ETSI TR 103 965 V1.1.1 (2024-10)

TECHNICAL REPORT

CYBER; Quantum-Safe Cryptography (QSC); Impact of Quantum Computing on Cryptographic Security Proofs

Reference

DTR/CYBER-QSC-0020

Keywords

Quantum Safe Cryptography, security

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from the ETSI Search & Browse Standards application.

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on ETSI deliver.

Users should be aware that the present document may be revised or have its status changed, this information is available in the Milestones listing.

If you find errors in the present document, please send your comments to the relevant service listed under Committee Support Staff.

If you find a security vulnerability in the present document, please report it through our Coordinated Vulnerability Disclosure (CVD) program.

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Cyber Security (CYBER).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

There is a common misconception that to make a classically secure cryptosystem quantum-safe, it suffices to replace its underlying computational-hardness assumptions with "quantum-hard" assumptions. However, this is not always the case. The present document provides an overview of the impact of quantum computing on cryptographic security proofs; it illustrates how for certain classes of cryptographic systems the security proofs need to be adapted, for which classes this has already successfully been done, and what the practical implications of these adaptations are.

The present document is meant for cryptographic experts who want to get insight into practical changes that need to be made to existing systems to make those systems quantum-safe, or who want to understand the fundamental challenges in proving security against a quantum adversary.

Introduction

The advent of a cryptographically-relevant quantum computer (or CRQC for short) will severely impact most currently-used cryptographic systems. Notably, a CRQC can factor integers and compute discrete logarithms in polynomial time, thereby breaking systems based on the hardness of these problems.

However, simply replacing these problems by others which are (believed to be) impervious even to a quantum computer does not completely solve the issue. This is due to the fact that many security proofs of cryptographic systems are no longer valid in the presence of a quantum-capable attacker; while this does not automatically imply that the affected systems would be broken by a quantum computer, it does raises questions on the exact security guarantees that the systems can provide.

The present document analyses the impact of quantum computers on cryptographic security proofs, describing the current knowledge on the topic and the expected effects on security.

1 Scope

The present document is intended to provide an overview of the impact of quantum computing on the security proofs of several cryptographic protocols. It focuses on cryptographic protocols that can be run on classical hardware; further, it discusses which security proofs are invalidated, or otherwise affected, in the presence of an attacker with access to a CRQC, and discusses for each affected system whether:

a) an alternative proof has been found that does provide security against quantum attacks, but possibly with a reduced security level;

b) no alternative proof has been found, but security is expected to still hold;

c) the cryptographic system is expected to be broken by quantum attacks, in a way which is not captured by the classical security proof, although no concrete quantum attack exists yet; or

d) a concrete quantum attack that breaks security, in a way which is not captured by the classical proof, is available.

In terms of the security proofs and problems under consideration, the present document includes the following:

1) The quantum random oracle model, and in particular its usage in:

   a) The Fiat-Shamir transformation.

   b) The Fujisaki-Okamoto transformation.

2) The rewinding technique for zero-knowledge proof systems.

3) The binding property of commitment schemes.

4) The universal-composability framework.

5) The indifferentiability framework.

6) Security proofs of pseudo-random functions.

In addition to presenting the theoretical developments on these topics, the present document elaborates on the practical consequences. In some cases, the security of classically secure schemes is uncertain in the face of a quantum adversary. In other cases, the security of the scheme holds, but the parameters need to be adjusted to retain the same level of security.

NOTE: The present document does not discuss so-called "quantum-annoying" schemes, which still base their security on computational problems that can be solved (relatively) efficiently by a quantum computer, but force such an attack to perform a high number of operations, hence making it impractical for the expected first generation of quantum computers.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long-term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] IETF RFC 2313 (1998): "PKCS# 1: RSA encryption version 1.5".

[i.2] D. Bleichenbacher: "Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS# 1". Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 1998.

[i.3] N. Koblitz, A.J. Menezes: "The random oracle model: a twenty-year retrospective". Designs, Codes and Cryptography 77.2 (2015): pp. 587-610.

[i.4] R. Canetti, et al.: "The random oracle methodology, revisited". Journal of the ACM (JACM) 51.4 (2004): pp. 557-594.

[i.5] J. Coron, et al.: "Universal padding schemes for RSA". Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 2002.

[i.6] J. Van De Graaf: "Towards a formal definition of security for quantum protocols". Université de Montréal, 1997.

[i.7] D. Unruh: "Quantum proofs of knowledge". Annual international conference on the theory and applications of cryptographic techniques. Springer, Berlin, Heidelberg, 2012.

[i.8] J. Watrous: "Zero-knowledge against quantum attacks". Proceedings of the 38th annual ACM symposium on Theory of Computing. 2006.

[i.9] J. Don, et al.: "Security of the Fiat-Shamir transformation in the quantum random-oracle model". Advances in Cryptology-CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II 39. Springer International Publishing, 2019.

[i.10] V. Lyubashevsky et al.: "Crystals-dilithium". Algorithm Specifications and Supporting Documentation (2020).

[i.11] R. El Bansarkhani and A. El Kaafaran: "Post-quantum attribute-based signatures from lattice assumptions." Cryptology ePrint Archive (2016).

[i.12] D. Pointcheval, and J. Stern: "Security arguments for digital signatures and blind signatures". Journal of cryptology 13 (2000): pp. 361-396.

[i.13] C. Schnorr: "Efficient signature generation by smart cards". Journal of cryptology 4 (1991): pp. 161-174.

[i.14] Q .Liu and M. Zhandry: "Revisiting post-quantum fiat-shamir". Advances in Cryptology-CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II 39. Springer International Publishing, 2019.

[i.15] M. Barbosa et al.: "Fixing and mechanizing the security proof of fiat-shamir with aborts and dilithium". Cryptology ePrint Archive (2023).

[i.16] E. Fujisaki and T. Okamoto: "Secure integration of asymmetric and symmetric encryption schemes". In CRYPTO'99, volume 1666 of LNCS, pp. 537-554. Springer, Heidelberg, August 1999.

[i.17] E. Fujisaki and T. Okamoto: "Secure integration of asymmetric and symmetric encryption schemes". Journal of Cryptology, 26(1): pp. 80-101, January 2013.

[i.18] D. Hofheinz, et al: "A modular analysis of the Fujisaki-Okamoto transformation". In TCC 2017, Part I, volume 10677 of LNCS, pp. 341-371. Springer, Heidelberg, November 2017.

[i.19] J. Duman et al.: "Faster Kyber and Saber via a generic Fujisaki-Okamoto transform for multi-user security in the Q-ROM". 2021.

[i.20] J. Bos et al.: "CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM". 2018 IEEE European Symposium on Security and Privacy (EuroS P), pp. 353-367.

[i.21] M. Bellare et al.: "Public-key encryption in a multi-user setting: Security proofs and improvements". In EUROCRYPT 2000, volume 1807 of LNCS, pp. 259-274. Springer, Heidelberg, May 2000.

[i.22] A. Ambainis et al.: "Quantum attacks on classical proof systems: The hardness of quantum rewinding". In 2014 IEEE 55th Annual Symposium on Foundations of Computer Science, pages 474-483, Oct 2014.

[i.23] D. Unruh: "Computationally Binding Quantum Commitments". In Proceedings, Part II, of the 35th Annual International Conference on Advances in Cryptology. 2016 pp. 497-527.

[i.24] D. Unruh: "Collapse-binding quantum commitments without random oracles". In Advances in Cryptology-ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II 22 (pp. 166-195). Springer Berlin Heidelberg.

[i.25] J. Czajkowski, et al.: "Post-quantum security of the sponge construction". In International Conference on Post-Quantum Cryptography (pp. 185-204). Cham: Springer International Publishing.

[i.26] S. Fehr: "Classical proofs for the quantum collapsing property of classical hash functions". In Theory of Cryptography: 16th International Conference, TCC 2018, Panaji, India, November 11-14, 2018, Proceedings, Part II 16 (pp. 315-338). Springer International Publishing.

[i.27] M. Zhandry: "New constructions of collapsing hashes". In Annual International Cryptology Conference (pp. 596-624). Cham: Springer Nature Switzerland.

[i.28] J. Czajkowski: "Quantum Indifferentiability of SHA-3". In IACR Cryptol. ePrint Arch. 2021.

[i.29] T. Saito et al.: "Tightly-secure key-encapsulation mechanism in the quantum random oracle model". IACR Cryptology ePrint Archive report 2017/1005. 2017.

[i.30] N. Bindel et al.: "Tighter proofs of CCA security in the quantum random oracle model". In TCC 2019, Part II, volume 11892 of LNCS, pp. 61-90. Springer, Heidelberg, December 2019.

[i.31] J. Håstad: "Solving simultaneous modular equations of low degree". SIAM J. Comput., 17(2): pp. 336-341, 1988.

[i.32] R. Cramer and V. Shoup: "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack". SIAM Journal on Computing 33(1), pp. 167-226. 2003.

