudorandom function instead of a block cipher. The particular KDF used is

	NIST [99, 104, 97]	ISO/IEC [63, 62]	ITU-T [67, 69]	TCG [126]	IETF [75, 43, 117]	ETSI [54, 50]	W3C [137, 136]	SOG-IS [113]	NSA-CSS [108, 71]	ECRYPT-CSA [44]	PQCRYPTO [109]
HMAC	✓	✓	✓	✓	✓	✓	✓	✓	D	✓	
KMAC*	✓										
CMAC*	✓	✓	✓	✓	✓	✓		✓		✓	

Table 5: Recommendation of selected and optional* authentication modes of operation by standardization bodies, and by academic projects. D: draft in progress.

DRAFT

	AMCL [89]	Botan [80]	Bouncy Castle [77, 78]	cryptlib [36]	Crypto++ [122]	GnuTLS [124]	IAIK-JCE [59]	Libgcrypt [123]	Libsodium [35]	mbed TLS [81]	NaCl [18]	Nettle [90]	NSS [92]	OpenSSL [125]	SJCL [120]	UniCrypt [15]	wolfCrypt [135]
HMAC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
KMAC*			U											✓			
CMAC*		✓	✓		✓		✓	✓		✓		✓	✓	✓			

Table 6: Support of selected and optional* authentication modes of operations by cryptographic libraries. U: likely addition in upcoming releases.

coined as KDFa in TCG notation [127], and it corresponds to the key derivation function (KDF) defined in NIST SP800-108 [100] in conjunction with HMAC. We recommend this mode as optional, assuming it is used with a secure HMAC instance (see Section 4.1).

4.3 Authenticated Encryption

Authenticated encryption (AE) primitives provide integrity and confidentiality simultaneously. In principle, the same goals can also be achieved by first encrypting a message using any method proposed in Section 4.2, and then authenticating the ciphertext using any method proposed in Section 4.1. However, combining the two primitives into one proved to be more robust and, more importantly, demonstrated higher degrees of efficiency. In particular, integrated AE modes get along with a single key instead of two, and they often can achieve the two security goals using the same shared internal building blocks.
The TCG Algorithm Registry [126] currently does not list any AE mode. As AE schemes are in many cases superior to separate encryption and MAC schemes, we nevertheless propose the implementation of AE modes in FutureTPM (but not as mandatory components). We discuss four candidate schemes below.

CCM. The CCM mode ('Counter with CBC-MAC') [96, 134] is a mode of operation of any 128-bit block cipher. A formal security analysis is conducted in [72], and an implication of the latter is that CCM is a secure AE mode when instantiated with any of the 128-bit block ciphers approved in Section 3. CCM is standardized by NIST [96], and is used in IEEE 802.11i WiFi, in a low-energy profile protocol of Bluetooth, and is available as an encryption engine in both TLS and IPsec. We note that drawbacks of the CCM mode are that it requires two (rather than one) block cipher invocations per message block, and that encryption operations cannot be performed online.

GCM. The GCM mode ('Galois/Counter Mode') [98] is a mode of operation of any 128-bit block cipher. A formal security analysis is conducted in [88], and an implication of the latter is that GCM is a secure AE mode when instantiated with any of the 128-bit block ciphers approved in Section 3. GCM is standardized by NIST [98] and in ISO/IEC 19772:2009, and is used in IEEE 802.11 WiFi, IEEE 1619 based storage, and is available as an encryption engine in TLS, SSH, and IPsec. Attractive properties of the GCM mode are that it is parallelizable, and that it was designed and published without any patent restrictions.

EAX. The EAX mode ('Encrypt-then-Authenticate-then-Translate') is a mode of operation of any block cipher. A formal security analysis is conducted in [13], and an implication of the latter is that EAX is a secure AE mode when instantiated with any of the block ciphers approved in Section 3. Note that, in contrast to CCM and GCM, the EAX mode also works with ciphers with block length 256, e.g. Rijndael. That is, EAX can be secure in the QS2 setting, while CCM and GCM do not reach beyond QS1. While EAX is not as widely standardized as CCM and GCM, a close relative to EAX, dubbed EAX', is used in ANSI C12.22 in the context of grid security.

Ascon. Ascon [4] is a cipher suite providing authenticated encryption with associated data (AEAD) and hashing functionality. Ascon has been selected as the primary choice for lightweight

DRAFT

authenticated encryption in the final portfolio of the CAESAR competition[3] and has been submitted to the NIST lightweight cryptography competition[4].
The main strength of Ascon comes from its robust mode of operation for authenticated encryption. The mode is based on a duplex structure like MonkeyDuplex [32] (which is a standard mode of operation in sponge-based AE constructions) but uses a stronger keyed initialization and keyed finalization. This provides additional robustness against key recovery attacks. Similar to block cipher modes, the Ascon mode can be instantiated with a larger permutation and key size to offer the required post-quantum security level. However, no cipher has been proposed with this target in mind yet.
The Ascon cipher suite itself has been designed for lightweight use cases and provides the family member Ascon-80pq with 80-bit post-quantum security (against Grover key search). Ascon-80pq uses a 320-bit permutation, a capacity of 256 bits and a key size of 160 bits.

DRAFT

[3] https://competitions.cr.yp.to/caesar.html
[4] https://csrc.nist.gov/projects/lightweight-cryptography

	NIST [94, 101, 96, 98]	ISO/IEC [65, 60]	ITU-T [68]	TCG [126]	IETF [55, 134, 10, 112, 7, 115, 87, 110]	ETSI [47, 48, 49, 53]	W3C [137, 136]	SOG-IS [113]	NSA-CSS [108]	ECRYPT-CSA [44]	PQCRYPTO [109]
CFB	✓	✓		✓				✓		L	
CBC	✓	✓	✓	✓	✓	✓	✓	✓	✓	L	
CTR	✓	✓		✓	✓	✓	✓	✓		L	
CCM*	✓	✓			✓	✓		✓	✓	L	
GCM*	✓	✓			✓	✓	✓	✓		✓	✓
EAX*		✓						✓		✓	
Ascon*											

Table 7: Recommendation of selected and optional* encryption modes of operation by standardization bodies, and by academic projects. L: legacy usage only.

DRAFT

	AMCL [89]	Botan [80]	Bouncy Castle [77]	cryptlib [36]	Crypto++ [122]	GnuTLS [124]	IAIK-JCE [59]	Libgcrypt [123]	Libsodium [35]	mbed TLS [81]	NaCl [18]	Nettle [90]	NSS [92]	OpenSSL [125]	SJCL [120]	UniCrypt [15]	wolfCrypt [135]
CFB	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓		✓			
CBC	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓
CTR	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓	✓		✓
CCM*		✓	✓		✓	✓	✓	✓				✓		✓	✓		✓
GCM*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓
EAX*		✓	✓		✓							✓					
Ascon*																	

Table 8: Support of selected and optional* encryption modes of operation by cryptographic libraries.

5 Public-Key Encryption

We propose a selection of public key encryption methods (PKE and KEMs) that we assess to be secure against quantum attackers. Our choice is crucially influenced by the (intermediate) results of the ongoing NIST PQC competition [107], so that our project can benefit from academic cryptanalysis efforts and optimized implementation techniques. Our selection offers multiple performance-security tradeoffs and diversity in the sense that not all schemes rely on the same hardness assumption.

