ETSI GR QSC 003 V1.1.1 (2017-02)

Quantum Safe Cryptography;
Case Studies and Deployment Scenarios

Disclaimer

The present document has been produced and approved by the Quantum-Safe Cryptography (QSC) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
DGR/QSC-003

Keywords
algorithm, authentication, confidentiality, 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 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is 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 IPR Policy, no investigation, 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.

Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) Quantum-Safe Cryptography (QSC).

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.

1 Scope

The present document examines a number of real-world uses cases for the deployment of quantum-safe cryptography (QSC). Specifically, it examines some typical applications where cryptographic primitives are deployed today and discusses some points for consideration by developers, highlighting features that may need change to accommodate quantum-safe cryptography. The main focus of the document is on options for upgrading public-key primitives for key establishment and authentication, although several alternative, non-public-key options are also discussed.

The present document gives an overview of different technology areas; identify where the security and cryptography currently resides; and indicate how things may have to evolve to support quantum-safe cryptographic primitives. Clauses five and six discuss network security protocols, using TLS and S/MIME as typical examples. These are contrasted in clauses seven and eight by an examination of security options for IoT and Satellite use cases, which have very different requirements and constraints than traditional internet-type services. Some alternatives to public key protocols are reviewed in clause nine. Authentication requirements are discussed in clause ten and some forward-looking examples providing advanced functionality are examined in clause eleven.

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] ETSI: "Quantum safe cryptography and security," ETSI White Paper No. 8, 2015.

[i.2] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2", 2008.

[i.3] Draft RCF draft-ietf-tls-tls13-09: "The Transport Layer Security (TLS) protocol version 1.3", 5 October 2015.

[i.4] C. Peikert: "Lattice Cryptography for the Internet" IACR ePrint 2014/070, 2014.

[i.5] J. W. Bos, C. Costello, M. Naehrig and D. Stebila: "Post-quantum key exchange for the TLS protocol from the ring learning with errors problem" IACR ePrint Archive 2014/599, 2014.

[i.6] V. Singh: "A Practical Key Exchange for the Internet using Lattice Cryptography" IACR ePrint 2015/138, 2015.

[i.7] E. Alkim, L. Ducas, T. Pöppelmann and P. Schwabe: "Post-quantum key exchange - a new hope" IACR ePrint 2015/1092, 2015.

[i.8] Draft IETF draft-whyte-qsh-tls13-01: "Quantum-safe hybrid (QSH) ciphersuite for Transport Layer Security (TLS) version 1.3 (draft RFC)", 20 September 2015.

[i.9] O. Garcia-Morchon, R. Rietman, L. Tolhuizen, J.-L. Torre-Arce, S. Bhattacharya and M. Bodlaender: "Efficient quantum-resistant trust Infrastructure based on HIMMO", IACR ePrint 2016/410, 2016.

[i.10] D. McGrew: "Living with post quantum security", NIST workshop on cubersecurity in a post quantum world, 2015.

[i.11] Z. Zheng, W. White and J. Schanck: "A quantum-safe circuit-extension handshake for Tor" in NIST Workshop on Cybersecurity in a Post-Quantum World, 2015.

[i.12] ETSI GR QSC 001 (V1.1.1): "Quantum-Safe Cryptography (QSC); Quantum-safe algorithmic framework".

[i.13] IETF RFC 5751: "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2", 2010.

[i.14] D. McGrew, P. Kampanakis , S. Fluhrer, S.-L. Gazdag , D. Butin and J. Buchmann: "State Management for Hash-Based Signatures" IACR ePrint, vol. 2016/357, 2016.

[i.15] Philips: "Philips Hue".

NOTE: Available at www.meethue.com.

[i.16] O. Garcia-Morchon: "Security for Pervasive Healthcare" PhD Thesis, RWTH University, 2011.

[i.17] ZigBee® Alliance.

NOTE: Available at www.zigbee.org.

[i.18] IETF RFC 7228: "Terminology for Constrained-Node Networks", 2014.

[i.19] A. Waller, A. Byrne, R. Griffin, S. La Porta, B. Ammar and D. Lund: "Case Study Specification and Requirements" 2015.