[i.33] K. Hövelmanns et al.: "Generic authenticated key exchange in the quantum random oracle model". In PKC 2020. LNCS, vol. 12111, pp. 389-422. Springer, Cham (2020).

[i.34] U. Maurer et al.: "Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology". In Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings, volume 2951 of Lecture Notes in Computer Science, pp. 21-39. Springer, 2004.

[i.35] T. Carstens et al.: "On quantum indifferentiability". IACR Cryptol. ePrint Arch., 2018: pp. 257, 2018.

[i.36] T. Ristenpart et al.: "Careful with composition: Limitations of the indifferentiability framework". In Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings, volume 6632 of Lecture Notes in Computer Science, pp. 487-506. Springer, 2011.

[i.37] J. Coron et al.: "Merkle-Damgård Revisited: How to Construct a Hash Function". In Advances in Cryptology CRYPTO 2005, volume 3621, pp. 430-448. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. Series Title: Lecture Notes in Computer Science.

[i.38] M. Zhandry: "How to construct quantum random functions". In 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012, pp. 679-687. IEEE Computer Society, 2012.

[i.39] O. Goldreich et al.: "How to construct random functions (extended abstract) ". In 25th Annual Symposium on Foundations of Computer Science, West Palm Beach, Florida, USA, 24-26 October 1984, pp. 464-479. IEEE Computer Society, 1984.

[i.40] H. Kuwakado and M. Morii: "Quantum distinguisher between the 3-round feistel cipher and the random permutation". In IEEE International Symposium on Information Theory, ISIT 2010, June 13-18, 2010, Austin, Texas, USA, Proceedings, pp. 2682-2685. IEEE, 2010.

[i.41] H. Kuwakado and M. Morii: "Security on the quantum-type even-mansour cipher". In Proceedings of the International Symposium on Information Theory and its Applications, ISITA 2012, Honolulu, HI, USA, October 28-31, 2012, pp. 312-316. IEEE, 2012.

[i.42] D. Boneh and M. Zhandry: "Quantum-secure message authentication codes". In Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings, volume 7881 of Lecture Notes in Computer Science, pp. 592-608. Springer, 2013.

[i.43] M. Kaplan et al.: "Breaking symmetric cryptosystems using quantum period finding". In Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II, volume 9815 of Lecture Notes in Computer Science, pp. 207-237. Springer, 2016.

[i.44] NIST: "Post-Quantum Cryptography Standardization - Post-Quantum Cryptography". Csrc.nist.gov. 3 January 2017. Retrieved 24 November 2023.

[i.45] E. Fujisaki et al.: "RSA-OAEP is secure under the RSA assumption". J. Cryptology, 17(2): pp. 81-104, 2004.

[i.46] E. Ebrahimi: "Post-quantum Security of Plain OAEP Transform". In Public-Key Cryptography - PKC 2022. PKC 2022. Lecture Notes in Computer Science, vol 13177. Springer, Cham.

[i.47] C. Peikert: "Lattice cryptography for the Internet". Cryptology ePrint Archive, Report 2014/070, 2014.

[i.48] J. Coron et al.: "GEM: A generic chosen-ciphertext secure encryption method". In CT-RSA 2002, volume 2271 of LNCS, pp. 263-276. Springer, Heidelberg, February 2002.

[i.49] M. Bellare and P. Rogaway: "Optimal Asymmetric Encryption -- How to encrypt with RSA (Extended abstract) ". In Advances in Cryptology - Eurocrypt '94 Proceedings, Lecture Notes in Computer Science Vol. 950, A. De Santis ed, Springer-Verlag, 1995.

[i.50] J. Czajkowskiet al.: "Quantum Indifferentiability of SHA-3". IACR Cryptol. ePrint Arch., 192.

[i.51] D. Unruh: "Universally Composable Quantum Multi-party Computation". 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 (pp. 486-505). Springer.

[i.52] A. Fiat and A. Shamir: "How to prove yourself: Practical solutions to identification and signature problems". In Advances in Cryptology - CRYPTO' 86, pp. 186-194.

[i.53] R. Canetti: "Universally composable security: A new paradigm for cryptographic protocols". In Proc. 42nd IEEE Symp. Foundations of Computer Science, pp. 136-145, 2001.

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

asymmetric cryptography: cryptographic system that utilizes a pair of keys, a private key known only to one entity, and a public key which can be openly distributed without loss of security

cryptographic hash function: function that maps a bit string of arbitrary length to a fixed length bit string (*message digest* or *digest* for short*)* with specific mathematical properties

NOTE: See clause 5.3 for details.

cryptographic key: binary string used as a secret by a cryptographic algorithm

EXAMPLE: AES-256 requires a random 256-bit string as a secret key.

cryptographic protocol: system of rules that allows two or more communicating entities to reach a security-related goal using cryptographic algorithms

entity: person, device or system

key encapsulation mechanism: method to secure the establishment of a cryptographic key for transmission using public key cryptography

message digest/digest: fixed-length output of a cryptographic hash function over a variable length input

private key: key in an asymmetric cryptographic scheme that is kept secret

public key: key in an asymmetric cryptographic scheme that can be made public without loss of security

public-key cryptography: See asymmetric cryptography.

quantum-safe: resistant to quantum attacks

random oracle: theoretical black box that responds to every unique query with a uniformly random selection from the set of possible responses, with repeated queries receiving the same response

security level: measure of the strength of a cryptographic algorithm

NOTE: If 2^{λ} operations are required to break the cryptographic algorithm/scheme/method, then the security level is λ. Sometimes also referred to as *bit-strength.*

3.2 Symbols

For the purposes of the present document, the following symbols apply:

$A \gg B$	Informal notation to denote that a quantity A is much larger than another quantity B.
$A \approx B$	Informal notation to denote that a quantity A is approximately as large as another quantity B.
$f = \mathcal{O}(g)$	Given two function $f(x)$ and $g(x)$, taking as input non-negative integers, there exists a positive constant c and a positive number x' such that $f(x) \leq g(x)$ for all $x \geq x'$.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

CBC	Cipher Block Chaining
CRQC	Cryptographically relevant quantum computer
DEM	Data-encapsulation mechanism
EUF-CMA	Existential UnForgeability under Chosen Message Attack
FO	Fujisaki-Okamoto
GMAC	Galois Message Authentication Code
IND-CCA	INDistinguishability under Chosen Ciphertext Attack
IND-CPA	INDistinguishability under Chosen Plaintext Attack
ITM	Interactive Turing Machine
KEM	Key-encapsulation mechanism
MAC	Message Authentication Code
MPC	Multi-Party Computation
OAEP	Optimal Asymmetric Encryption Padding
OW-CPA	One-Way under Chosen Plaintext Attack
OW-PCA	One-Way under Plaintext Checking Attacks
PKE	Public Key Encryption
PMAC	Parallelizable Message Authentication Code
PRF	Pseudo-Random Function
QPRF	Quantum Pseudo-Random Function
Q-ROM	Quantum random-oracle model
ROM	Random-oracle model
RSA	Rivest, Shamir, Adleman
UC	Universal Composability
ZKPoK	Zero-Knowledge Proof of Knowledge

4 Cryptographic Security Proofs and Quantum Attackers

Reasoning about the security of a cryptographic primitive is not a trivial task. A naive way to design a cryptographic system would be to go through the following steps: first, create a functional design for a system and deploy it. Then wait until someone finds an attack that breaks the system; at this point, change the system to prevent said attack or change the recommended parameters and wait for a new attack. If no new attack is published in reasonable time, one might assume that the cryptographic scheme is secure. However, it is unclear what a "reasonable time" should be: historically, there are for instance cryptographic systems that were broken after five years of silence, such as those used in the PKCS#1 family of standards [i.1]. More precisely, PKCS#1 version 1.5 [i.1] contained a padding protocol for RSA that was standardized in 1993, but it was not until 1998, that a chosen-ciphertext attack was found by Bleichenbacher [i.2]. Therefore, this approach is risk-prone and not satisfactory.

To make more meaningful statements about the security of a scheme, a definition of security needs to be in place. Such a definition should specify how an attacker is modelled and what the objective of the attacker is. The general aim is to show that an hypothetical attacker that can break the system can also solve some well-studied computational problem with a comparable effort. Under the assumption that such a computational problem is intractable (due to years of study and scrutiny of its hardness), one can therefore rule out the existence of such an hypothetical attacker. In other words, the security of the system is reduced to the hardness of a mathematical problem. More formally, such a reduction is proved as follows:

- Assume there exists a probabilistic polynomial-time algorithm A (modelling a hypothetical attacker) that can compromise a certain security goal of the scheme in time T, given certain powers. Here T is polynomial in λ, the desired security level of the scheme (typically chosen to be equal to 128 or 256).

- Create an algorithm $B = B(A)$ (possibly probabilistic) that, given A, can solve the mathematical problem in time $f(T)$ for some function f.