5.1 Lattice-Based

New Hope. The New Hope cryptosystem [3], based on Ring-LWE [84], can be seen as the "purest" (but optimized) instantiation of the encryption scheme / KEM described in [84] using the ring $\mathbb{Z}_q[X]/(X^n+1)$ with $n = 512$ and $q = 12289$. This choice of parameters (together with an appropriate error distribution) results in over 100 bits of security against quantum attacks under a rather conservative security analysis [3]. While this is below the usually-desired 128-bits, due to the rather conservative analysis and the fact that hardware TPMs are resource-constrained, we believe that NewHope with these parameters provides a good security / efficiency trade-off.
Of equal importance, the ring $\mathbb{Z}_q[X]/(X^n+1)$ with $n = 512$ and $q = 12289$ can also be reused for the digital signature BLISS [40] (described in the next section), which has small signatures and a similar security of around 100-bits. Using the same ring allows us to reuse the same operations for both primitives, which is a big advantage for devices that cannot devote too many resources (e.g. code size) to cryptographic operations.

CRYSTALS-Kyber. The Kyber [24] encryption scheme / KEM is part of the CRYSTALS (Cryptographic Suite for Algebraic Lattices) submitted to the NIST standardization process. The security of Kyber is based on the hardness of Module-LWE (or generalized-LWE) [26, 76], which is at least as hard as Ring-LWE. In addition to the (possibly) stronger security guarantees, an advantage of Kyber is that it easily allows one to vary security without re-implementing the main components of the encryption scheme. In particular, the only operations that have to be implemented for all security levels, in addition to SHAKE, are polynomial operations in the ring $\mathbb{Z}_q[X]/(X^n+1)$ for $q = 3329$ and $n = 256$. Then one can vary the underlying dimension of the generalized problem in intervals of 256.
All lattice-based schemes in the NIST process that achieve around 128-bits of quantum security have the dimensions of their (Generalized)-LWE instance between 700 and 800. Because the dimension of a Kyber LWE instance can be any multiple of 256, one can set the dimension to be 768 while still being able to use the very efficient operations over the ring $\mathbb{Z}_q[X]/(X^n+1)$ with n a power-of-2. Furthermore, Kyber is very efficient on many (even low-power) devices due to its extensive use of NTT, which allows all computation to be fast and in-place. The result is that Kyber (and New Hope with $n = 512$) have the smallest memory consumption on low-powered Cortex M0 chips of all lattice-based schemes [25]. This makes Kyber a very good choice for all platforms.

NTTRU. The NTTRU encryption scheme [85] is an instantiation of the classic NTRU encryption scheme [58] over the cyclotomic ring $\mathbb{Z}_q[X]/(X^{768}-X^{384}+1)$ with $q = 7681$ chosen such that the polynomial $X^{768}-X^{384}+1$ splits into factors of degree 3. The particular split of this polynomial allows for an NTT-based implementation of the main algebraic operations of the scheme to be as efficient as the one in Kyber and New Hope. The performance advantage of NTTRU

DRAFT

over Kyber and NewHope comes from the fact that there is no public part that is derived via a cryptographic PRF from a seed during key generation, encryption, or decryption. Because this latter operation takes up a significant (more than half) amount of time in Kyber and NewHope, removing it allows for a very noticeable saving. NTTRU is currently the fastest quantum-resistant public key encryption candidate and may therefore be useful in environments where speed is of utmost importance.

5.2 Code-Based

BIKE. BIKE [46] is a code-based Key Encapsulation Mechanism (KEM), including three similar constructions, denoted by BIKE-1, BIKE-2 and BIKE-3. All the three variants use a bit-flipping decoder for a Quasi-Cyclic Moderate-Density-Parity-Check (QC-MDPC) code in their decapsulation algorithms. Each variant offers different performance trade-offs.
Based on the NIST comments, BIKE offers key and ciphertext sizes and performance that are competitive with ring and module lattice schemes (especially at the lower security categories). The BIKE variants are structurally quite similar to well-studied lattice cryptosystems. BIKE 1 and 2 are similar to NTRU, and BIKE 3 is similar to RLWE cryptosystems, substituting shortness in the Hamming metric for shortness in the Euclidean metric. The security of the BIKE suite was proven based on decisional variants with parity of the Quasi-Cyclic Syndrome Decoding (QCSD) and Quasi-Cyclic Codeword Finding (QCCF) problems. Security strengths of BIKE are based on information-set-decoding attacks, which have a long history of analysis during which the complexity of such attacks has not greatly changed.
In the NIST first round submission, BIKE KEM variants only provided IND-CPA security. But in the second round submission, each variant also has an IND-CCA construction.

DRAFT

6 Signature Schemes

We propose a selection of signature schemes that we assess to be secure against quantum attackers. As in Section 5 our choice is crucially influenced by the (intermediate) results of the ongoing NIST PQC competition [107]. Again, our selection offers multiple performance-security tradeoffs and diversity in the sense that not all schemes rely on the same hardness assumption.

We note that Deliverable D2.1 [30] of this project did not yet consider the signature schemes BLISS and Rainbow. We introduce them to the candidate list as a result of closely monitoring the NIST PQC evaluation process, and because we believe that they are particulary suitable in the FutureTPM context due to their attractive profile in hardware implementations.

6.1 Lattice-Based

BLISS. BLISS [40] is a digital signature scheme based on the hardness of finding short vectors in the NTRU lattice. As many other lattice-based signature schemes, the construction is based on the "Fiat-Shamir with Aborts" framework [82] where, to keep the size of the signature low, the signer performs rejection sampling on the potential signature to create a distribution independent of the secret key. The unique feature of this signature is that it crucially uses the structure of the NTRU problem by sampling a distribution from a bimodal Gaussian in order to create a Gaussian distribution using rejection sampling.
The BLISS signature scheme was not submitted to the NIST process because of its crucial use of Gaussian sampling, which is notoriously hard to do in constant-time. For the NIST process, it was therefore decided to instead use the slightly longer signatures (like CRYSTALS-Dilithium described below) which only use uniform sampling. Nevertheless, BLISS was used as an option in the open-source StrongSwan VPN, and there has been recent work showing that a careful implementation can be constant time [9]. As previously mentioned, a big advantage of BLISS in this project is that for somewhat lower security levels (appropriate for IoT), the BLISS signature scheme and the New Hope encryption scheme can work over the same polynomial ring $\mathbb{Z}_q[X]/(X^n+1)$ and reuse many of the same operations. If one wanted higher security levels of BLISS, the value of q would need to increase, whereas the value of q is fixed for all security levels of New Hope.

CRYSTALS-Dilithium. Dilithium [41] is a signature scheme that's part of the CRYSTALS suite submitted to the NIST standardization process. Like the encryption scheme Kyber, it is based on the hardness of Generalized (Module)-LWE, and additionally the Module-SIS problem. Its design stems from the "Fiat-Shamir with Aborts" approach and for simplicity of universal constant-time implementation, it only uses sampling from the uniform distribution. Like Kyber, Dilithium crucially uses NTT for its ring operations which also results in a very small operational footprint and makes it quite efficient across many platforms. Also like Kyber, it is easy to vary security of the scheme by simply extending the number of elements in $\mathbb{Z}_q[X]/(X^n+1)$ (with $n=256$ and $q\approx 2^{23}$) that make up the Module-LWE/SIS instance.

6.2 Multivariate-Based

Rainbow. Rainbow [37] is a multivariate digital signature scheme. It is a generalization of the Unbalanced Oil and Vinegar (UOV) structure. This design allows parameterizations that are

DRAFT

more efficient at the cost of additional algebraic structure. The Rainbow signature scheme was analyzed to be EUF-CMA secure utilizing a hash construction with a random salt.
Since the original Rainbow signature scheme was published in 2005, the scheme has been studied with various parameters. NIST commented that the spectrum of Rainbow parameters allows for optimization in a diverse array of use cases. In the NIST second round submission, the key generation algorithm for the original Rainbow scheme was improved, the nine parameter sets in the first round submission was narrowed down to three sets. Two variants of Rainbow signatures were proposed in order to make a trade-off in key size and performance.
It is also commented by NIST that a further benefit of Rainbow is that it has also been studied in other contexts, including in lightweight applications.

