A Status Update on Quantum Safe Cryptography

Martin Strand
Norwegian Defence Research Establishment
Email: martin.strand@ffi.no

***Abstract*—There is an ongoing effort to standardize asymmetric cryptography that should not be attackable by a quantum computer. The process is now into its final rounds. From 69 initial submissions from academia and industry, a small number of standardized schemes are expected in 2022–2024. The standards coming out from this work are expected to be used to secure civilian and military networks alike in the future, superseding current asymmetric techniques. In addition to providing an introduction to the process to this community, we also report on our own experiments with these schemes, highlighting which trade-offs one could prepare for.**

I. Introduction

STRONG cryptography is a prerequisite for secure communications in any domain. For the last 20 years, we have had access to efficient elliptic curve cryptography (ECC) and the Advanced Encryption Standard (AES) which in combination allows secure key exchange, signatures and traffic encryption. These techniques are publicly available and the security is solely based on the key being secret. In case of loss of equipment, this reduces risk, as no secret algorithm is involved. At the same time, these schemes have been studied extensively and are endowed with high confidence.

However, the ongoing process for standardizing post-quantum cryptography by the United States' National Institute of Standards of Technology (NIST) is a response to a worrying development which threatens the current golden age of secure communication. Due to the 1994 algorithm by Shor [1], we know that a sufficiently advanced quantum computer will be able to break our current techniques for asymmetric crypto, and the development of such computers have reached a level where we must consider our alternatives to for example ECC. Fortunately, AES remains safe, albeit with larger keys than one would have used by default 20 years ago.

The process is wholly civilian, but should also be of interest to a military research community outside cryptography. NIST standards receive substantial scrutiny, including from the US National Security Agency (NSA), and may later be deemed suitable for classified information from national authorities worldwide, as has been the case for previous NIST standards such as AES.

NIST keeps strong control over the standardization process to create trust in the end result. The community invested in the process is also watching closely to ensure that any decision is well-founded and understandable. As an added benefit, this should also serve to increase other national authorities' confidence in the final standards. The scrutiny originates from the history of a previously NIST standardized pseudorandom generator, which is now retracted due to documented backdoors [2] and allegations that the NSA used its influence to standardize the algorithm and had it included in popular software libraries [3]. Allegations are by themselves sufficient to erode trust in a process.

Due to the open nature of the process, the standards may be suitable for future deployment in field equipment likely to be lost or discarded. The crucial question is whether one can find a suitable combination of performance, power consumption and price.

This paper is intended as a gentle tutorial to the standardization process, the candidates, timeline, impact and a brief report on our own experimentation with these schemes on a small microcontroller.

We are specifically not going into in-depth discussions of the related field "quantum cryptography". An argument for this choice is included in Section V.

II. The Impact of the Quantum Threat

The concept of quantum computing has been known for decades, and although the promise of an advanced, working quantum computer has been repeatedly postponed, we cannot ignore it. Classified information must commonly be kept private for

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/ICMCIS52405.2021.9486413

TABLE I: Standardization Finalists and Alternates

	KEM		Signature	
	Name	Problem	Name	Problem
Finalists	Classic McEliece [4]	decoding	DILITHIUM [5]	lattices (MLWE, SIS)
	KYBER [6]	lattices (MLWE)	FALCON [7]	lattices (NTRU, SIS)
	NTRU [8]	lattices (NTRU)	Rainbow [9]	multivar. eq. (OUV)
	SABER [10]	lattices (MLWR)		
Alternates	BIKE [11]	decoding	GeMSS [12]	multivariate eq.
	FrodoKEM [13]	lattices (LWE)	Picnic [14]	hashfunc., zero-knowledge
	HQC [15]	decoding	SPHINCS+ [16]	hashfunctions
	NTRU Prime [17]	lattices (NTRU)		
	SIKE [18]	isogenies		

up to 30 years. Replacing an information system may require a full decade. Hence, any technique currently in use should remain secure up to even 40 years into the future. Unless one can reasonably assume that no powerful quantum computer is available by then, action should be taken already now.