This is called a *security reduction*. If the powers of an attacker are correctly modelled, such a reduction implies that the attacker has to attack either the implementation of the scheme or the underlying mathematical assumption to compromise the security goal. Ideally, $f(T) \approx T$, in which case it is shown that breaking the cryptographic system takes approximately as much time as solving the mathematical problem. By contrast, if $f(T) \gg T$, then breaking the cryptographic system might be significantly easier than solving the mathematical problem. How close $f(T)$ is to T is referred to as the *tightness* of the reduction. Basing the security of a cryptographic scheme on a non-tight reduction, e.g. $f(T) = T^2$, might result in overly conservative parameter choices and impractical cryptographic protocol instantiations. However, these reductions do show that there is no structural weakness in the cryptographic system.

This is the reason that modern cryptographic systems generally base their security on the assumed hardness of some computational problem. When considering attackers that can use a CRQC, a general misconception is that simply swapping the computational problems underlying a cryptographic system for "quantum-hard" computational problems (i.e. problems that are considered intractable even for a quantum computer) is enough to make the system quantum-safe. Unfortunately, this is not always true. As discussed above, all mathematical proofs of security have to specify the powers of the attacker. A quantum attacker has properties that are not modelled in classical proofs. Therefore, in addition to using quantum-hard computational problems, the proofs themselves often need to be modified as well.

Modifying a security proof in such a way that it accommodates for quantum attackers is often not trivial, and sometimes no such proof is available. While this does not directly mean that the corresponding scheme is broken (since it might be the case that such a proof exists, but cryptographers have been unable to find it yet), it does raise concerns on their quantum-security. Moreover, even when a "quantum" proof is available, this often has an impact on the security level that is attained against quantum adversaries; the reduction might be less tight than their classical counterparts.

Finally, the formulation itself of the security goals of cryptographic schemes is a delicate task, and given the often counter-intuitive properties of quantum computing, some formulations turn out to be insufficient to achieve the desired level of security. In this case, new formulations and associated proofs are needed.

5 Mathematical preliminaries

5.1 Indistinguishability

5.1.0 Introduction

The notion of indistinguishability is often used to argue about the security of encryption schemes. The general idea behind such arguments is that an attacker with certain powers cannot tell the difference between encryptions of two different messages. Since the attacker cannot distinguish, they do not learn anything about the message. Within the domain of indistinguishability for encryption, there are three different ways to model the adversary's powers. These different models correspond to specific real-life situations. The three attacker models are chosen plaintext attack, non-adaptive chosen ciphertext attack and adaptive chosen ciphertext attack. The strongest guarantees are obtained when an encryption scheme is indistinguishable under adaptive chosen ciphertext attacks (IND-CCA2). IND-CCA2 security is considered to be the norm for general-purpose encryption schemes to be deployed in practice. It is common to abbreviate IND-CCA2 to IND-CCA. The following subsections provide more details on the attacker models.

5.1.1 Chosen Plaintext Attack (CPA)

If an encryption scheme attains indistinguishability under Chosen Plaintext Attacks (IND-CPA), then an adversary is not able to obtain any information about messages that are freshly encrypted, even if they can encrypt messages of their own choice.

5.1.2 Non-Adaptive Chosen Ciphertext Attack (CCA1)

If an encryption scheme attains indistinguishability under non-adaptive Chosen Ciphertext Attacks (IND-CCA1), then an adversary is not able to obtain any information about messages that are freshly encrypted even if they have access to encryptions (ciphertexts) of messages of their own choice and even if they get access to decryptions of ciphertexts of their own choice. It is assumed, however, that the decryptions of ciphertexts should be executed before the selection of the messages to be freshly encrypted. This security notion was believed to be sufficient for security against real-world threats. However, Bleichenbacher published a practical attack against an IND-CCA1-secure version of RSA in 1998 [i.2], proving this security notion inadequate in practice.

5.1.3 Adaptive Chosen Ciphertext Attack (CCA2)

If an encryption scheme attains indistinguishability under adaptive Chosen Ciphertext Attack (IND-CCA2), then an adversary is not able to obtain any information about messages that are freshly encrypted even if they have access to encryptions (ciphertexts) of messages of their own choice and can get access to decryptions of ciphertexts of their own choice. The subtlety here is that the adversary is allowed to obtain decryptions of ciphertexts even after the message to be encrypted has been selected (and encrypted), although it is not allowed to ask for a decryption of this target ciphertext. This security notion is now believed to be sufficient for security against real-world threats. Specifically, a newer IND-CCA2-secure version of RSA is resistant to Bleichenbacher's attack.

5.2 Qubits

The main difference between classical computers and quantum computers is that quantum computers operate on qubits, whereas classical computer operate on classical bits. Loosely speaking, while a classical bit can assume a value of either 0 or 1, a qubit can be in a state which is a combination of 0 and 1. This is called superposition. When a qubit is measured (in the so-called computational basis), its superposition collapses to a classical 0 or 1, according to a probability distribution associated with the superposition.

Qubits can be modelled mathematically independently of their physical implementation. The state of a qubit can be denoted using a vector $\binom{\alpha}{\beta}$ of length two with complex coefficients α, βsuch that $|\alpha|^2 + |\beta|^2 = 1$. The "zero" state is interpreted as $\binom{0}{1}$ and the "one" state is interpreted as $\binom{1}{0}$. A qubit $\binom{\alpha}{\beta}$ is therefore a superposition of α times "zero" plus β times "1". The coefficients of this linear combination relate to the probability of a 0 or 1 after the superposition collapses.

A more common and convenient notation for qubits is the so-called bra-ket notation. The zero state is represented by $|0\rangle$ and the one state is represented by $|1\rangle$. A qubit $\binom{\alpha}{\beta}$ can therefore be expressed as $\alpha|0\rangle + \beta|1\rangle$. Most quantum algorithms require more than two qubits. The shorthand notation of n qubits in the zero state is $|00 \ldots 0\rangle$. A superposition of n qubits is written as:

$$\textstyle\sum_{x\in\{0,1\}^n} a_x\, |x\rangle$$

for complex coefficients a_x (commonly referred to as *amplitudes*) such that $\sum_{x\in\{0,1\}^n} |a_x|^2 = 1$. A collection of several qubits is often referred to as a quantum *register*.

An important theory within the field of quantum information is the no-cloning theorem, which states that it is impossible to make a perfect independent copy of an arbitrary unknown quantum state. In more formal terms, if given an arbitrary qubit $|\phi\rangle$ and a fixed qubit, say, $|0\rangle$, it is impossible to make a copy of $|\phi\rangle$ and "store" it in the qubit $|0\rangle$. In mathematical terms, there exists no quantum operator U such that $U(|\phi\rangle|0\rangle) = |\phi\rangle|\phi\rangle$ for every qubit $|\phi\rangle$.

5.3 Cryptographic Hash Functions

A cryptographic hash function is a one-way function that maps an arbitrarily long bitstring - also referred to as a message - to a fixed-size bitstring called a hash digest. Additionally, cryptographic hash functions have three important properties:

1) Pre-image resistance: Given a hash digest, it is computationally infeasible to find a message that maps to it.

2) Second pre-image resistance: Given a message and its hash digest, it is computationally infeasible to find another message that maps to the same hash digest.

3) Collision resistance: It is computationally infeasible to find two different messages that map to the same hash digest.

5.4 Proofs of Knowledge

5.4.0 Introduction

An interactive proof-of-knowledge protocol involves two entities: a prover P and a verifier V. The goal of the protocol is to convince the verifier V that the prover P knows a certain secret with certain properties. In a trivial proof-of-knowledge protocol, the prover would simply send the secret to the verifier, proving that they know the secret. The verifier can then verify that this secret does indeed enjoy the claimed properties. More advanced proof-of-knowledge protocols can prove the same properties without revealing any information about the secret. Proof-of-knowledge protocols are generally randomized; a proof of knowledge protocol is said to be *public-coin* if the verifier makes their choices of randomness publicly known.

Generally, proof-of-knowledge protocols with additional properties are of particular interest. A Zero-Knowledge interactive Proof-of-Knowledge protocol (ZKPoK) is a proof-of-knowledge protocol where the verifier is convinced that the prover knows a certain secret with specific properties, without learning any new information about that secret. The secret is often referred to as the *witness*. A concrete example is a ZKPoK where the prover wants to prove that they know the plaintext to a certain ciphertext without revealing anything about this plaintext. The ciphertext is then considered public information and the witness is the plaintext (and the used randomness if the encryption scheme is non-deterministic). Notice that ZKPoKs are therefore a sub-variant of so-called "zero-knowledge proofs", where the prover wishes to convince the verifier that there *exists* a witness for a given public statement, without necessarily claiming to know the witness. In the previous example with encryption, the prover would therefore want to prove that a given string is a valid ciphertext, but with no further claim on their knowledge of the underlying plaintext value.

Typically, it is required that for a ZKPoK system at least three properties hold with overwhelming probability: correctness / completeness, soundness, and zero-knowledge.