6.3 Hash-Based

SPHINCS+. SPHINCS+ [17] is a stateless variation of the stateful XMSS hash-based signature scheme [28]. The distinguishing characteristic of this signature scheme is that its security is based only on the hardness of symmetric primitives. The signature size is around 30KB, which is acceptable in many scenarios. The main downside of the scheme is that it is slow and resource-intensive (c.f. [25]). In applications where speed and, to a lesser extent, signature size are not very important and there is a general consensus within the cryptographic community that it will be standardized by NIST.

DRAFT

7 Conclusion

This report evaluates cryptographic primitives of various kinds, ranging from primitives (block ciphers, hash functions) over modes of operation (encryption, authentication, authenticated encryption) to public key methods (public key encryption, signatures). In each category we identify candidates suitable for implementation in the FutureTPM project, taking into account many aspects of practical relevance and, most importantly, their potential to be secure against powerful quantum adversaries.
For reference, in Tables 9–12 we reproduce our choice and decisions in compact form.

Primitive	Type	Algorithm	QS0	QS1	QS2	Notes
Hash	*Mandatory:*	SHA-256	✓	✓	✓	Except for col. resist.
		SHA-384	✓	✓	✓	
		SHA-512	✓	✓	✓	
		SHA3-256	✓	✓	✓	Except for col. resist.
		SHA3-384	✓	✓	✓	
		SHA3-512	✓	✓	✓	
	Optional:	SHAKE128	✓	✓	✓	Except for col. resist.
		SHAKE256	✓	✓	✓	
		BLAKE2b-256	✓	✓	✓	Except for col. resist.
		BLAKE2b-384	✓	✓	✓	
		BLAKE2b-512	✓	✓	✓	
		SM3	✓	✓	✓	Except for col. resist.
		Ascon-XOF	✓	✓	✓	Except for col. resist.
Block cipher	*Mandatory:*	AES-256	✓	✓		
	Optional:	Rijndael-256	✓	✓		192-bit block version
		Rijndael-256	✓	✓	✓	256-bit block version
		Camellia-256	✓	✓		

Table 9: Summary of recommendations of symmetric cryptographic primitives

DRAFT

Mode	Type	Algorithm	Notes
MAC	*Mandatory:*	HMAC	only w/ SHA-256, SHA-384, SHA-512
	Optional:	KMAC	only w/ SHA3-*, SHAKE128, SHAKE256
		CMAC	only w/ block cipher from Table 9
Symmetric encryption	*Mandatory:*	CFB	only w/ block cipher from Table 9
	Optional:	CBC	only w/ block cipher from Table 9
		CTR	only w/ block cipher from Table 9
Authenticated encryption	*Optional:*	CCM	only w/ block cipher from Table 9
		GCM	only w/ block cipher from Table 9
		EAX	only w/ block cipher from Table 9
		Ascon	

Table 10: Summary of recommendations of symmetric modes of operation

Class	Scheme	Assumption	Notes
Lattice based	New Hope	RingLWE	works well with BLISS signatures
	CRYSTALS-Kyber	ModuleLWE	
	NTTRU		optimized NTRU
Code based	BIKE		

Table 11: Summary of recommendations of public-key encryption methods

Class	Scheme	Assumption	Notes
Lattice based	BLISS	RingLWE	works well with NewHope encryption
	CRYSTALS-Dilithium	ModuleLWE/SIS	
Multivariate based	Rainbow		optimized UOV
Hash based	SPHINCS+		minimal assumptions

Table 12: Summary of recommendations of signature schemes

DRAFT

8 List of Abbreviations

Abbreviation	Translation
AE	Authenticated Encryption
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
BIKE	Bit Flipping Key Encapsulation
BLISS	Bimodal Lattice Signature Scheme
CBC	Cipher Block Chaining
CCM	Counter with CBC-MAC
CFB	Ciphertext Feedback
CMAC	Cipher-based MAC
CRYSTALS	Cryptographic Suite for Algebraic Lattices
CTR	Counter
DAA	Direct Anonymous Attestation
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
EAX	Encrypt-then-Authenticate-then-Translate
EC	European Commission
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve DSA
ECRYPT-CSA	European Network of Excellence in Cryptology - Coordination & Support Action
ETSI	European Telecommunications Standards Institute
GCM	Galois/Counter Mode
HMAC	Hash-based MAC
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IND-CPA	Indistinguishability under Chosen-Plaintext Attack
ISO/IEC	International Organization for Standardization/International Electrotechnical Commission
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
IV	Initialization Vector
KDF	Key Derivation Function
KEM	Key Encapsulation Mechanism
KMAC	Keccak MAC
LWE	Learning With Errors

DRAFT

Abbreviation	Translation
MAC	Message Authentication Code
MD	Merkle–Damgard
MLWE	Module LWE
NSA/CSS	National Security Agency/Central Security Service
NIST	National Institute of Standards and Technology
NTT	Number-Theoretic Transform
PKE	Public Key Encryption
PQC	Post-Quantum Cryptography
PRF	Pseudorandom Function
RLWE	Ring LWE
OFB	Output Feedback
QR	Quantum-Resistant
QROM	Quantum Random Oracle Model
ROM	Random Oracle Model
SHA	Secure Hash Algorithm
SOG-IS	Senior Officials Group Information Systems Security
TBD	To Be Determined
TCG	Trusted Computing Group
TDES	Triple-DES
TPM	Trusted Plaform Module
XOF	Extendable-Output Function
XOR	Exclusive OR

DRAFT

References

[1] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, pages 553–572, 2010.

[2] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooperand Quynh Dang, Yi-Kai, Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, and Daniel Smith-Tone. STATUS REPORT ON THE FIRST ROUND OF THE NIST. Technical report, National Institute of Standards and Technology, 2017. available at https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8240.pdf.

[3] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 327–343, 2016.

[4] ASCON Project. ASCON: Lightweight authenticated encryption & hashing. https://ascon.iaik.tugraz.at/.

[5] Jean-Philippe Aumasson, Samuel Neves, Zooko Wilcox-O'Hearn, and Christian Winnerlein. BLAKE2 - fast secure hashing. https://blake2.net/.

[6] Jean-Philippe Aumasson, Samuel Neves, Zooko Wilcox-O'Hearn, and Christian Winnerlein. BLAKE2: simpler, smaller, fast as MD5, January 2013. https://blake2.net/blake2.pdf.

[7] M. Badra. Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode. RFC 5487, March 2009. https://rfc-editor.org/rfc/rfc5487.txt.

[8] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, pages 719–737, 2012.

[9] Gilles Barthe, Sonia Belaïd, Thomas Espitau, Pierre-Alain Fouque, Mélissa Rossi, and Mehdi Tibouchi. GALACTICS: gaussian sampling for lattice-based constant-time implementation of cryptographic signatures, revisited. *IACR Cryptology ePrint Archive*, 2019:511, 2019.

[10] M. Bellare, T. Kohno, and C. Namprempre. The Secure Shell (SSH) Transport Layer Encryption Modes). RFC 4344, January 2006. https://rfc-editor.org/rfc/rfc4344.txt.

[11] Mihir Bellare. New proofs for NMAC and HMAC: Security without collision-resistance. In *Advances in Cryptology (CRYPTO)*, volume 4117 of *LNCS*, pages 602–619, Santa Barbara, CA, August 2006. Springer.