The threat is most dire against asymmetric encryption. Schemes like RSA encryption, Diffie-Hellman key exchange and the Digital Signature Algorithm (DSA) are based on algebra with a finite number of elements. Shor demonstrated how a quantum computer can find periods in these structures, and these periods can be used to factor large numbers or compute discrete logarithms. The running time for factorization then becomes polynomial in the input number.

The commonly deployed symmetric encryption schemes do not contain these structures, and are therefore not susceptible to attacks based on Shor's algorithm. A more general algorithm by Grover [19] can search an unstructured set of n elements in time $O(\sqrt{n})$, in theory reducing the security of a 128 bit encryption scheme to 64 bit. However, when taking concrete computational overhead into consideration the advantage diminishes considerably. An estimation by Jaques, Naehrig, Roetteler and Virdia [20] gives the following figures for concrete attack times:

- AES-128: $2^{83.4}$ operations
- AES-192: $2^{126.1}$ operations
- AES-256: $2^{190.5}$ operations

In words, this means that our current symmetric techniques most probably will remain secure against quantum computers, also at lower key sizes.

III. The NIST Process and Candidates

The NIST Post-Quantum Cryptography Standardization process started in 2015–2016, with a call for proposals posted 20 December 2016. Both encryption and signature schemes were solicited. One year later, NIST announced that they had accepted 69 proposals as "complete and proper". Following internal, public and scientific scrutiny, 26 algorithms prevailed into the second round, which was announced on 30 January 2019. The third round finalists were presented on 22 July 2020. The statement listed seven finalists that could be considered for standardization at the end of this round, and another eight alternate candidates that may enter a fourth round for future standardization. The finalists and alternates are listed in Table I.

NIST's outspoken goal for the process is to provide full transparency of the process, and they are soliciting contributions from the scientific community with regards to attacks, security proofs, implementations and other studies of the candidates. The process is receiving significant attention. Draft standards are expected to be ready during 2022–2024.

A. Families of Candidates

One can hardly discuss modern cryptograhy without mentioning the underlying mathematics. The schemes are designed so that – in theory – any adversary able to decrypt a message or forge a signature should also be able to solve a mathematical problem believed to be hard. Classical schemes are often said to be based on the discrete logarithm problem (e.g., Diffie-Hellman key exchange) or factorization (e.g., the RSA cryptosystem). This is not completely precise, as these schemes have spawned their own particular computational problems, for which one possible way to solve would be the aforementioned problems.

The original NIST call asked for submissions in three categories: key encapsulation mechanisms

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/ICMCIS52405.2021.9486413

(KEM), public-key encryption schemes, and signatures. A KEM is a scheme where one can always assume that the message is a short, random message (i.e., a key), whereas an encryption scheme should accept any sufficiently short message. Only KEMs and signatures made it to this point in the process, which perhaps reflects on the fact that a common use of public-key encryption is to encrypt keys for use in symmetric encryption:

- In the case of TLS, certificates, signatures and Diffie-Hellman key exchange are used to set up a key for a symmetric sceme, which will be used to encrypt the communication data [21].
- Pretty Good Privacy (PGP) uses a symmetric encyption scheme to encrypt the message, and the key for this scheme is then encrypted using the public key of the recipient [22].

Signatures are used to attest to the validity of a message. One can publish a verification key while keeping the signing key private. Any message signed using the signing key can be verified by any holder of the verification key, and the verification should only be successful if the message was indeed signed using the right signing key.