5.4.1 Correctness or Completeness

The terms correctness and completeness are used interchangeably. A ZKPoK system attains correctness or completeness. The property is as follows: if the prover does indeed know a secret with the claimed property, and if both prover and verifier follow the instructions of the protocol, then the verifier will accept the prover's claim of knowledge of the secret.

5.4.2 Soundness

If the prover does *not* know a secret with the claimed property, then the verifier will reject the prover's claim of knowledge of the secret with overwhelming probability.

5.4.3 Zero-Knowledge

The verifier gains no new information on the secret as a result of the protocol.

5.4.4 Sigma Protocols

The most common subclass of interactive ZKPoK protocols is the class of Sigma protocols. Protocols in this family adhere to the following three-step approach:

1) The prover computes a value a, based on some randomness r, and possibly on their (supposed) knowledge of a witness w for the public statement s. The prover then sends a to the verifier.

2) The verifier, upon receiving a, and possibly based on s, computes a *"challenge"* value c and sends it back to the prover.

3) The prover computes a value z based on s, w, r and c, and sends it to the verifier.

The verifier, based on all elements that they have received and computed, then either "accepts" (that the prover holds a witness) or "rejects".

6 The Rewinding Technique for Zero-Knowledge Proofs

Several cryptographic security proofs are no longer valid in a quantum setting, due to the difficulty of transposing a mathematical technique known as rewinding. This technique is often used to prove the soundness property of ZKPoKs (see clause 5.4). This clause describes the issue and the potential impact of the invalidity of the rewinding technique on security of protocols that are in use today. Notice that the issue also affects general zero-knowledge proofs, and not only proofs *of knowledge*, but that it has a more marked impact for this specific variant, as discussed later in this clause. For this reason, this clause focuses on ZKPoKs.

The rewinding technique is used in ZKPoKs to prove both soundness and zero knowledge; the problem is illustrated by focusing on the first property. Soundness is generally formalized in the following terms: if a prover can convince the verifier that they know w with non-negligible probability, then the value of w can be extracted using oracle access to the prover, where "oracle access" means the internal states and variables of the prover remain inaccessible. This means that the steps a prover would undergo to convince the verifier that they know w can be used to obtain w, and thus knowledge of w and ability to prove it are equivalent.

NOTE: It may seem like extracting w would violate the zero-knowledge property. However, the subtlety lies in the fact that the oracle access to the prover P is only available to P. A verifier does not have oracle access to the prover and can therefore not extract the witness w.

Such a statement is then proved by producing an *extractor* that, given the public statement and oracle access to the prover, outputs a valid witness w to the statement. Such an extractor would typically feed the prover some input values (e.g. a challenge in the case of Sigma protocols), record the produced output, and then "rewind" the prover to a previous snapshot and feed it some other input values, recording the produced output, and so on.

This proof technique is perfectly valid in the classical model. If a *quantum proof of knowledge* is needed, however, then the snapshots of the prover state should be quantum-accessible, but that is impossible as shown in [i.6] for two reasons:

1) The *no-cloning theorem* states that unknown quantum information cannot be copied. "Taking a snapshot" of a certain state of the prover would mean copying that state and saving it for later access, but this theorem implies that this operation is not possible. This means that the extraction algorithm cannot be adapted to work in the quantum setting.

2) When a quantum state is measured, it collapses to a classical state, which destroys information. Such a measurement would typically be performed by the extractor, and this process would therefore destroy information that could be necessary later on.

On the other hand, Unruh [i.7] showed that classical proofs of knowledge (for sigma protocols) can be quantum proofs of knowledge, if the protocol satisfies the relatively standard variant property of *special soundness* (which states that any prover that can produce two valid "replies" z_1 and z_2 corresponding to two different challenges $c_1 \neq c_2$, but with same first message a, is able to efficiently compute w), and if it additionally satisfies a new property called *strict soundness*. Informally stated, the strict soundness property says that for a given initial value a and challenge c as described above, the response value z is uniquely defined. This essentially ensures that z itself does not contain a lot of information, so measuring z does not disturb the quantum state too much. In turn, this makes it possible to apply a quantum rewinding technique.

For the zero-knowledge property, the issue is somewhat similar, in that a popular strategy to prove it requires building a simulator that, given black-box access to a verifier and no connection whatsoever to a valid prover, can produce an output which is indistinguishable from the output produced by the verifier in a normal prover-verifier interaction. Once again, these proofs typically rewind the verifier, which is not possible due to the no-cloning and destructive-measurement properties of quantum information, as discussed above.

The issue of zero-knowledge (not *proof of knowledge*) has been addressed by Watrous [i.8] by introducing a quantum rewinding technique. Building on this result and on strict soundness, Unruh [i.7] then shows that it is possible to create a quantum-computationally zero-knowledge quantum proof of knowledge, using the NP-complete problem of Hamiltonian cycles, under the assumption that quantum 1-1 one-way functions exist. In the same work, Unruh makes two proposals based on hash functions and block ciphers respectively. If these are quantum pseudo-random functions, the construction is a quantum 1-1 one-way function. Therefore, if they can be proven to be quantum pseudo-random functions, then the former construction is a quantum-computationally zero-knowledge quantum proof of knowledge. Since the problem of Hamiltonian cycles is NP-complete and any NP-relation can be reduced to the Hamiltonian cycle problem, the proposed protocol can be extended to prove any relation in NP.

7 Security Proofs in The Quantum Random Oracle Model

7.1 The Quantum Random Oracle Model

A family of cryptographic proofs models cryptographic hash functions as mathematical constructions known as random oracles. Proofs of this type are "valid in the Random Oracle Model" or "valid in the ROM", and are somewhat controversial [i.3] for the following reasons:

- On one hand, a random oracle is *not* an accurate representation of a hash function. In fact, there exist artificial protocols which are proven to be secure in the ROM, but which are provably *in*secure when a hash function is used instead of the ROM, regardless of the hash function used [i.4].
- On the other hand, proofs in the ROM work very well in practice. Counterexamples such as the one referenced above remain artificial and of no impact on concrete protocols. Additionally, proofs in the ROM typically have a quite tight reduction, which means that they result in smaller parameters (assuming the parameters are chosen based on the security reduction) and, therefore, better efficiency, when compared to proofs for the same security level that do not make use of the ROM.

Despite the downside expressed in the first point above, the ROM has been widely successful, and many protocols of daily use (e.g. some RSA implementations [i.5]) rely on it. The ROM framework provides a lot of control over the random oracle resulting in many convenient proof techniques to be used. Specifically, in proving reductions, it is possible to record the queries an adversary makes to the random oracle and it is possible to reprogram the random oracle by assigning specific digests to specific inputs, as long as it seems random to the adversary.

A security proof should model a real situation as accurately as possible. Hence, when considering a quantum-capable adversary, one should assume that it has quantum access to the building blocks in its possession. In particular, if the cryptographic protocol under scrutiny uses a hash function H, then it should be assumed that the adversary can query the hash function in superposition, i.e. that they can obtain the quantum state $\sum|x\rangle|H(x)\rangle$ for any superposition $\sum|x\rangle$ of input values in a single evaluation of the hash function.

Translated to the ROM, this means that such an attacker should be able to query the oracle once with a superposition $\sum|x\rangle$ of input values and obtain $\sum|x\rangle|H(x)\rangle$ (H here being the random oracle). Such a setting is called Quantum-accessible Random Oracle Model, or simply Quantum Random Oracle Model (Q-ROM).

The problem introduced by quantum computing is that security proofs that hold in the ROM do not automatically hold in the Q-ROM. For example, it is impossible to record or copy the queries an adversary makes without disturbing the superposition. This also makes it impossible to reprogram the random oracle, as that requires knowledge of which queries the adversary makes. This means that proofs using these techniques will need to be adapted to hold in the Q-ROM.

Adapting a proof in the ROM to a proof in the Q-ROM is not always possible. In fact, there are protocols that are provably secure in the ROM but provably *in*secure in the Q-ROM. However, much like the theory-vs-practice dichotomy of the ROM, such counterexamples are completely artificial, and it is often stated [i.9] that a concrete protocol that is secure in the ROM will remain secure in the Q-ROM (if based on quantum-safe hardness assumptions), although with no formal proof in this sense.

The following subclauses detail some typical scenarios where the Q-ROM plays a role, and what the impact of these considerations is.

7.2 The Fiat-Shamir Transformation

The *Fiat-Shamir transformation* turns a Sigma protocol (see clause 5.4.4) into a non-interactive ZKPoK. In its essence, the prover computes the first element a of the Sigma-protocol execution, and instead of waiting for the verifier to reply with a challenge c, he computes c as $c \coloneqq H(s, a)$, where s is the public statement and H denotes a cryptographic hash functions (see clause 5.3), and then proceeds to compute the resulting element z. Therefore, the prover can use this transformation to create a transcript $(a, H(s, a), z)$ that proves their knowledge of a certain witness w for s, *without* interacting with the verifier. In turn, the verifier only needs to be presented with a transcript to verify the prover's claim, with no further interaction needed.