[12] Mihir Bellare. New proofs for NMAC and HMAC: Security without collision-resistance. Cryptology ePrint Archive: Report 2006/043, 2014. Revised version https://eprint.iacr.org/2006/043.

DRAFT

[13] Mihir Bellare, Phillip Rogaway, and David A. Wagner. The EAX mode of operation. In *FSE*, volume 3017 of *Lecture Notes in Computer Science*, pages 389–407. Springer, 2004.

[14] Fabrice Benhamouda, Jan Camenisch, Stephan Krenn, Vadim Lyubashevsky, and Gregory Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, pages 551–572, 2014.

[15] Bern University of Applied Sciences (BFH) - E-Voting Group (EVG). UniCrypt (v. 2.3), September 2017. https://github.com/bfh-evg/unicrypt.

[16] Daniel J. Bernstein. Cost analysis of hash collisions : will quantum computers make SHARCS obsolete? In *Workshop on Special-purpose Hardware for Attacking Cryptograhic Systems (SHARCS)*, pages 105–116, Lausanne, Switserland, September 2009.

[17] Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. SPHINCS: practical stateless hash-based signatures. In *EUROCRYPT (1)*, volume 9056 of *Lecture Notes in Computer Science*, pages 368–397. Springer, 2015.

[18] Daniel J. Bernstein, Tanja Lange, and Peter Schwabe. NaCl (v. 20110221), February 2011. https://nacl.cr.yp.to/.

[19] Guido Bertoni, Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Team Keccak - Third party cryptanalysis. https://keccak.team/third_party.html.

[20] Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. Cryptographic sponge functions, January 2011. https://keccak.team/files/CSF-0.1.pdf.

[21] Guido Bertoni, Joan Daemen, Michaël Peeters, Gilles Van Assche, Ronny Van Keer, and Benoît Viguier. Kangarootwelve: fast hashing based on keccak-p. Cryptology ePrint Archive, Report 2016/770, 2016. https://eprint.iacr.org/2016/770.

[22] Eli Biham and Orr Dunkelman. A framework for iterative hash functions - HAIFA. Cryptology ePrint Archive: Report 2007/278, 2007. https://eprint.iacr.org/2007/278.

[23] Alex Biryukov, Mario Lamberger, Florian Mendel, and Ivica Nikolić. Second-order differential collisions for reduced SHA-256. In *Advances in Cryptology (ASIACRYPT)*, volume 7073 of *LNCS*, pages 270–287, Seoul, South Korea, December 2011. Springer.

[24] Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018*, pages 353–367, 2018.

[25] Leon Botros, Matthias J. Kannwischer, and Peter Schwabe. Memory-efficient high-speed implementation of Kyber on Cortex-M4. In *AFRICACRYPT*, volume 11627 of *Lecture Notes in Computer Science*, pages 209–228. Springer, 2019.

DRAFT

[26] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 97–106, 2011.

[27] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. In *Latin American Symposium on Theoretical Informatics (LATIN)*, volume 1380 of *LNCS*, pages 163–169, Campinas, Brazil, April 1998. Springer.

[28] Johannes A. Buchmann, Erik Dahmen, and Andreas Hülsing. XMSS - A practical forward secure signature scheme based on minimal security assumptions. In *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings*, pages 117–129, 2011.

[29] Liqun Chen, Nada El Kassem, Anja Lehmann, and Vadim Lyubashevsky. A framework for efficient lattice-based DAA. In *CYSARM@CCS*, pages 23–34. ACM, 2019.

[30] The FutureTPM Consortium. First report on new qr cryptographic primitives. Deliverable D2.1, September 2018.

[31] Jan Czajkowski, Leon Groot Bruinderink, Andreas Hülsing, Christian Schaffner, and Dominique Unruh. Post-quantum security of the Sponge construction. In *International Conference on Post-Quantum Cryptography (PQCrypto)*, volume 10786 of *LNCS*, pages 185–204, Fort Lauderdale, FL, USA, April 2018. Springer.

[32] Joan Daemen. Permutation-based encryption, authentication and authenticated encryption. DIAC 2012, 7 2012.

[33] Joan Daemen and Vincent Rijmen. AES Proposal: Rijndael, September 1999. https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf.

[34] Rafaël del Pino, Vadim Lyubashevsky, and Gregor Seiler. Short discrete log proofs for FHE and Ring-LWE ciphertexts. In *Public-Key Cryptography - PKC 2019 - 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Beijing, China, April 14-17, 2019, Proceedings, Part I*, pages 344–373, 2019.

[35] Frank Denis. Libsodium (v. 1.0.18), June 2019. https://libsodium.gitbook.io/doc/.

[36] Peter Gutmann / Digital Data Security Ltd. cryptlib (v. 3.4.4), January 2018. https://www.cs.auckland.ac.nz/~pgut001/cryptlib/index.html.

[37] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In *ACNS 2005*, volume LNCS 3531. Springer, 2005.

[38] Itai Dinur, Orr Dunkelman, and Adi Shamir. Collision attacks on up to 5 rounds of SHA-3 using generalized internal differentials. In *Fast Software Encryption (FSE)*, volume 8424 of *LNCS*, pages 219–240, Singapore, March 2013. Springer.

[39] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In *CRYPTO (2)*, volume 11693 of *Lecture Notes in Computer Science*, pages 356–383. Springer, 2019.

DRAFT

[40] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 40–56, 2013.

[41] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):238–268, 2018.

[42] Léo Ducas and Daniele Micciancio. Improved short lattice signatures in the standard model. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 335–352, 2014.

[43] D. Eastlake and T. Hansen. US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF). RFC 6234, May 2011. https://rfc-editor.org/rfc/rfc6234.txt.

[44] Nigel P. Smart (Ed.). ECRYPT-CSA: Algorithms, key size and protocols report. Deliverable D5.4, 2018.

[45] Thomas Espitau, Pierre-Alain Fouque, and Pierre Karpman. Higher-order differential meet-in-the-middle preimage attacks on SHA-1 and BLAKE. In *Advances in Cryptology (CRYPTO)*, volume 9215 of *LNCS*, pages 683–701, Santa Barbara, CA, August 2015. Springer.

[46] Nicolas Aragon et al. BIKE – Bit Flipping Key Encapsulation. https://bikesuite.org.

[47] European Telecommunications Standards Institute (ETSI). ETSI TS 102 825-5: Digital video broadcasting (DVB); content protection and copy management (DVB-CPCM); Part 5: CPCM security toolbox, February 2011. https://www.etsi.org/deliver/etsi_ts/102800_102899/10282505/01.02.01_60/ts_10282505v010201p.pdf.

[48] European Telecommunications Standards Institute (ETSI). ETSI TS 102 822-5-1: Broadcast and on-line services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 5: Rights management and protection (RMP); Sub-part 1: Information for broadcast applications, December 2012. https://www.etsi.org/deliver/etsi_ts/102800_102899/1028220501/01.07.01_60/ts_1028220501v010701p.pdf.

[49] European Telecommunications Standards Institute (ETSI). ETSI TS 103 127: Digital video broadcasting (DVB); content scrambling algorithms for DVB-IPTV services using MPEG2 transport streams, May 2013. https://www.etsi.org/deliver/etsi_ts/103100_103199/103127/01.01.01_60/ts_103127v010101p.pdf.

[50] European Telecommunications Standards Institute (ETSI). ETSI TS 102 225: Smart cards; secured packet structure for UICC based applications (release 13), July 2018. https://www.etsi.org/deliver/etsi_ts/102200_102299/102225/13.00.00_60/ts_102225v130000p.pdf.