The finalists and alternates are each based on one or more problems from the following families: general decoding, lattices, multivariate equations, elliptic curve isogenies, hash funtions, and zero-knowledge proofs. Introducing the mathematical details for each of these families is out of scope for for this text. Instead, we try to give a high-level overview of the intuition behind some of the problems:

- The decoding problem is the oldest, and possibly the most accessible for the communcations community. It is common to protect a message against errors during transmission by using error-correcting codes. One can turn this into a encryption scheme by publishing (parameters for) the encoding algorithm while keeping the data needed for decoding private. During encryption, the sender randomly adds a number of errors to the encoding, so that only the decoding party can recover the original message. For general linear codes, this is known to be NP-hard. The oldest such scheme is that of McEliece [23], variants of which are submitted to the NIST process. It is very likely that one such scheme will be standardized due to its decades long history. The disadvangate is that the keys need to be very large to provide sufficient security.
- The lattice-based schemes seem to hit a combination of speed, efficieny and security. Three out of the four finalists in the encryption category are based on lattices, and so are two of the three signature finalists. At most one of the encryption schemes is expected to be standardized. Lattice problems are tightly connected to linear algebra. The particular problem flavours represented among the finalists are essentially large systems of linear equations, but with a small added error from a narrow discrete Gaussian distribution. One can instantiate these problems with or without extra algebraic structure. The extra structure enables quicker computations, but is a less conservative choice in terms of security.
- One of the signature schemes is based on solving systems of multivariate polynomials. While solving a system of linear equations is easy, it becomes NP-hard when the terms are quadratics. In particular, the public key can be viewed as the composition of a small number of functions, and the private key is the collection of individual functions, which makes it easier to invert the quadratic system.
- The final problem we want to discuss here is that of finding isogenies. This is perhaps the least accessible family. The problem asks the attacker to reverse engineer a special function between two given elliptic curves. If the adversary is unable to do so, then we can use such mappings to do key exchange by agreeing on a certain characteristic of a hidden curve in a Diffie-Hellman-like protocol. One such scheme is among the alternates. It enjoys small keys and ciphertexts, but requires more processor time than most of its competitors. More investigation on the concrete hardness of finding isogenies is still needed.

B. Security Requirements

The submitters were asked to provide parameter sets for different security levels. Security is fundamentally connected to the computers we expect our adversaries will have access to within the forseeable future. For quantum computers, this is clearly

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/ICMCIS52405.2021.9486413

unknown today. NIST have therefore chosen to tie their security goals to two existing primitives, AES and the standardized secure hashing algorithm SHA-2. To reach security category 1, the scheme must be at least as hard to break as AES-128. The highest level – category 5 – requires security equivalent to AES-256. The complete listing is as follows:

Cat. 1: as secure as AES-128
Cat. 2: as secure as SHA-256
Cat. 3: as secure as AES-196
Cat. 4: as secure as SHA-384
Cat. 5: as secure as AES-256

We have previously discussed the concrete security of AES against quantum computers. The submitters have been allowed to adjust their parameters based on new research on AES and SHA-2 security and as a consequence of new attacks on their own schemes.

We note that the established naming convention may be confusing. SHA- 256 and SHA-384 are both variants of SHA-2, but with different output length. The relevant security target in this instance is the cost of finding a collision in SHA-256 and SHA-384 resp. AES security is measured as the cost of finding the encryption key.

IV. Testing on Microcontrollers

Each submitter was required to include a reference implementation of their proposal. The PQM4 project gathers these implementations with the necessary modifications for them to run on the ARM Cortex M4 processor. Recall that the schemes had different parameter sizes. In particular, the submission named "Classic McEliece" is not included in the PQM4 library due to its large memory footprint.

The library consists of two sets of schemes, one for KEMs and one for signatures. Each scheme comes i several variants, one for each parameter set. Every KEM variant exposes three functions:

- key generation `crypto_kem_keypair`
- encapsulation `crypto_kem_enc`
- decapsulation `crypto_kem_dec`

The signature API contains:

- key generation `crypto_sign_keypair`
- signing `crypto_sign`
- verify and unpack `crypto_sign_open`

For brevity and clarity, we use the slightly more condensed notation KEM.{keypair, enc, dec} and SIGN.{keypair, sign, open}. Secret (private) and public keys are denoted by *sk* and *pk* respectively.

We refer to the PQM4 project [24] for further implementation details and detailed benchmarks.