The security of the PoK obtained in this way can be proven in a black-box fashion in the ROM (i.e. by modelling the hash function H as a random oracle) under the assumption that the original interactive PoK proof is secure. The Fiat-Shamir transformation is notably used to construct digital-signature schemes from Sigma protocols [i.10], [i.11], [i.12] and [i.13].

It should be noted that the issue is often seen as of theoretical nature within the cryptographic community: as pointed out in the previous clause, it is often expected that a protocol which is secure in the ROM (and is based on appropriate computational-hardness assumptions) will remain secure against quantum attackers. In particular, post-quantum standard candidates in the NIST competition were *not* required to have a valid security proof in the ROM at the time of submission, and the parameters of these candidates are *not* based on the security reduction, but rather on the best well-known attacks (which do not target the reduction itself).

The security of the Fiat-Shamir transformation in the Q-ROM has been the subject of various research articles [i.7], [i.8], [i.9] and [i.14]. The main issue in carrying the proof over from the ROM to the QROM is as follows. In the ROM security proof of the Fiat-Shamir transformation, the reprogrammability of the random oracle is crucial. Specifically, a dishonest non-interactive prover that can fool a verifier with probability p can be used to construct a dishonest interactive prover that can fool a verifier with similar probability. For a dishonest non-interactive prover for security statement s that produces commitment a, and response z (constructed using a challenge $c=H(a,s)$), the following dishonest interactive prover P can be constructed: P sends a to the verifier and receives a challenge c'. P reprograms the oracle such that $H(a,s)=c'$. Now z is a valid response. This is a tight reduction classically [i.52]. The necessity of reprogramming led to the belief that the Fiat-Shamir transformation is not secure in the QROM [i.22]. However, the works of [i.9] and [i.14] show a quantum reprogrammability technique that can be applied if the underlying Sigma protocol has specific properties, referred to as either *quantum computationally unique responses* [i.9] or *collapsing sigma protocol* [i.7]. This property holds if it is computationally infeasible for a quantum adversary to determine whether a superposition of valid responses for a given (a, c) pair has been measured or not.If the Sigma protocol has this property, then its Fiat-Shamir transformation is a quantum proof-of-knowledge as well, although with a less tight reduction than in the classical setting.

In terms of the tightness of the obtained reduction, in the Q-ROM case, given q queries to the random oracle, the probability that an adversary can produce a valid transcript without knowledge of a witness is a factor $\mathcal{O}(q^2)$ larger than the probability that an adversary can break the non-interactive Sigma protocol. This means that the tightness of the proof changes from a factor q to a factor $\mathcal{O}(q^2)$ in the Q-ROM. The practical consequence is that Q-ROM Fiat-Shamir reductions result in larger parameters than ROM Fiat-Shamir reductions, e.g. a larger challenge set used in the interactive protocol or a higher number of parallel repetitions of the interactive protocol - although once again, in practice the parameters of these constructions are often based on the best well-known attacks and not on the tightness of the reduction.

Dilithium (the only NIST post-quantum draft standard that uses the Fiat-Shamir transformation) is proven secure in the Q-ROM [i.15].

7.3 The Fujisaki-Okamoto Transformation and Related Constructions

7.3.0 History of Transformations

IND-CCA2 is the standard for encryption schemes these days [i.44]. For secure messaging, it is required that long messages can be sent securely, without any information about the message leaking to eavesdroppers. The most common approach towards building an IND-CCA2 secure messaging protocol uses the KEM-DEM paradigm [i.32], where an IND-CCA2 asymmetric encryption scheme is combined with an IND-CCA2 symmetric encryption scheme to produce an IND-CCA2 hybrid encryption scheme. It has the advantage of being public-key based, so no keys need to be pre-shared. Additionally, the efficiency of the symmetric cipher makes it a suitable approach for arbitrarily long messages.

The DEM part of KEM-DEM is often instantiated using AES or another symmetric cipher, of which the classical and quantum security are well-studied and accepted. The only other requirement to obtain a quantumly-secure hybrid encryption scheme is therefore to combine it with a quantum-secure IND-CCA2 KEM.

The first IND-CCA2 asymmetric encryption scheme was RSA-OAEP, which uses the OAEP transformation [i.49], [i.45]. OAEP was originally designed for the RSA cryptosystem, but can be applied in a broader way. It takes a deterministic partial-domain one-way trapdoor permutation and produces an IND-CCA2 encryption scheme. Many schemes that came after RSA, however, were not based on partial-domain one-way trapdoor permutations, specifically the post-quantum class of algorithms.

Over time, multiple algorithms were designed to transform asymmetric encryption schemes with different security properties into IND-CCA2 KEMs. Specifically, the Fujisaki-Okamoto transformation [i.17], [i.16], the REACT transformation [i.48] and the GEM transformation [i.48]. In [i.18], Hofheinz et al. broke down the Fujisaki-Okamoto transformation into multiple smaller transformations with possible variations. As a result, they were able to show that the REACT transformation and GEM transformation are in fact slight variations of smaller transformations that are used within the Fujisaki-Okamoto transformation. The proof techniques for the properties of the REACT and GEM transformation follow the same steps as this modular subtransform of the Fujisaki-Okamoto transform.

Because of this modularity with variations, the Fujisaki-Okamoto transformation can take a non-deterministic encryption scheme with the OW-CPA or IND-CPA property and produce an IND-CCA2 KEM of which the security is based on the same assumption as the original scheme. The classical security proof is tight when the scheme is IND-CPA and not tight when the scheme is only OW-CPA. The parameters of the final scheme will therefore need to be bigger for the same security level if the original scheme has OW-CPA compared to IND-CPA.

The pure REACT and GEM transformations require the OW-PCA (One-Way against Plaintext Checking Attacks) property, which is different from the OW-CPA property and is a lot less common for schemes to attain. Specifically, Peikert shows that many natural lattice-based encryption schemes do not have the OW-PCA property due to the equivalence of the search and decisional Learning With Errors problems [i.47]. For such schemes, the full Fujisaki-Okamoto transform is more suitable.

7.3.1 The Original Fujisaki-Okamoto Transform

The Fujisaki-Okamoto transform was first introduced in 1999 [i.16] and later improved in 2013 [i.17]. Generally, only the latter is considered, because the proof is tighter and certain issues with the previous version were fixed in the later version.

In the classical proof, the success probability of an adversary trying to break the IND-CCA2 property of the final scheme is roughly a factor q larger than the probability of an adversary trying to break the IND-CPA property of the underlying scheme, where q is the number of random-oracle queries the adversary is allowed to make, which is considered a tight reduction, since it is practically impossible to get a smaller tightness gap in the (Q)ROM. There are two problems when this transformation is used for post-quantum KEMs:

1) This proof requires the IND-CPA PKE to be perfectly correct, while almost all post-quantum candidates have a small correctness error. This is specifically the case for NIST finalist Crystals-Kyber [i.20].

2) This proof technique uses the ROM, so there is no guarantee that the proof or its tightness gap hold against quantum adversaries.

3) This proof holds in the single-user setting.

NOTE: Problems 1 and 3 also apply to classical KEMs.

7.3.2 Solving The Correctness Problem

Unfortunately, the proof techniques to prove the IND-CCA2 security of the FO transform in the ROM cannot easily be translated into proof techniques for the Q-ROM. This is due to some issues that arise with quantum computation. Specifically, a common proof strategy in the ROM for FO transforms is to show that, if the evaluation of a random oracle on a given input can be distinguished from a uniformly random value, then the adversary has to have queried the oracle on that input already. The adversary therefore knows the input, and it is argued that this implies that the adversary has broken the security of the asymmetric encryption scheme that is used in the FO transform. Since quantum adversaries can access the random oracle in superposition, it is not immediately clear how this proof technique would work in the Q-ROM.

The work of Hövelmans et al. [i.18] introduces a variety of FO transformations that are secure in the ROM and allow the IND-CPA PKE to have a small decryption errors. Additionally, the security reductions to an IND-CCA2 KEM are tight. The transformation that is generally used in follow-up work is $FO_m^{\perp}$, and goes as follows.

Assume an asymmetric IND-CPA encryption scheme PKE with key generation function GEN, encryption function ENC and decryption function DEC. GEN takes a security level and produces a public key and a secret key, ENC takes a message, a public key and randomness and produces a ciphertext and DEC takes a ciphertext and a secret key and produces a decryption in the message space or failure ($\perp$).