[51] European Telecommunications Standards Institute (ETSI). ETSI TS 103 666-2: Smart secure platform (ssp); part 2: Integrated SSP (iSSP) characteristics (release 15), November 2019. https://www.etsi.org/deliver/etsi_ts/103600_103699/10366602/15.00.00_60/ts_10366602v150000p.pdf.

DRAFT

[52] European Telecommunications Standards Institute (ETSI). ETSI TS 119 312: Electronic signatures and infrastructures (ESI); cryptographic suites, February 2019. https://www.etsi.org/deliver/etsi_ts/119300_119399/119312/01.03.01_60/ts_119312v010301p.pdf.

[53] European Telecommunications Standards Institute (ETSI). ETSI TS 133 210: Digital cellular telecommunications system (phase 2+) (GSM); universal mobile telecommunications system (UMTS); LTE; 3G security; network domain security (NDS); IP network layer security, April 2019. https://www.etsi.org/deliver/etsi_ts/133200_139/133210/15.02.00_60/ts_133210v150200p.pdf.

[54] European Telecommunications Standards Institute (ETSI). ETSI TS 133 401: Digital cellular telecommunications system (phase 2+); universal mobile telecommunications system (UMTS); LTE; 3GPP system architecture evolution (SAE); security architecture, October 2019. https://www.etsi.org/deliver/etsi_ts/133400_133499/133401/15.09.00_60/ts_133401v150900p.pdf.

[55] S. Frankel, R. Glenn, and S. Kelly. The AES-CBC Cipher Algorithm and Its Use with IPsec. RFC 3602, September 2003. https://rfc-editor.org/rfc/rfc3602.txt.

[56] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *ACM symposium on Theory of Computing (STOC)*, pages 212–219, Philadelphia, PA, May 1996. ACM.

[57] Jian Guo, Pierre Karpman, Ivica Nikolić, Lei Wang, and Shuang Wu. Analysis of BLAKE2. In *RSA Conference*, volume 8366 of *LNCS*, pages 402–423, San Francisco, CA, February 2014. Springer.

[58] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, pages 267–288, 1998.

[59] Institute for Applied Information Processing and Communication. IAIK-JCE provider (v. 5.60), September 2019. https://jce.iaik.tugraz.at/.

[60] International Organization for Standardization. ISO/IEC 19772:2009, information technology - security techniques - authenticated encryption, February 2009. https://www.iso.org/standard/64575.html.

[61] International Organization for Standardization. ISO/IEC 18033-3:2010, information technology - security techniques - encryption algorithms - part 3: Block ciphers, December 2010. https://www.iso.org/standard/54531.html.

[62] International Organization for Standardization. ISO/IEC 9797-1:2011, information technology - security techniques - message authentication codes (MACs) - part 1: Mechanisms using a block cipher, March 2011. https://www.iso.org/standard/50375.html.

[63] International Organization for Standardization. ISO/IEC 9797-2:2011, information technology - security techniques - message authentication codes (MACs) - part 2: Mechanisms using a dedicated hash-function, May 2011. https://www.iso.org/standard/51618.html.

DRAFT

[64] International Organization for Standardization. ISO/IEC 29192-5:2016, information technology - security techniques - lightweight cryptography - part 5: Hash-functions, August 2016. https://www.iso.org/standard/67173.html.

[65] International Organization for Standardization. ISO/IEC 10116:2017, information technology - security techniques - modes of operation for an n-bit block cipher, July 2017. https://www.iso.org/standard/64575.html.

[66] International Organization for Standardization. ISO/IEC 10118-3:2018, information technology - security techniques - hash-functions - part 3: Dedicated hash-functions, October 2018. https://www.iso.org/standard/67116.html.

[67] International Telecommunication Union (ITU-T). Recommendation itu-t h.235.3 - series H: Audiovisual and multimedia systems, infrastructure of audiovisual services - systems aspects, September 2005. https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=H.235.3.

[68] International Telecommunication Union (ITU-T). Recommendation itu-t y.2704 - series Y: Global information infrastructure, internet protocol aspects and next-generation networks, January 2010. https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=Y.2704.

[69] International Telecommunication Union (ITU-T). Recommendation itu-t g.9807.1 - series G: Transmission systems and media, digital systems and networks, access networks - optical line systems for local and access networks, June 2016. https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=G.9807.1.

[70] International Telecommunication Union (ITU-T). Recommendation itu-t x.509 - series X: Data networks, open system communications and security, October 2016. https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=X.509.

[71] M. Jenkins and L. Zieglar. Commercial National Security Algorithm (CNSA) Suite Profile of Certificate Management over CMS. Internet-Draft draft-jenkins-cnsa-cmc-profile, Internet Engineering Task Force, May 2019. Work in Progress, https://datatracker.ietf.org/doc/html/draft-jenkins-cnsa-cmc-profile/.

[72] Jakob Jonsson. On the security of CTR + CBC-MAC. In *Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 76–93. Springer, 2002.

[73] Dmitry Khovratovich, Christian Rechberger, and Alexandra Savelieva. Bicliques for preimages: attacks on Skein-512 and the SHA-2 family. In *Fast Software Encryption (FSE)*, volume 7549 of *LNCS*, pages 244–263, Washington, DC, USA, March 2012. Springer.

[74] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III*, pages 552–586, 2018.

[75] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-Hashing for Message Authentication. RFC 2104, February 1997. https://rfc-editor.org/rfc/rfc2104.txt.

DRAFT

[76] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 75(3):565–599, 2015.

[77] Legion of the Bouncy Castle Inc. Bouncy castle (v. 1.62), June 2019. https://bouncycastle.org/.

[78] Legion of the Bouncy Castle Inc. Java FIPS roadmap, October 2019. https://www.bouncycastle.org/fips_java_roadmap.html.

[79] Qipeng Liu and Mark Zhandry. Revisiting post-quantum Fiat-Shamir. In *CRYPTO (2)*, volume 11693 of *Lecture Notes in Computer Science*, pages 326–355. Springer, 2019.

[80] Jack Lloyd. Botan (v. 2.11.0), July 2019. https://botan.randombit.net/.

[81] ARM Ltd. mbed TLS (v. 2.16.2), June 2019. https://tls.mbed.org/.

[82] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, pages 598–616, 2009.

[83] Vadim Lyubashevsky and Gregory Neven. One-shot verifiable encryption from lattices. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, pages 293–323, 2017.

[84] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, pages 1–23, 2010.

[85] Vadim Lyubashevsky and Gregor Seiler. NTTRU: truly fast NTRU using NTT. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):180–201, 2019.

[86] M. Matsui, J. Nakajima, and S. Moriai. A Description of the Camellia Encryption Algorithm. RFC 3713, April 2004. https://rfc-editor.org/rfc/rfc3713.txt.

[87] D. McGrew and D. Bailey. AES-CCM Cipher Suites for Transport Layer Security (TLS). RFC 6655, July 2012. https://rfc-editor.org/rfc/rfc6655.txt.

[88] David A. McGrew and John Viega. The security and performance of the galois/counter mode (GCM) of operation. In *INDOCRYPT*, volume 3348 of *Lecture Notes in Computer Science*, pages 343–355. Springer, 2004.

[89] MIRACL Labs. Apache milagro crypto library (amcl) (v. 3.2), December 2018. https://tls.mbed.org/.

[90] Niels Möller. Nettle (v. 3.5.1), June 2019. http://www.lysator.liu.se/~nisse/nettle/.

[91] Paweł Morawiecki, Josef Pieprzyk, and Marian Srebrny. Rotational cryptanalysis of round-reduced Keccak. In *Fast Software Encryption (FSE)*, volume 8424 of *LNCS*, pages 241–262, Singapore, March 2013. Springer.