NOTE: Available at http://www.safecrypto.eu/.

[i.20] A. Menezes, P. van Oorschot and S. Vanstone: "Chapter 13: Key Management Techniques, Handbook of Applied Cryptography".

NOTE: Available at http://cacr.uwaterloo.ca/hac/.

[i.21] Kerberos® Consortium.

NOTE: Available at www.kerberos.org.

[i.22] IETF RFC 1510: "The Kerberos Network Authentication Service (V5)", 1993.

[i.23] IETF RFC 7252: "The Constrained Application Protocol (CoAP)", 2014.

[i.24] IETF RFC 4279: "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", 2005.

[i.25] O. Garcia-Morchon: "DTLS-HIMMO: Achieving DTLS certificate security with symmetric key overhead" in 20th European Symposium on Research in Computer Security (ESORICS), 2015.

[i.26] R. Blom: "Non-public key distribution" in CRYPTO 82, New York, 1983.

[i.27] T. Matsumoto and H. Imai: "On the key predistribution system - A practical solution to the key distribution problem" in CRYPTO 87.

[i.28] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro and M. Yung: "Perfectly-secure key distribution for dynamic conferences" in CRYPTO 92, 1992.

[i.29] W. Zhang, M. Tran, S. Zhu and G. Cao: "A Random PerturbationBased Pairwise Key Establishment Scheme for Sensor Networks" in ACM MobiHoc, 2007.

[i.30] M. Albrecht, C. Gentry, S. Halev and J. Katz: "Attacking cryptographic schemes based on "perturbation polynomials" in 16th ACM conference on Computer and communications security (CCS '09), 2009.

[i.31] O. Garcia-Morchon, R. Rietman, L. Tolhuizen, J.-L. Torre-Arce, S. Moon, D. Gomez-Perez, J. Gutierrez and B. Schoenmakers: "Attacks and parameter choices in HIMMO" IACR ePrint 2016/152, 2016.

[i.32] TUD: "Practical hash based signatures", 2016.

NOTE: Available at www.pqsignatures.org.

[i.33] IEEE 1609.2-2013™: "Wireless Access in Vehicular Environments", 2013.

[i.34] NIST: "The keyed-hash Meassage Authentication Code (HMAC)" FIPS-198-1, 2008.

[i.35] ISO/IEC 9797 parts 1 and 2: "Message Authentication Codes (MACs)", 1999.

[i.36] L. Ducas, V. Lyubashevsky and T. Prest: "Efficient identity-based encryption over NTRU lattices," IACR ePrint 2014/794, 2014.

[i.37] D. Apon, X. Fan and F.-H. Liu: "Fully secure lattice-based IBE as compact as PKE" IACR ePrint 2016/125, 2016.

[i.38] S. Agrawal, D. Boneh and X. Boyen: "Lattice basis delegation in fixed dimension and shorter ciphertext hierarchical IBE" in EUROCRYPT 2010 Volume 6110 of the series Lecture Notes in Computer Science pp 553-, 2010.

[i.39] D. Cash, D. Hofheinz, E. Kiltz and C. Peikert: "Bonsai Trees, or How To Delegate a Lattice Basis" Journal of Cryptology October 2012, vol. 25, no. 4, pp. 601-609, 2012.

[i.40] KLU: "HEAT project".

NOTE: Available at https://heat-project.eu/.

[i.41] K. Xagawa: "Improved (hierarchical) inner-product encryption from lattices" IACR ePrint 2015/249, 2015.

[i.42] S. Argawal, D. Freeman and V. Vaikuntanathan: "Functional encryption for inner product predicates from learning with errors" IACR ePrint 200/410, 2011.

[i.43] C. Gentry, A. Sahai and B. Waters: "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based" IACT ePrint 2013/340, 2013.

[i.44] Z. Barkerski, C. Gentry and V. Vaikuntanathan: "(Leveled) fully homomorphic encryption without bootstrapping" IACR ePrint 2011/277, 2011.

[i.45] NIST: "Report on Post Quantum cryptography" NISTER 8105, 2016.