For key generation, given security level λ:

1) Calculate $(pk', sk') = GEN(\lambda)$

2) Sample a uniformly random λ-bit string s

3) Return (sk', s)

For encapsulation, given public key pk:

1) Sample a uniformly random plaintext p from the message space of PKE

2) Calculate ciphertext $c = ENC\big(p, pk, G(p)\big)$ for cryptographic hash function G

3) Return c and $H(p)$ for cryptographic hash function H different from G

For decapsulation, given secret key sk, public key pk and message $m = (c', p')$:

1) parse $(sk', s) = sk$

2) $p'' = DEC(c', sk)$

3) If $p'' = \perp$ or $c' \neq ENC\big(p'', pk, G(p'')\big)$, return $H(s \vee c)$

4) Else, return $H(p'')$

NOTE: The original transformation in [i.18] also added c to the input of H in step 4 of decapsulation, but in [i.30] it is shown that this is not necessary.

7.3.3 Solving the Q-ROM Problem

This leaves one issue, namely that of a proof in the Q-ROM. The original work of [i.18] only proved IND-CCA2 security of the above construction in the ROM and introduced an extended version to prove IND-CCA2 security in the Q-ROM. The later work of [i.33] proves the above construction to be an IND-CCA2 KEM in the Q-ROM with a quadratic tightness gap in the number of queries.

Unfortunately, the quantum security proof only applies to specific variations of the Fujisaki-Okamoto transform [i.18]. The REACT and GEM transformations [i.48] do not fall under this specific result, but were proven to be quantumly secure with linear tightness gap if the encryption scheme has the OW-qPVCA property, which stands for quantum plaintext checking attacks, where the adversary can make quantum queries to a plaintext checking oracle [i.47].

With respect to OAEP, the quantum security of the transform was proven if the permutation is a deterministic quantum partial-domain one-way permutation [i.46].

7.3.4 Solving the User Setting Problem

The construction in clause 7.3.2 is specifically secure in a single user setting, where one user publishes its public key so that other users can send them encrypted messages. However, single user security is not enough when it is possible that a user sends the same message to multiple different users with each a public/private key pair. It is then considered to be a multi-user setting. Security for single-user IND-CCA2 secure schemes can completely break in multi-user settings, as shown by [i.31], which shows that the message can be decrypted by anyone if the basic RSA cryptosystem is used. Fortunately, [i.21] proves that IND-CCA2 security in the single user setting implies IND-CCA2 security in the multi-user setting, but the tightness gap grows linearly in the number of users.

A much smaller tightness gap is achieved by [i.19], who slightly alter the FO transform to obtain domain separation. Instead of calculating $G(p)$ in step 2 of encapsulation and $G(p'')$ in decapsulation, they compute $G(p \vee pk)$ and $G(p'' \vee pk)$. They additionally show how using a small uniformly random part of the public key can be used instead of the entire public key, without affecting the proof or its tightness.

7.3.5 Crystals-Kyber

The NIST finalist Crystal-Kyber [i.20] uses a slightly altered version of the FO transform in [i.18] that leverages the same security arguments for IND-CCA2 security and additionally applies the suggestion from [i.19] for protection against multi-user attacks, using the full public key as input to G. In the security considerations section of [i.20], the tightness of the reductions is taken into account. Their ROM reduction is tight, so basing parameters on that reduction would not require larger values. However, their Q-ROM reduction is non-tight. They argue that this non-tightness is not a problem for practical security, since:

1) the attacks that would be able to abuse this non-tightness are impractical;

2) they introduce a tight reduction to IND-CCA2 security from a non-standard assumption using a technique from [i.29]. This new reduction requires statistical disjointness of the IND-CPA version of Kyber, which they argue to be the case.

Therefore, the authors of [i.20] decided not to increase the parameters based on the non-tight Q-ROM reduction. However, the choice to ignore non-tightness has to be a calculated and thoroughly argued one, since non-tightness can be abused by adversaries, as seconded by the authors of [i.20], and non-tight Q-ROM reductions can therefore not always be ignored.

8 Commitment Schemes

With a commitment scheme, a party holding a secret message m can produce an element called *commitment* on m, such that the commitment reveals no information on m, yet being bound to it. More precisely, a commitment scheme consists of two algorithms:

1) *Commit* algorithm: on input a message m, it produces a commitment c and *opening information* u, i.e. $(c, u) = \text{commit}(m)$.
Concretely, the party holding the secret message m would run this algorithm and publish c, while holding u private, in what is known as *commit phase.*

2) *Verification* algorithm: on input a message m, commitment c, and opening information u, it outputs either 1 or 0 (to be interpreted as true or false).
Typically, in what is known as a *reveal phase,* the party holding the secret m and the opening information u obtained from the commit algorithm would publish both m and u, and other parties that already had access to the commitment c can run the verification algorithm to check that m is indeed the message associated to c.

Commitment schemes are required to be *correct,* meaning that $\text{verify}(m, u, c) = 1$ if $(c, u) = \text{commit}(m)$, and satisfy the *binding property* and the *hiding property.* The binding property says that given $(c, u) = \text{commit}(m)$, it is infeasible to find $m' \neq m, u'$ such that $\text{verify}(m', u', c) = 1$. This means that after committing to a message m, one cannot later claim to have committed to a different message. The hiding property says that it is infeasible to determine m if only given the commitment value c. This means that it is infeasible to figure out what the message m is until the owner of the secret message publishes the opening information u in the reveal phase.

The infeasibility notion of the binding and hiding property can be concretely defined in several ways, giving rise to different variants of these properties. More precisely, a commitment scheme can be information-theoretically binding, meaning that for each commitment c there exists only one pair (m, u) which $\text{verify}(m, u, c) = 1$. Alternatively, a commitment scheme can be computationally binding, meaning that there exist multiple pairs m', u' with $m' \neq m$ that are accepted for a single c (i.e. that satisfy $\text{verify}(m', u', c)$), but that they are infeasibly hard to compute.

The hiding property can be either information-theoretical, statistical, or computational. Information-theoretical hiding means that for a given commitment c, all possible values m' are equally likely to have generated c. If it is statistically hiding, all values m' are almost equally likely to have generate c, with negligible differences. If it is computationally hiding, then it is infeasibly hard to compute a message m' such that $(c, \ldots) = \text{commit}(m')$, when given only c as input.

It is easy to prove that no commitment scheme can be both information-theoretically binding and information-theoretically/statistically hiding. Intuitively, this is because if a commitment is information-theoretically binding, there is only one pair m, u such that $\text{verify}(m, u, c)$ returns 1. It is therefore theoretically possible to exhaustively try all m', u' to figure out which message was committed to. Similarly, if a commitment scheme is information-theoretically hiding, then there should be multiple m', u' such that $\text{verify}(m', u', c)$ returns 1. It is then theoretically possible to exhaustively search for other m', u' that are accepted.

Since the present document focuses on the security of classical protocols against quantum-capable attackers, it will be assumed that all values m, c and u are classical. The issue with commitment schemes is that a quantum-capable attacker can break their security in a way which is not captured by the above definitions; in other terms, these mathematical definitions do not fully express the properties that one would reasonably ask from a quantum-safe commitment scheme.

More precisely, the work of [i.22] shows that computational binding is insufficient for a commitment scheme that wants to achieve quantum security. More specifically, the authors show that there exists a commitment scheme that attains computational binding, yet there exists a quantum-polynomial-time adversary A' who can produce a commitment c, and can then find valid opening information u' for any requested message m' in the message space. This seems to be in contradiction with the binding property, but this is not the case: the adversary A' never computes two pairs $(m, u), (m', u')$ with $m \neq m'$ that are both valid openings for the commitment. The algorithm only finds one pair, but the message can be chosen after the commitment has been made. Notice that this attack has no equivalent in the classical setting, since any classical algorithm that takes as input c and m', and produces valid opening information u', could be run several times with different inputs m'', hence violating the notion of binding; this is not the case for the quantum algorithm devised by the authors of [i.22], which makes use of a quantum state that collapses after computing u', hence making it impossible to run the algorithm again.

Even though the attack is only showed to work for a specific, somewhat contrived commitment scheme, it does indicate that the notion of classical computational binding is insufficient in the presence of quantum adversaries.

One solution is to only use information-theoretically binding commitment schemes (which then have to be statistically/computationally hiding), but this generally results in inefficient schemes, having in particular long commitments. Other alternatives introduce new binding properties that can solve the aforementioned problem, but they all have certain disadvantages that are undesired for practical and efficient commitment schemes. An overview of these can be found in [i.23]. Additionally, in [i.23], Unruh introduces a new binding property called *collapse-binding*, which does not suffer from the same disadvantages as the other propositions, and in particular does not conflict with information-theoretical/statistical hiding the way that information-theoretical binding would and allows for short commitments. Unruh therefore claims that collapse-binding commitments in the quantum setting are similar to computationally binding commitments in the classical setting.

Only some intuition on the collapse-binding property will be given in the present document; the reader can refer to the original article for a more detailed and formal discussion. Given an information-theoretically binding commitment c, there exists only one message m_c for which c is a valid commitment; hence if a quantum adversary A attempts to produce a (classical) commitment c and a superposition of valid messages $|\varphi\rangle$ for c, necessarily $|\varphi\rangle = |m_c\rangle$, i.e. the quantum state $|\varphi\rangle$ is actually a trivial superposition, and will always result in the same message m_c when measured. This means that if the adversary produces c and $|\varphi\rangle$, and $|\varphi\rangle$ is subsequently measured, then this measurement will produce no effect, and the adversary will have the same behaviour whether this measurement is performed or not. With collapse-binding, this definition is relaxed: for any quantum-polynomial-time adversary A, measuring $|\varphi\rangle$ only modifies the behaviour of A with negligible probability.