DRAFT

[92] Mozilla. Network security services (NSS) (v. 3.45), July 2019. https://developer.mozilla.org/en-US/docs/Mozilla/Projects/NSS.

[93] National Institute of Standards and Technology. FIPS-197: Advanced Encryption Standard (AES), November 2001. https://www.nist.gov/publications/advanced-encryption-standard-aes.

[94] National Institute of Standards and Technology. SP 800-38A: Recommendation for block cipher modes of operation: Methods and techniques, December 2001. https://csrc.nist.gov/publications/detail/sp/800-38a/final.

[95] National Institute of Standards and Technology. FIPS PUB 180-2: Secure Hash Standard (SHS), August 2002. https://csrc.nist.gov/publications/detail/fips/180/2/archive/2002-08-01.

[96] National Institute of Standards and Technology. SP 800-38C: Recommendation for block cipher modes of operation: the CCM mode for authentication and confidentiality, May 2004. https://csrc.nist.gov/publications/detail/sp/800-38c/final.

[97] National Institute of Standards and Technology. SP-800-38B: Recommendation for block cipher modes of operation: the (CMAC) mode for authentication, May 2005. https://csrc.nist.gov/publications/detail/sp/800-38b/final.

[98] National Institute of Standards and Technology. SP 800-38D: Recommendation for block cipher modes of operation: Galois/counter mode (GCM) and GMAC, November 2007. https://csrc.nist.gov/publications/detail/sp/800-38d/final.

[99] National Institute of Standards and Technology. FIPS-198-1: The keyed-hash message authentication code (HMAC), July 2008. https://csrc.nist.gov/publications/detail/fips/198/1/final.

[100] National Institute of Standards and Technology. SP SP800-108: Recommendation for key derivation using pseudorandom functions, October 2009. https://csrc.nist.gov/publications/detail/sp/800-108/final.

[101] National Institute of Standards and Technology. SP 800-38A Addendum: Recommendation for block cipher modes of operation: Three variants of ciphertext stealing for CBC mode, October 2010. https://csrc.nist.gov/publications/detail/sp/800-38a/addendum/final.

[102] National Institute of Standards and Technology. FIPS PUB 180-4: Secure Hash Standard (SHS), August 2015. https://doi.org/10.6028/NIST.FIPS.180-4.

[103] National Institute of Standards and Technology. FIPS PUB 202: SHA-3 standard: Permutation-based hash and extendable-output functions, August 2015. https://doi.org/10.6028/NIST.FIPS.202.

[104] National Institute of Standards and Technology. NIST SP-800-185: SHA-3 derived functions: cSHAKE, KMAC, TupleHash, and ParallelHash, December 2016. https://doi.org/10.6028/NIST.SP.800-185.

DRAFT

[105] National Institute of Standards and Technology. SP 800-57 Part 1 Rev. 4: Recommendation for key management, part 1: General, January 2016. https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-4/final.

[106] National Institute of Standards and Technology. SP 800-131A Rev. 2: Transitioning the use of cryptographic algorithms and key lengths, March 2018. https://csrc.nist.gov/publications/detail/sp/800-131a/rev-2/final.

[107] NIST: National Institute of Standards and Technology. PQC standardization process: Second round candidate announcement. https://csrc.nist.gov/news/2019/pqc-standardization-process-2nd-round-candidates.

[108] National Security Agency - Central Security Service (NSA-CSS). Commercial national security algorithm suite (CNSA), August 2015. https://apps.nsa.gov/iaarchive/programs/iad-initiatives/cnsa-suite.cfm.

[109] PQCRYPTO project. Initial recommendations of long-term secure post-quantum systems. Report D5.4, March 2015.

[110] E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446, August 2018. https://rfc-editor.org/rfc/rfc8446.txt.

[111] M-J. Saarinen and J-P. Aumasson. The BLAKE2 Cryptographic Hash and Message Authentication Code (MAC). RFC 7693, November 2015. https://rfc-editor.org/rfc/rfc7693.txt.

[112] J. Salowey, A. Choudhury, and D. McGrew. AES Galois Counter Mode (GCM) Cipher Suites for TLS. RFC 5288, August 2008. https://rfc-editor.org/rfc/rfc5288.txt.

[113] Senior Officials Group Information Systems Security. SOG-IS crypto evaluation scheme - agreed cryptographic mechanisms version 1.1, June 2018. https://www.sogis.eu/documents/cc/crypto/SOGIS-Agreed-Cryptographic-Mechanisms-1.1.pdf.

[114] S. Shen, X. Lee, R. Tse, W. Wong, and Y. Yang. The SM3 Cryptographic Hash Function. Internet-Draft draft-sca-cfrg-sm3-02, Internet Engineering Task Force, January 2018. Work in Progress, https://datatracker.ietf.org/doc/html/draft-sca-cfrg-sm3-02.

[115] S. Shen, Y. Mao, and NSS. Murthy. Using Advanced Encryption Standard Counter Mode (AES-CTR) with the Internet Key Exchange version 02 (IKEv2) Protocol. RFC 5930, July 2010. https://rfc-editor.org/rfc/rfc5930.txt.

[116] Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 513–159, Vancouver, Canada, November 2002. IEEE.

[117] JH. Song, R. Poovendran, J. Lee, and T. Iwata. The AES-CMAC Algorithm. RFC 4493, June 2006. https://rfc-editor.org/rfc/rfc4493.txt.

[118] Standardization Administration of the People's Republic of China. GB/T 32905-2016: Information security techniques - SM3 cryptographic hash algorithm, August 2016. http://www.gbstandards.org/GB_standards/GB-T%2032905-2016.html.

DRAFT

[119] Standardization Administration of the People's Republic of China. GB/T 32907-2016: Information security techniques - SM4 block cipher algorithm, August 2016. http://www.gbstandards.org/GB_standards/GB-T%2032907-2016.html.

[120] Emily Stark, Mike Hamburg, and Dan Boneh. Stanford Javascript Crypto Library (SJCL) (v. 1.0.8), November 2018. http://bitwiseshiftleft.github.io/sjcl/.

[121] The Crypto++ Project. Crypto++ 6.0.0 benchmarks. https://www.cryptopp.com/benchmarks.html.

[122] The Crypto++ Project. Crypto++ (v. 8.2), April 2019. https://www.cryptopp.com/.

[123] The GnuPG Project. Libgcrypt (v. 2.2.17), July 2019. https://gnupg.org/software/libgcrypt/.

[124] The GnuTLS Project. GnuTLS (v. 3.6.9), July 2019. https://www.gnutls.org/.

[125] The OpenSSL Project. OpenSSL (v. 1.1.1c), May 2019. https://www.openssl.org/.

[126] Trusted Computing Group. TCG algorithm registry, family "2.0", level 00 revision 01.27, February 2018. https://trustedcomputinggroup.org/resource/tcg-algorithm-registry/.

[127] Trusted Computing Group (TCG). Trusted platform module library - part 1: Architecture (Family 2.0, Revision 01.38), September 2016. https://trustedcomputinggroup.org/wp-content/uploads/TPM-Rev-2.0-Part-1-Architecture-01.38.pdf.

[128] R. Tse, W. Wong, and M-J. Saarinen. The SM4 Blockcipher Algorithm And Its Modes Of Operations. Internet-Draft draft-ribose-cfrg-sm4-10, Internet Engineering Task Force, April 2018. Work in Progress, https://datatracker.ietf.org/doc/html/draft-ribose-cfrg-sm4-10.