During the summer of 2020, we challenged two undergraduate students to implement a simple key agreement protocol on a pair of microcontrollers in order to get hands-on experience with the kind of performance one can expect from systems like this [25]. Our code is available on GitHub [26]. The protocol is displayed in Fig. 1. There, pk_{KEM} is the public key encapsulation key, and $\mathsf{sign}(pk_{\text{KEM}})$ is the signature on the key. The response $\text{KEM.enc}(k_{\text{session}})$ is a symmetric key, encapsulated using the key encapsulation mechanism under the key pk_{KEM}. Notice that we omit keys when they are implicitly given by the context.

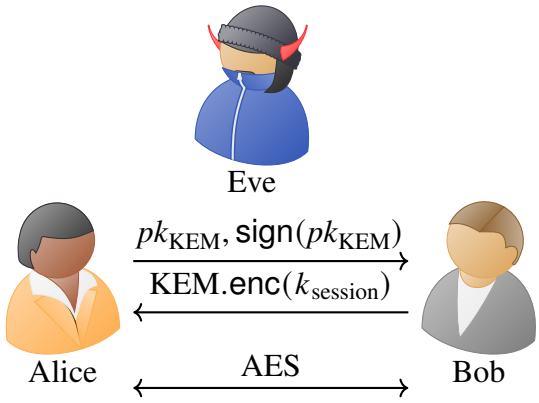

Fig. 1: Demonstration Protocol Between Alice and Bob in the Presence of the Adversary Eve

Our hardware of choice was a Teensy 3.6 with a 180 MHz 32 bit processor and 256 kB RAM. The device is Arduino compatible, so we could control pins, serial communication and lights with simple instructions. Two such devices were connected to a computer, and interconnected using short jumper cables. This enabled communication between the protocol parties. Our code contains instructions on how to test the protocol with any of the available schemes. Once the key agreement process is completed, the common key is loaded into a software AES implementation. This allows the devices to receive a message from the computer, encrypt it and send it to its peer, where it is decrypted and sent back to the computer. This is essentially the operation of dedicated crypto equipment, and demonstrates the feasibility of using these algorithms in the future. Please keep in mind that this is non-optimized software running on a USD 4 processor when reading Tables II and III.

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/ICMCIS52405.2021.9486413

As such, the differences in orders of magnitude is more important than the concrete numbers. Final implementations can use dedicated hardware or FPGAs to do the same operations, thereby improving efficiency considerably.

The devices play one out of two roles, *initiator* or *responder*. Fig. 2a and 2b show the pseudocode for these two roles.

V. Quantum Key Distribution

Quantum key distribution (QKD) is often touted as the solution to the threat posed by quantum computers. Based on the laws of physics, QKD is unconditionally secure against a passive eavesdropper. The idea is based on the fact that any observer who tries to measure a stream of photons will modify the stream in the process, and in doing so introduce errors that will allow the parties to detect the adversary.

However, there are several problems concerning QKD. It is vulnerable to an active adversary, so the parties still need mutual authentication. An active adversary could pose as Bob towards Alice, and as Alice towards Bob. Then Alice and Bob would end up with two different sets of keys, and all traffic being decrypted and reencrypted by the adversary. Furthermore, QKD requires a dedicated infrastructure capable of transmitting photons unaltered between two endpoints. It seems unlikely that such a network will be provided between arbitrary IoT devices.

Although China is in the process of deploying a large scale quantum key distribution network using satellites, we do not expect this technology to offer any practical competition to the upcoming NIST standards.

A. Going Forward

It is likely that future operations will use a number of small, affordable, and unmanned devices and sensors beyond enemy lines. Their usefulness can be even better if they can communicate between coalition partners. For instance, one partner may be able to deploy ground sensors, another may have drones to harvest information, and these may leave and join the battlefield, and potentially fall into the hands of an advanced and motivated adversary. There should therefore not be any crucial long- term keys onboard these devices, which calls for key agreement as they go.

This process is thus good news for future operations, and our demonstrations also serves to indicate that it is feasible to use these algorithms on very small devices. We should expect new implementations for even smaller devices in the upcoming years.