In the same work, Unruh discusses how collapse-binding commitment schemes can be obtained, focusing in particular on the somewhat canonical approach of using a cryptographic hash function H (possibly modelled as a random oracle), and then setting $\text{commit}(m) \coloneqq (H(m|u), u)$ for random u, with the trivial verification function that checks whether $c = H(m|u)$ or not. While the collision resistance of H implies that the scheme obtained in this way is computationally binding, Unruh shows that it does not automatically imply collapse-binding. In order to achieve this property, the hash function needs to be *collapsing*, i.e. to satisfy a new property introduced in the same article. A collapsing hash function is defined by means of a game, where the adversary A attempts to produce a (classical) digest h and a quantum register M containing a superposition of inputs m_i such that $H(m_i) = h$; in a variant of the game, the register M is measured before being given back to the adversary, and the hash function H is said to be collapsing if no quantum-polynomial-time A can detect whether this measurement was performed or not (with non-negligible probability).

Unruh shows that random oracles satisfy the collapsing property, and conjectures that functions like SHA-3 are collapsing. Subsequent works [i.24], [i.25] proved that SHA-2 and SHA-3 are collapsing, but under some unproven assumptions on some of their components. Fehr [i.26] introduced in 2018 an alternative framework to define the notion of collapsing and collapse-binding, with much simpler proofs and arguments as a consequence. Finally, Zhandry [i.27] introduced in 2022 several different constructions of collapsing hash functions, based on different quantum-hardness assumptions.

9 Security Under Parallel Composition

9.0 Introduction

Game-based and simulation-based proofs provide security under sequential composition with the same protocol (game-based) or arbitrary protocols (simulation-based). However, this does not guarantee that certain protocols are secure when they are initiated in *parallel*. There are two frameworks that guarantee security under parallel composition: the Universal Composability (UC) framework and the Indifferentiability framework. In essence, they both provide the same guarantees, namely that if a larger process initiates another process as a sub-routine, that the behaviour of the larger process changes negligibly if the sub-process is exchanged for some ideal functionality. The two frameworks take very different approaches towards formalizing systems and sub-systems, but in principle a proof in one can also be written as a proof in the other. However, the way that processes are formalized does influence how easy it is to write a proof for a certain process. For example, it is quite natural to write a proof for a protocol between different parties in the UC framework, whereas proofs for cryptographic primitives, such as hash functions, can be written more easily in the Indifferentiability framework.

9.1 The Universal-Composability Framework

9.1.1 The Classical Universal-Composability Framework

The Universal Composability (UC) framework was introduced by Canetti in 2001 [i.53]. Oftentimes, a communication or security protocol can be subdivided into one outer protocol that uses multiple subroutines, which are we referred to as sub-protocols in the present document. A protocol or subprotocol tries to attain a certain security property, such as authentication or authorization. In the UC framework, how well a protocol does at attaining such a security property is measured by how closely it manages to realize its idealized functionality.

The unique and attractive feature of the UC framework is that, if a real sub-protocol is proved to behave like an ideal sub-protocol in the UC framework, then the real sub-protocol can always be substituted for the ideal sub-protocol in the security analysis of the outer protocol, regardless of what this outer protocol does. This is a very powerful tool to extend and build up proofs for complex protocols. There are two flavours of this protocol: one is statistical UC and the other is computational UC. The first provides statistical security, so attackers can be computationally unbounded without breaking the protocol, and the latter provides computational security, in which case only probabilistic polynomial-time attackers are considered.

The way this framework is formalized, is by modelling all involved systems as Interactive Turing Machines (ITMs), which are abstract models of computation that can simulate any computer algorithm that communicates with other systems (additionally modelled as ITMs). The UC framework guarantees the parallel composability property by introducing an environment ITM, which initiates adversaries (modelled as ITMs), a protocol π between other ITMs, and observes outputs. The goal of a proof in the UC framework is to show that for all possible environments and all possible adversaries, it is possible to find a simulator such that an execution of the environment with the adversary and sub-protocol ρ behaves almost identically to an execution of the environment with the simulator and ideal sub-protocol ϕ.

NOTE: Formally, the adversary outputs a 0 or 1 after observing the protocol and initiating adversaries. The probabilities that the output of the adversary is 0 or 1 are analysed, and these probabilities should be negligibly close in both scenarios.

Given the fact that a simulator on the ideal sub-protocol ϕ has less power than the adversary on the real sub-protocol ρ, this shows that any attack on the real sub-protocol ρ would also work on the sub-protocol ϕ, indicating that the real protocol emulates the ideal protocol.

This framework is very suitable for proving the security of protocols in general, but it is perhaps even more relevant for the area of Multi-Party Computation (MPC). In MPC, protocols are created that should have the same functionality and guarantees that a trusted third party provides, without actual access to a trusted third party. The goal is therefore to show that the protocol emulates a trusted third party.

9.1.2 Universal Composability and Quantum Adversaries

With regard to quantum security, theorem 2 of [i.51] proves that security proofs in the statistical UC framework still hold in a quantum setting. More concretely, if π is a classical protocol that statistically UC-emulates a certain classical functionality F, then π statistically quantum-UC-emulates F. This means that proofs in the statistical UC framework still hold against quantum adversaries, provided that the underlying primitives are quantum safe. However, it is not generally true that classical statistical indistinguishability implies quantum statistical indistinguishability. Additionally, classical computational UC-security does not imply quantum-computational UC-security, so this would require a new proof.

9.2 The Indifferentiability Framework

9.2.1 The Classical Indifferentiability Framework

The indifferentiability framework was introduced in 2004 by Maurer et al. [i.34]. It is an answer to a problem seen in proofs on indistinguishability. Indistinguishability proofs are used to motivate why a certain cryptographic system can replace an ideal function *F* with a cryptographic system *S*, by showing that an adversary cannot distinguish between *F* and *S*. However, this is only possible for systems with entropy. In other words, if *S* uses keys or other sources of randomness unknown to the attacker, then indistinguishability proofs are sufficient, but not if *S* is deterministic or the randomness is made public. The main application of the indifferentiability framework is hash functions, since cryptographic systems often specify a concrete hash function.

In the indifferentiability framework, systems are modelled as conditional probability distributions with inputs and outputs. Even though it seems restrictive, it is still possible to model computer algorithms in this framework, because the output of a computer algorithm can also be modelled as a conditional probability distribution conditioned on the input. To capture the nature of cryptographic systems, the Indifferentiability framework models the input through two channels: a private channel and a public channel.

The difference between *indistinguishability* and *indifferentiability* is subtle, but indistinguishability says that for a real system C', an ideal system C, and for every system D, called the distinguisher, the distinguisher behaves almost identically (e.g. the probabilities that it outputs either 0 or 1 are negligibly close) if:

1) The distinguisher has no access to the input interface of C', but observes the output interface of C'.

2) The distinguisher has no access to the input interface of C, but observes the output interface of C.

A proof of indistinguishability is enough when the goal is to substitute ideal system C for C', under the following assumptions:

1) No external party can influence the behaviour of C.

2) No external party has access to the randomness of C.

Such assumptions are acceptable for keyed primitives, as long as the key is not known, because a keyed primitive is essentially a random primitive drawn from a distribution of primitives with deterministic behaviour. However, for other applications such assumptions are not reasonable, which is the case for hash functions because they essentially are primitives with fixed behaviour. If a hash function is used instead of a random oracle, indistinguishability is not sufficient.

Since the indifferentiability framework is generally used to prove that specific constructions of hash functions are indifferentiable from random oracles, the following is an explanation on how that is done. Generally, hash functions use mechanisms that are based on other primitives. For example, SHA-3 is built using a sponge construction, which uses a compression function as a primitive. In the following, denote by C^F the outer construction - e.g. sponge construction - with an ideal inner construction, and by F the ideal version of the inner construction, such as an *ideal compression function*, which is basically a random oracle with a fixed-length input.

In the indifferentiability game, there is a distinguisher D, who needs to distinguish two scenarios. In scenario one, the distinguisher D is provided with:

- The output of construction C^F using ideal primitive F.

- The ideal primitive F. If the primitive is a symmetric primitive, such as a block cipher, then an efficient inverse F^{-1} exists, which the distinguisher also has access to.

In scenario two, the distinguisher D is provided with:

- The output of random oracle H.
- A simulator S^H that simulates the primitive F (and F^{-1}) based on the random oracle H.