[129] S. Turner. SHA-3 Related Algorithms and Identifiers for PKIX. Internet-Draft draft-turner-lamps-adding-sha3-to-pkix-01, Internet Engineering Task Force, September 2017. Work in Progress, https://datatracker.ietf.org/doc/html/draft-turner-lamps-adding-sha3-to-pkix-01.

[130] Dominique Unruh. Post-quantum security of Fiat-Shamir. In *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, pages 65–95, 2017.

[131] B. Viguier, D. Wong, G. Van Assche, Q. Dang, and J. Daemen. Kangarootwelve. Internet-Draft draft-irtf-cfrg-kangarootwelve-00, Internet Engineering Task Force, August 2019. Work in Progress, https://datatracker.ietf.org/doc/draft-irtf-cfrg-kangarootwelve/.

[132] Virtual Applications and Implementations Research Lab. eBACS: ECRYPT benchmarking of cryptographic systems. http://bench.cr.yp.to/results-hash.html.

[133] Xiaoyun Wang and Hongbo Yu. SM3 cryptographic hash algorithm. *Journal of Information Security Research*, 2(11):983–994, January 2016.

DRAFT

[134] D. Whiting, R. Housley, and N. Ferguson. Counter with CBC-MAC (CCM). RFC 3610, September 2003. https://rfc-editor.org/rfc/rfc3610.txt.

[135] wolfSSL, Inc. wolfcrypt (v. 4.0.0), March 2019. https://www.wolfssl.com/products/wolfcrypt-2/.

[136] World Wide Web Consortium. W3C recommendation: XML encryption syntax and processing version 1.1, April 2013. https://www.w3.org/TR/xmlenc-core1/.

[137] World Wide Web Consortium. W3C recommendation: Web cryptography API, January 2017. https://www.w3.org/TR/WebCryptoAPI/.

A NIST PQC Competition Round-Two Schemes

This section gives an overview of the progress that has been made in the NIST PQC competition since the completion of FutureTPM deliverable D2.1. After the initial submission of 69 schemes to round one, in January 2019 the NIST announced the progression of 26 schemes to round two [107]. (Some of the original candidates were considered not competitive, others were merged together into one submission.) Also, the submitters were given the opportunity to adjust their schemes and alter their parameter sets. In particular, the parameter sizes of some round-two submissions considerably differ from those of the first round.

In the following table we list a selection of parameter sets for all round-two submissions. For each entry the key size, the ciphertext or signature size, as well as the underlying assumption and sub-assumption are listed.

The status entries are:

- 2nd round: Currently being evaluated in the second round.
- 1st round: Replaced by other parameter sets and not being evaluated in round two.
- Attacked: Attacked but neither broken nor withdrawn, i.e. the submitters may be able to mitigate the attack.
- Broken: Attacked and broken or withdrawn.

Parameter sets of round-one schemes which are not being considered in the second round are not listed. For more details, we refer to the original submission documents and the NIST status report for the second round [2].

For qTesla we only list the broken parameter sets and those that have not been attacked. Other variants were dropped by the submitters and are therefore only listed in the notes next to their corresponding broken parameter set.

DRAFT

Parameter Set	Assumption	Sub-assumption	Status	Cat.	$\|pk\|$	$\|sk\|$	$\|c\|$	Notes
NEWHOPE512-CPA-KEM	Lattice	RLWE	2nd round	1	928	896	1088	
NEWHOPE1024-CPA-KEM	Lattice	RLWE	2nd round	5	1824	1792	2176	
NEWHOPE512-CCA-KEM	Lattice	RLWE	2nd round	1	928	1888	1120	
NEWHOPE1024-CCA-KEM	Lattice	RLWE	2nd round	5	1824	3680	2208	
Kyber512	Lattice	MLWE	2nd round	1	800	1632	736	
Kyber768	Lattice	MLWE	2nd round	3	1184	2400	1088	
Kyber1024	Lattice	MLWE	2nd round	5	1568	3168	1568	
FrodoKEM-640	Lattice	LWE	2nd round	1	9616	19888	9720	
FrodoKEM-976	Lattice	LWE	2nd round	3	15632	31296	15744	
FrodoKEM-976	Lattice	LWE	2nd round	3	21520	43088	21632	
BIKE-1	Codes	short Hamming	2nd round	1	20326	1988	20326	
BIKE-1	Codes	short Hamming	2nd round	3	39706	3090	39706	
BIKE-1	Codes	short Hamming	2nd round	5	65498	4110	65498	
BIKE-2	Codes	short Hamming	2nd round	1	10163	1988	10163	
BIKE-2	Codes	short Hamming	2nd round	3	19853	3090	19853	
BIKE-2	Codes	short Hamming	2nd round	5	32749	4110	32749	
BIKE-3	Codes	short Hamming	2nd round	1	22054	1876	22054	
BIKE-3	Codes	short Hamming	2nd round	3	43366	2970	43366	
BIKE-3	Codes	short Hamming	2nd round	5	72262	4256	72262	
ntruhrss701	Lattice	NTRU	2nd round	3	1138	1450	1138	
ntruhrss4096821	Lattice	NTRU	2nd round	5	1230	1590	1230	
ntruhrss2048677	Lattice	NTRU	2nd round	3	930	1234	930	
ntruhrss2048509	Lattice	NTRU	2nd round	1	699	935	699	
sntrup653	Lattice	NTRU	2nd round	2	994	1518	897	
ntrulpr653	Lattice	NTRU	2nd round	2	897	1125	1025	
sntrup761	Lattice	NTRU	2nd round	3	1158	1763	1039	
ntrulpr761	Lattice	NTRU	2nd round	3	1322	1294	1184	
sntrup857	Lattice	NTRU	2nd round	4	1322	1463	1184	
ntrulpr857	Lattice	NTRU	2nd round	4	1184	1999	1312	
sntrup4591761	Lattice	NTRU	1st round	4	1218	1600	1047	
ntrulpr4591761	Lattice	NTRU	1st round	4	1047	1238	1175	

DRAFT

Parameter Set	Assumption	Sub-assumption	Status	Cat.	$\|pk\|$	$\|sk\|$	$\|c\|$	Notes
SIKEp434	Isogeny	Isogeny	2nd round	3	330	374	346	
SIKEp503	Isogeny	Isogeny	2nd round	3	378	434	402	
SIKEp610	Isogeny	Isogeny	2nd round	3	462	524	486	
SIKEp751	Isogeny	Isogeny	2nd round	3	564	644	596	
mceliece348864	Codes	Goppa	2nd round	1	261120	6452	128	
mceliece460896	Codes	Goppa	2nd round	3	524169	13568	188	
mceliece6688128	Codes	Goppa	2nd round	5	1044992	13892	240	
mceliece6960119	Codes	Goppa	2nd round	5	1047319	13908	226	
mceliece8192128	Codes	Goppa	2nd round	5	1357824	14080	240	
hqc-128-1	Codes	short Hamming	2nd round	1	6170	252	6423	
hqc-192-1	Codes	short Hamming	2nd round	3	10918	404	10981	
hqc-192-2	Codes	short Hamming	2nd round	3	11688	404	11749	
hqc-256-1	Codes	short Hamming	2nd round	5	15898	532	15960	
hqc-256-2	Codes	short Hamming	2nd round	5	16926	566	16984	
hqc-256-3	Codes	short Hamming	2nd round	5	17714	566	17777	
LAC-128	Lattice	RLWE	2nd round	1	544	512	712	
LAC-192	Lattice	RLWE	2nd round	3	1056	1024	1188	
LAC-256	Lattice	RLWE	2nd round	5	1056	1024	1424	
NTS-KEM (12,64)	Codes	Goppa	2nd round	1	319488	9248	1024	
NTS-KEM (13,80)	Codes	Goppa	2nd round	3	929760	17556	1296	
NTS-KEM (13,136)	Codes	Goppa	2nd round	5	1419704	19922	2024	
ROLLO-I-128	Codes	low rank	2nd round	1	465	40	465	
ROLLO-II-128	Codes	low rank	2nd round	1	1546	40	1674	
ROLLO-III-128	Codes	low rank	2nd round	1	643	40	1188	
ROLLO-I-256	Codes	low rank	2nd round	5	947	40	947	
ROLLO-II-256	Codes	low rank	2nd round	5	2493	40	2621	
ROLLO-III-256	Codes	low rank	2nd round	5	1138	40	2196	
LEDACrypt KEM	Codes	short Hamming	2nd round	1	4488	468	4488	DFR 2^{-64}
LEDACrypt KEM	Codes	short Hamming	2nd round	3	7240	660	7240	DFR 2^{-64}
LEDACrypt KEM	Codes	short Hamming	2nd round	5	11136	884	11136	DFR 2^{-64}