In parallel to this, there is much work left to do with regards to side channel attacks on these small

```
set up serial interfaces (0, 1)
generate KEM keypair (pk_KEM, sk_KEM)
generate signature keypair (pk_SIGN, sk_SIGN)
set pk_KEM||s to SIGN.sign_{sk_SIGN}(pk_KEM)
send (pk_SIGN, pk_KEM||s) to responder

wait for c from responder on serial 1
set k to KEM.dec(c)
initiate AES-GCM with k

repeat:
  wait for input on serial interfaces
  on input from responder:
    decrypt
    print to serial 0
  on input from serial 0:
    encrypt
    print to serial 1
```

(a) Initiator

```
set up serial interfaces (0, 1)
wait for (pk_SIGN, pk_KEM||s) from initiator on
    serial 1
if not SIGN.open_{pk_SIGN}(pk_KEM||s):
  halt
set k to 32 random bytes
set c to KEM.enc_{pk_KEM}(k)
send c to initiator on serial 1
initiate AES-GCM with k

repeat:
  wait for input on serial interfaces
  on input from responder:
    decrypt
    print to serial 0
  on input from serial 0:
    encrypt
    print to serial 1
```

(b) Responder

Fig. 2: Roles in the protocol

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/ICMCIS52405.2021.9486413

TABLE II: Benchmarking of KEM Variants. Times in Milliseconds, Sizes in Bytes.

Cat.	System	keypair	enc	dec	Pub. key	Pri. key	Ciphertext	Full name
1	KYBER	3	4	3	800	1632	736	kyber512
	NTRU	998	6	7	930	1234	930	ntruhps2048677
	NTRU	1066	3	8	1138	1450	1138	ntruhrss701
	SABER	3	5	5	672	1568	736	lightsaber
	FrodoKEM	463	455	451	9616	19888	9720	frodokem640shake
	SIKE	270	440	470	330	374	346	sikep434
2	SIKE	379	619	662	378	434	402	sikep503
3	KYBER	5	6	6	1184	2400	1088	kyber768
	NTRU	1462	7	10	1230	1590	1230	ntruhps4096821
	SABER	6	8	9	992	2304	1088	saber
	FrodoKEM	1005	1009	1000	15632	31296	15744	frodokem976shake
	NTRU Prime	2186	5	7	1158	1763	1039	sntrup761
	NTRU Prime	1736	9	11	1039	1294	1167	ntrulpr761
	SIKE	673	1234	1241	462	524	486	sikep610
5	KYBER	9	10	9	1568	3168	1568	kyber1024
	SABER	10	13	14	1312	3040	1472	firesaber
	SIKE	1178	1918	2059	564	644	596	sikep741

TABLE III: Benchmarking of Signature Variants. Times in Milliseconds, Sizes in Bytes.

Cat.	System	keypair	sign	open	Pub. key	Pri. key	Ciphertext	Full name
1	DILITHIUM	9	24	9	1184	2800	2044	dilithium2
	FALCON	1049	233	4	1281	897	690	falcon-512
	FALCON	1136	110	4	57344	897	690	falcon-512-tree
2	DILITHIUM	13	23	14	1472	3504	2701	dilithium3
3	DILITHIUM	18	30	19	1760	3856	3366	dilithium4
4/5	FALCON	1760	505	7	1793	2305	1330	falcon-1024

devices. This is particularly important in scenarios where we assume the adversary to be able to capture active and deployed equipment. We encourage any research group able to produce new results in this field to share these with the post quantum cryptography community, such that our future standards will be as good as possible.

VI. Conclusion

The cryptographic community is taking the quantum threath seriously, and strong asymmetric schemes will be available within few years. Commercial and civilian applications is still the main focus of the process. However, in a recent announcement, NSA stated that they expect that lattice-based signature and key encapsulation schemes from the NIST process would be approved for US National Security Systems. This means that we should expect to find these schemes in future communication protocols, and interested parties should perhaps already now start considering this future development.