3 Abbreviations

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

6LoWPAN	Ipv6 over Low power Wireless Personal Area Networks
ABE	Attribute-based Encryption
AES	Advanced Encryption Standard
CoAP	Constrained Application Protocol
COTS	Commercial Off The Shelf
DH	Diffie-Hellman
DSA	Digital Signature Algorithm
DTLS	Datagram Transport Layer Security
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
FHE	Fully Homomorphic Encryption
HEAT	Homomorphic Encryption Applications and Technology
HFE	Hidden Field Equations
HIBE	Hierarchical Identity-Based Encryption
HIMMO	Hiding Information Mixing Modular Operations
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force

IKE	Internet Key Exchange
IoT	Internet of Things
IPsec	Internet Protocol Security
KDC	Key Distribution Centre
KMS	Key Management Server
KTC	Key Translation Centre
LoRA™	Low Power Wide Area Network for IoT
LTE™	Long Term Evolution
MAC	Message Authentication Codes
MIT	Massachusetts Institute of Technology
oneM2M	Standards for machine to machine
PKC	Public Key Cryptography
PKI	Public Key Infrastructure
PSK	Pre-shared key
QSC	Quantum-Safe Cryptography
QSH	Quantum Safe Hybrid
RFC	Request For Comments
RSA	Rivest Shamir Adleman
S/MIME	Secure/Multipurpose Internet Mail Extensions
SHA	Secure Hash Algorithm
SMTP	Simple Mail Transfer Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
V2X	Vehicle to everything
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network
W3C	Worldwide Web Consortium

4 QSC deployment scenarios

Cryptography is already widely-used and is rapidly becoming ubiquitous, appearing in everything from internet and mobile applications to emerging technologies such as the Internet of Things (IoT). Over the past 20 to 30 years, information storage has transitioned from a paper-based society, where physical copies of sensitive documents were once locked in filing cabinets and safes, to one where sensitive documents are now stored electronically. Although not obviously visible, this migration continues to occur. More information is now stored on databases within cloud environments, completely off-site to where the data originated. This poses an interesting problem for the future: how to keep sensitive data from unauthorised access both while being transferred over a network and while stored electronically.

Furthermore, quantum computers are no longer the thought experiments they once were not very long ago. There are many approaches to quantum computation, including super-conducting qubits, ion traps, nuclear magnetic resonance, quantum annealing and others. As of this date, small quantum computers exist in laboratories, although they are sufficiently under-powered to solve complex cryptographic problems in reasonable periods of time.

While these small quantum computers pose no threat to information security at present, it is already possible to observe their efficiency in solving certain classes of mathematical problems. This is why there is an increased priority by industry and governments on quantum computer research. This priority is evidenced by the propensity for increased investment in recent years. This is also why there is an increased priority on investments in quantum safe cryptography.

The wide range of applications being built today is accompanied by a diversity of security, efficiency and policy requirements and a variety of different computing platforms ranging from highly constrained devices to high end computing; so it seems unlikely that there would be a single one-size fits all solution for quantum resistance. The document presents some real-world use cases of where cryptography is deployed today and investigates how things may need change to migrate to quantum-safe cryptography.

The present document gives an overview of different technology areas, identify where the security and cryptography currently resides, and indicate how things might have to evolve or change to support quantum-safe cryptographic primitives. More detailed analysis of these examples may appear as separate ISG documents.

NOTE: The present document is a survey and should not be treated as an official ETSI endorsement of any products or standards mentioned below. Nor is it the intention of the document to prescribe how protocols defined and maintained by any other standards bodies should evolve. The intention is simply to discuss the consequences of using certain primitives in some typical example use-cases.

5 Network security protocols

5.1 Introduction

An over-simplified but stereotypical model for public key-based communications is the following. Two parties wish to establish a secure and authenticated communications link across a network. One or both parties obtain signed certificates from a trusted Public Key Infrastructure (PKI) containing the identity and public key of the other party with whom they wish to communicate. After verifying the validity of the certificate and the counterpart's identity, a public-key based handshake protocol is used to establish a secret session key known only to the two parties, and this session key is typically input to a block cipher to encrypt the subsequent communications between the pair.