DRAFT

Parameter Set	Assumption	Sub-assumption	Status	Cat.	$\|pk\|$	$\|sk\|$	$\|c\|$	Notes
LEDACrypt KEM	Codes	short Hamming	2nd round	1	6520	468	6520	DFR 2^{-128}
LEDACrypt KEM	Codes	short Hamming	2nd round	3	12032	660	12032	DFR 2^{-128}
LEDACrypt KEM	Codes	short Hamming	2nd round	5	19040	884	19040	DFR 2^{-128}
R5ND_1 KEM_0d	Lattice	LWR/RLWE	2nd round	1	643	16	682	
R5ND_3 KEM_0d	Lattice	LWR/RLWE	2nd round	3	909	24	981	
R5ND_5 KEM_0d	Lattice	LWR/RLWE	2nd round	5	1178	32	1274	
R5ND_1 KEM_5d	Lattice	LWR/RLWE	2nd round	1	445	16	549	
R5ND_3 KEM_5d	Lattice	LWR/RLWE	2nd round	3	780	24	859	
R5ND_5 KEM_5d	Lattice	LWR/RLWE	2nd round	5	972	32	1063	
RQC-128	Codes	low rank	2nd round	1	853	40	1690	
RQC-192	Codes	low rank	2nd round	3	1391	40	2766	
RQC-256	Codes	low rank	2nd round	5	2248	40	4552	
LightSABER	Lattice	MLWE	2nd round	1	672	1568	736	
SABER	Lattices	MLWE	2nd round	3	992	2304	1088	
FireSABER	Lattice	MLWE	2nd round	5	1312	3040	1472	
BabyBear	Lattice	IMLWE	2nd round	1	804	40	917	
MamaBear	Lattice	IMLWE	2nd round	4	1194	40	1307	
PapaBear	Lattice	IMLWE	2nd round	5	1584	40	1697	

DRAFT

Parameter Set	Assumption	Sub-assumption	Status	Cat.	$\|pk\|$	$\|sk\|$	$\|signature\|$	Notes
Dilithium-1024x768	Lattice	Fiat-Shamir	2nd round	1	1184		2044	
Dilithium-1280x1024	Lattice	Fiat-Shamir	2nd round	2	1472		2701	
Dilithium-1536x1280	Lattices	Fiat-Shamir	2nd round	3	1760		3366	
Falcon-512	Lattices	Hash then sign	2nd round	1	897	768	617.38 (avg.)	
Falcon-1024	Lattices	Hash then sign	2nd round	4,5	1793	1280	1233.29 (avg.)	
GeMSS128	Multivariate	HFE	2nd round	1	352190	13440	258 (bits)	
GeMSS192	Multivariate	HFE	2nd round	1	1237960	34070	411 (bits)	
GeMSS256	Multivariate	HFE	2nd round	3	3040700	75890	576 (bits)	
LUOV	Multivariate	UOV	2nd round	1	11500	32	239	
LUOV	Multivariate	UOV	2nd round	3	35400	32	337	
LUOV	Multivariate	UOV	2nd round	5	82000	32	440	
MQDSS-31-48	Multivariate	Fiat-Shamir	Attacked	1,2	46	16	20854	
MQDSS-31-64	Multivariate	Fiat-Shamir	Attacked	3,4	64	24	43728	
picnic-L1-FS	Symmetric	ZKP	2nd round	1	32	16	32838 (avg.)	
picnic-L3-FS	Symmetric	ZKP	2nd round	3	48	24	74134 (avg.)	
picnic-L5-FS	Symmetric	ZKP	2nd round	5	64	32	128176 (avg.)	
qTESLA-p-I	Lattices	Fiat-Shamir	2nd round	1	14880	5184	2592	
qTESLA-p-III	Lattices	Fiat-Shamir	2nd round	3	38432	12352	5664	
qTESLA-I-s	Lattices	Fiat-Shamir	broken	1	1504	1216	2144	qTESLA-I dropped
qTESLA-II-s	Lattices	Fiat-Shamir	broken	2	2336	1600	2144	qTESLA-II dropped
qTESLA-III-s	Lattices	Fiat-Shamir	broken	3	3104	2368	2848	qTESLA-III dropped
qTESLA-V-s	Lattices	Fiat-Shamir	broken	5	6432	4672	5920	qTESLA-V dropped
qTESLA-V-size-s	Lattices	Fiat-Shamir	broken	5	6432	4672	4640	qTESLA-V-size dropped
Rainbow-Ia	Multivariate	UOV	2nd round	1	149000	93000	512 bits	
Rainbow-IIc	Multivariate	UOV	2nd round	3	710600	511400	1248 bits	
Rainbow-Vc	Multivariate	UOV	2nd round	5	1705500	1227100	1632 bits	
SPHINCS+-128s	Symmetric	Hash	2nd round	1	32	64	8080	
SPHINCS+-192s	Symmetric	Hash	2nd round	3	48	96	17064	
SPHINCS+-256s	Symmetric	Hash	2nd round	5	64	128	29792	

DRAFT

B DAA Construction

We briefly describe our progress with the construction of a quantum-resistant DAA scheme. A detailed analysis will appear in Deliverable D2.3.

Since DAA and group signatures share many similarities, we used the currently most efficient post-quantum group signature [34] (without a bound on the number of group members) as a starting point to our DAA construction. There are two main differences between group signatures and DAA schemes. The first is that DAA schemes do not have an opener, but instead it should be possible to link two signatures under the same basename. For this, we switch out the verifiable encryption [83] scheme used in [34] and instead use a Ring-LWE weak-PRF (e.g. [8]) together with a particular proof of knowledge from [14] that allows for linking under the same basename.

Another difference is that security of DAA schemes crucially requires that the authority giving out the TPM secret keys does not learn all the secret information that the TPM possesses. This feature is needed for defending against being framed by a corrupt authority. The group signature from [34] did not have such property and the entire secret key of every user was known to the authority. In [34], the secret key of the user was a signature of a selectively-secure standard model signature scheme of Agrawal, Boneh, and Boyen [1], where the user's identity serves as the message. In our DAA scheme, we will instead use the idea from the signature scheme of Ducas and Micciancio [42] where, in addition to a message, there is also a Tag. We observe that in this scenario, the message (i.e. the secret identity of the user) can be hidden from the signer, and the signer just signs (similarly to [1]) a Tag of his choosing. This idea allows us to obtain a user's secret key without revealing it to the authority. The description of our scheme can be found in [29]. An implementation will be done in the last part of the project.

DRAFT