The reseach community in collaboration with a handful of large internet companies have started real-life tests of these algorithms. Attention is in particular being given to the TLS protocol [27], which powers secure services on the internet. Recent work from Bosch have integrated post-quantum algorithms in industrial machine-to-machine communications [28]. However, more attention is needed regarding acceptable performance, security level and usecases. Interested parties are highly encouraged to join the NIST "PQC Forum" mailing list and voice their needs, opinions and results from practical experiments.

Acknowledgment

The author would like to thank the students Ella Wolff Kristensen and Ludvig Ellingsen for their hard and excellent work during the special summer of 2020. The results would not have been possible without the kind assistance from the PQM4 team.

References

[1] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *35th FOCS*. Santa Fe, NM, USA: IEEE Computer Society Press, Nov. 20–22, 1994, pp. 124–134.

[2] D. R. L. Brown and K. Gjøsteen, "A security analysis of the NIST SP 800–90 elliptic curve random number generator," in *CRYPTO 2007*, ser. LNCS, A. Menezes, Ed., vol. 4622. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, Aug. 19–23, 2007, pp. 466–481.

DOI til publisert versjon/DOI to published version: 10.1109/ICMCIS52405.2021.9486413

[3] J. Ball, J. Borger, and G. Greenwald, "Revealed: how us and uk spy agencies defeat internet privacy and security," *The Guardian*, sep 6 2013, https://www.theguardian.com/world/2013/sep/05/nsa-gchq-encryption-codes-security. Accessed 2020-12-17.

[4] D. J. Bernstein, T. Chou, T. Lange, I. von Maurich, R. Misoczki, R. Niederhagen, E. Persichetti, C. Peters, P. Schwabe, N. Sendrier, J. Szefer, and W. Wang, "Classic McEliece," National Institute of Standards and Technology, Tech. Rep., 2019, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[5] V. Lyubashevsky, L. Ducas, E. Kiltz, T. Lepoint, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS-DILITHIUM," National Institute of Standards and Technology, Tech. Rep., 2019, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[6] P. Schwabe, R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, G. Seiler, and D. Stehlé, "CRYSTALS-KYBER," National Institute of Standards and Technology, Tech. Rep., 2019, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[7] T. Prest, P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang, "FALCON," National Institute of Standards and Technology, Tech. Rep., 2019, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[8] Z. Zhang, C. Chen, J. Hoffstein, W. Whyte, J. M. Schanck, A. Hulsing, J. Rijneveld, P. Schwabe, and O. Danba, "NTRUEncrypt," National Institute of Standards and Technology, Tech. Rep., 2019, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[9] J. Ding, M.-S. Chen, A. Petzoldt, D. Schmidt, and B.-Y. Yang, "Rainbow," National Institute of Standards and Technology, Tech. Rep., 2019, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[10] J.-P. D'Anvers, A. Karmakar, S. S. Roy, and F. Vercauteren, "SABER," National Institute of Standards and Technology, Tech. Rep., 2019, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[11] N. Aragon, P. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Gueron, T. Guneysu, C. Aguilar Melchor, R. Misoczki, E. Persichetti, N. Sendrier, J.-P. Tillich, G. Zémor, and V. Vasseur, "BIKE," National Institute of Standards and Technology, Tech. Rep., 2019, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[12] A. Casanova, J.-C. Faugère, G. Macario-Rat, J. Patarin, L. Perret, and J. Ryckeghem, "GeMSS," National Institute of Standards and Technology, Tech. Rep., 2019, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[13] M. Naehrig, E. Alkim, J. Bos, L. Ducas, K. Easterbrook, B. LaMacchia, P. Longa, I. Mironov, V. Nikolaenko, C. Peikert, A. Raghunathan, and D. Stebila, "FrodoKEM," National Institute of Standards and Technology, Tech. Rep., 2019, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[14] G. Zaverucha, M. Chase, D. Derler, S. Goldfeder, C. Orlandi, S. Ramacher, C. Rechberger, D. Slamanig, J. Katz, X. Wang, and V. Kolesnikov, "Picnic," National Institute of Standards and Technology, Tech. Rep., 2019, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[15] C. Aguilar Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, E. Persichetti, and G. Zémor, "HQC," National Institute of Standards and Technology, Tech. Rep., 2019, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[16] A. Hulsing, D. J. Bernstein, C. Dobraunig, M. Eichlseder, S. Fluhrer, S.-L. Gazdag, P. Kampanakis, S. Kolbl, T. Lange, M. M. Lauridsen, F. Mendel, R. Niederhagen, C. Rechberger, J. Rijneveld, P. Schwabe, and J.-P. Aumasson, "SPHINCS+," National Institute of Standards and Technology, Tech. Rep., 2019, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[17] D. J. Bernstein, C. Chuengsatiansup, T. Lange, and C. van Vredendaal, "NTRU Prime," National Institute of Standards and Technology, Tech. Rep., 2019, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[18] D. Jao, R. Azarderakhsh, M. Campagna, C. Costello, L. De Feo, B. Hess, A. Jalali, B. Koziel, B. LaMacchia, P. Longa, M. Naehrig, J. Renes, V. Soukharev, D. Urbanik, and G. Pereira, "SIKE," National Institute of Standards and Technology, Tech. Rep., 2019, available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[19] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *28th ACM STOC*. Philadephia, PA, USA: ACM Press, May 22–24, 1996, pp. 212–219.