It then needs to be shown that there exists a polynomial-time simulator S^H such that for all polynomial-time distinguishers D, the probability that D can distinguish the two scenarios is negligible in the security parameter. More formally, assume that D outputs a bit, where without loss of generality it outputs 0 if it thinks it is provided with scenario 1, and 1 if it is provided with scenario 2. It then needs to be proven that:

$$\exists S^H . \forall D . |Pr[D(C^F, F/F^{-1}) = 1] - Pr[D(H, S^H) = 1]| \leq \epsilon,$$

where ϵ is negligible in the security parameter and S^H, D are polynomial-time.

9.2.2 Quantum Indifferentiability

The indifferentiability framework and the proofs built upon it are inherently classical. That is, it is not evident whether classical indifferentiability proofs still hold against quantum adversaries. Classical indifferentiability has been proven for many constructions already. Specifically, the sponge construction used in SHA-3 was shown to be classically indifferentiable from a random oracle in [i.28].

The indifferentiability game for hash functions from the previous clause can easily be extended to the quantum case, by making both S^H and D *quantum*-polynomial-time algorithms. In [i.35], Carstens et al. prove under some quantum-information-theoretical conjecture that the sponge and Feistel constructions are not information-theoretically quantum-indifferentiable, which are popular constructions for cryptographic primitives. If this conjecture were true, then SHA-3 would not be quantum-indifferentiable from a random oracle. Fortunately, the work of [i.50] proves the sponge construction to be quantum-indifferentiable, disproving the earlier conjecture. They do this using the compressed oracle technique. As a result, SHA-3 was proven to be quantum-indifferentiable from a random oracle and can safely be used to instantiate random oracles.

9.3 Limitations

It is important to note that both the UC and indifferentiability frameworks have limitations, as illustrated in the work of Ristenpart et al. [i.36]. They first examine the indifferentiability framework and provide a scheme that is secure in the ROM, but insecure when instantiated with a concrete hash function, even though this hash function is indifferentiable from its ideal functionality: the random oracle. This scheme is a hash-based storage auditing scheme, which can be used when a server stores files and the user wants to verify that the file is present in the database (e.g. the database owner did not throw away random files to save space). The scheme uses an ideal compression function f.

NOTE: An ideal compression function is a function that takes a fixed-length input and provides an output of smaller length such that it is hard to determine what the input was, given the output. These can be used as building blocks to build hash functions.

When a user wants to verify that their file M is still in the database, they send the challenge C. The database owner then has to provide the response:

$$r = f(f(IV, M), C),$$

for some fixed constant string IV (the initialization vector).

The construction as provided above was shown to be indifferentiable from a random oracle in [i.37]. However, the database owner can cheat by computing $Y = f(IV, M)$ when the document is initially received and computing $f(Y, C)$ as a response to any challenge C in the future. Coron et al. analyzed the proofs and concluded that the indifferentiability claims break for security notions captured by experiments that have multiple, disjoint adversarial stages. This is the case for the hash-based storage auditing scheme. In other words, a proof is multi-stage if an adversary can derive some state S from the input it gets that is smaller than the input itself and can use S to answer challenges. Examples of such experiments are the security notions of deterministic public-key encryption, password-based cryptography, hash function non-malleability and key-dependent message security. Security notions that are not affected are those that involve a single stage with a stateful adversary, such as IND-CPA, IND-CCA and EUF-CMA. Ristenpart et al. [i.36] additionally show that the same limitations hold for the Universal Composability framework.

10 Pseudo-random functions

10.1 The Quantum Security of Pseudo-Random Functions

In the previous clauses, the focus has been on asymmetric primitives. This has largely been the focus of post-quantum security analyses, since Shor's quantum algorithm breaks current asymmetric primitives in polynomial time, whereas symmetric primitives were believed to still be secure, albeit at a cost of (at worst) half the bit security on account of Grover's algorithm. However, the underlying mathematical problems are not the only consideration in the post-quantum security of the asymmetric primitives. Specifically, a lot of additional research has gone into understanding the post-quantum security of hash functions, and a lot of the constructions underlying hash functions are used in symmetric primitives. This raises the question whether security notions need to be redefined for symmetric primitives as well.

The first results regarding the quantum security of symmetric constructions was on account of Zhandry, who provided the first analysis of Quantum-safe Pseudo-Random Functions (QPRFs) [i.38]. He provided two models to reason about the powers of a quantum adversary for QPRFs, which apply to all symmetric primitives in general:

1) *Standard Security:* a quantum adversary can do local quantum computations, but input to and output from the primitive in question is purely classical.

NOTE: This is not to be confused with the standard model of reductions.

2) *Quantum Security*: a quantum adversary has quantum access to the primitive in question, such that a quantum state (e.g. a superposition) can be provided as input and the output is a quantum state as well.

Even though the quantum security gives a lot of power to the adversary, which might not directly be applicable to all practical situations, it captures a wider class of attackers.

EXAMPLE: If a quantum internet becomes wide-spread, this class of attackers becomes more prominent.

The conservative long-term approach is therefore to use symmetric primitives that attain quantum security, but for the foreseeable future, standard security is more realistic.

10.2 Pseudo-Random Functions and Message Authentication Codes

In [i.38], Zhandry notes that classical proofs of existing PRFs based on pseudo-random generators used reasoning that does not apply to quantum adversaries. Simply put, the classical proofs used the argument that a classical adversary can only call the PRF a polynomial number of times, which evaluates a polynomial number of 'internal states', even though the PRFs have an exponential number of internal states.

In particular, Goldreich et al. create in [i.39] a PRF using a keyed length-doubling pseudo-random generator and construct a binary tree. The first node is the key itself. Then the edge between this node and the left child-node is assigned the value 0 and the edge between the node and its right child-node is assigned value 1. Then the pseudo-random generator is applied to the value in the node to obtain a string that is twice as large as the string in the node itself. The left half is assigned to the left child node (with edge value 0) and the right half is assigned to the right child node (with edge value 1). This is done recursively.

Whenever the PRF is called on an input b, it decomposes the value into bits b_0 through b_N for some N. Then it starts at the first node, it traverses down the edge with value b_0, at the next node it traverses down the edge with value b_1, and so on. The value at the leaf node is then provided as output.

Each call to the PRF only visits a polynomial number of nodes on each level. Clearly, there are an exponential number of nodes. This argument can then be used to construct a polynomial-time adversary B, who can distinguish the underlying pseudo-random generator from random, given a polynomial-time adversary A who can distinguish the PRF from random. More specifically, B succeeds with polynomially smaller success probability than A, so it still runs in polynomial-time, if it wants to achieve the same success probability.

Zhandry notes that many other PRFs have security proofs with similar arguments. However, quantum adversaries could access the PRFs in superposition, possibly accessing all of the (exponentially many) nodes at the same time with one query. Now the adversary B constructed from A succeeds with exponentially smaller probability. To solve this gap, Zhandry provides quantum-security proofs for PRFs based on pseudo-random generators, pseudorandom synthesizers or lattices. Additionally, Zhandry proves that if secure PRFs exist, then there are standard-secure PRFs that are not QPRFs. Zhandry specifically shows that certain standard-secure PRFs can be turned into PRFs with a hidden period, which can be extracted using Simon's algorithm by quantum adversaries, but not by classical adversaries. They are therefore not QPRFs. In other words, there are PRFs that are indistinguishable from random, if a quantum adversary has classical access to the PRF, but it is distinguishable from random if the adversary has quantum access to the PRF, so even though the three PRFs for which Zhandry provides alternative proofs turned out to be quantum-safe, it does not generally hold that all standard-secure PRFs are quantum-safe. Some negative results are already known, namely PRFs based on three-round Feistel cipher are prone to quantum distinguishing attacks [i.40] and PRFs based on the Even-Mansour cipher are also prone to quantum distinguishing attacks [i.41].

The quantum-security results for PRFs have direct consequences for other cryptographic applications. For example, Boneh and Zhandry [i.42] show that quantum-safe PRFs are quantum-safe Message Authentication Codes (MACs). More specifically, they are existentially unforgeable under *quantum* chosen-message attacks. However, not all MAC constructions are quantum-safe. Notably, Kaplan et al. [i.43] show that Simon's algorithm can be used to break standardized modes of operation such as CBC-MAC, PMAC and GMAC in the quantum security model. These are based on block ciphers and are still broken if the underlying block cipher is quantum-safe.

Annex A:
Change history

Date	Version	Information about changes
02/2023	V0.0.6	Text was added to the preliminaries; the placeholders have been replaced with complete text. The introduction to Proof of Knowledge systems was moved to the preliminaries. A few terms have been added and minor stylistic changes have been made.
02/2023	V0.0.7	An executive summary was added, as well as more details to Fiat-Shamir problem.
06/2024	V1.0.0	More references have been added, The Fiat-Shamir problem has been elaborated on, several passages have been updated for clarification and parts that are subject to change over time and therefore difficult to maintain have been removed, the references have been cleaned up.
06/2024	V1.0.1	References have been corrected and added. The reference to OW-PCA security has been clarified.

History

Document history		
V1.1.1	October 2024	Publication