[20] S. Jaques, M. Naehrig, M. Roetteler, and F. Virdia, "Implementing grover oracles for quantum key search on AES and LowMC," in *EUROCRYPT 2020, Part II*, ser. LNCS, A. Canteaut and Y. Ishai, Eds., vol. 12106. Zagreb, Croatia: Springer, Heidelberg, Germany, May 10–14, 2020, pp. 280–310.

[21] E. Rescorla, "The transport layer security (tls) protocol version 1.3," Internet Requests for Comments, RFC Editor, RFC 8446, August 2018.

[22] J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer, "Openpgp message format," Internet Requests for Comments, RFC Editor, RFC 4880, November 2007, http://www.rfc-editor.org/rfc/rfc4880.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4880.txt

[23] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, "On the inherent intractability of certain coding problems (corresp.)," *IEEE Trans. Inf. Theory*, vol. 24, no. 3, pp. 384–386, 1978. [Online]. Available: https://doi.org/10.1109/TIT.1978.1055873

[24] M. J. Kannwischer, J. Rijneveld, P. Schwabe, and K. Stoffelen, "PQM4: Post-quantum crypto library for the ARM Cortex-M4," https://github.com/mupq/pqm4.

[25] E. W. Kristensen, L. Ellingsen, and M. Strand, "Testing av kvantesikre kandidatalgoritmer på en mikrokontroller – Norges sikreste chat," FFI, Tech. Rep. 20/02837, dec 2020.

[26] ——, "PQCTeensy," 2020, https://github.com/ForsvaretsForskningsinstitutt teensy-post-quantum. Accessed 2020-12-17.

[27] C. Paquin, D. Stebila, and G. Tamvada, "Benchmarking post-quantum cryptography in TLS," in *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020, Paris, France, April 15-17, 2020, Proceedings*, ser. Lecture Notes in Computer Science, J. Ding and J. Tillich, Eds., vol. 12100. Springer, 2020, pp. 72–91. [Online]. Available: https://doi.org/10.1007/978-3-030-44223-1_5

[28] S. Paul and P. Scheible, "Towards post-quantum security for cyber-physical systems: Integrating PQC into industrial M2M communication," in *Computer Security - ESORICS 2020 - 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14-18, 2020, Proceedings, Part II*, ser. Lecture Notes in Computer Science, L. Chen, N. Li, K. Liang, and S. A. Schneider, Eds., vol. 12309. Springer, 2020, pp. 295–316. [Online]. Available: https://doi.org/10.1007/978-3-030-59013-0_15

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/ICMCIS52405.2021.9486413