Most current public-key-based communications are designed to be secure against *classical* adversaries. This means that the handshake mechanism allows two authenticated parties to agree on a secret session key that is secure against attackers with traditional computing resources. It is widely accepted that most currently-deployed public-key based communications will become vulnerable to a future attacker with access to large-scale quantum computers. For this reason, a growing body of research is being focused on developing quantum-safe public-key based handshake protocols.

Protocols such as Internet Protocol Security (IPsec), Internet Key Exchange (IKE), Transport Layer Security (TLS) protocol, Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP) and others are ubiquitous internet or application level protocols used to secure a host of modern communications applications including web browsing, e-mails, Virtual Private Networks (VPNs), Voice over Internet Protocol (VoIP), instant messaging, etc. chapter 4 of the ETSI whitepaper [i.1] gives an overview of the sorts of changes that would need to be considered to incorporate quantum-safe primitives into common network protocols such as these.

Most of these protocols are defined and maintained by the Internet Engineering Task Force (IETF), Worldwide Web Consortium (W3C) or similar groups and it is not in the remit of ETSI ISG QSC to decide how these protocols should evolve. However, given the ubiquitous nature of these protocols, it is necessary to have some understanding of the compatibility of any ETSI recommended primitives with the wider commercial infrastructure.

Clauses 5.2 to 5.3.3 focus on TLS as an important example of a real-world use case. They look at some specific proposals in the literature for ways to upgrade TLS to be quantum secure. The TLS [i.2] and [i.3] protocol suite provides a cryptographic layer through which network application protocols such as Hypertext Transfer Protocol Secure (HTTPS) (used for web browsing), SMTP (e-mail) and VoIP (voice) can be securely tunnelled. TLS is widely used to underpin the security of many of the other technology areas discussed in the remainder of the present document.

5.2 TLS

5.2.1 TLS cryptography

TLS version 1.2, defined in [i.2] and its intended upgrade, still in draft at [i.3], make wide use of public-key cryptography supported by PKI to provide key establishment and authentication services. These are currently based on the well-known factoring or discrete logarithm primitives Rivest Shamir Adelman (RSA), Diffie-Hellman (DH), Digital Signature Algorithm (DSA), Elliptic Curve Diffie-Hellman (ECDH) and Elliptic Curve Digital Signature Algorithm (ECDSA) and it is precisely these primitives that need to be upgraded to be quantum-safe. Since TLS is so widely used, it is here that the best and most modern primitives to provide secure and efficient quantum-safe replacements for the current Public Key Cryptographic (PKC) protocols will need to be deployed.

TLS also makes use of symmetric cryptography e.g. the block cipher Advanced Encryption Standard (AES) for data encryption and the Secure Hash Algorithm (SHA) for digital signatures and certificate verification. Since these primitives may be regarded as already quantum-safe, or easily upgraded to be quantum-safe by increasing key or block sizes, they will not be discussed further here and the focus will instead be on the public-key primitives.

There have been three main approaches suggested so far for possible migration paths for upgrading TLS to incorporate quantum-safe primitives.

NOTE: The large amount of work required to define new cipher suites or upgrade any other infrastructure required to support quantum safe primitives is not discussed here.

5.2.2 Drop-in replacement

The most straightforward proposal is to replace some or all of the current public-key primitives with like-for-like quantum-safe drop-in replacements, assuming that suitable alternatives with similar security levels and efficiency properties are available. A promising example of this approach is the Ring Learning With Errors proposal [i.4]. An early proof of concept demonstration [i.5] integrated a version of this scheme into OpenSSL and compared this against standard TLS using elliptic curve cryptography. The authors reported that that their preliminary constant run-time implementation looked practical, producing a typical reduction of 1.2x in throughput for serving HTTPS connections. More recent implementations [i.6] and [i.7] report greatly increased throughputs of 8x-20x over [i.5] and halve the communications overhead, for the same 128-bit security level. These papers all include security, soundness and implementation analysis for the schemes presented.

5.2.3 Hybrid scheme

A second proposal is to introduce *hybrid* schemes which derive an encryption key from some combination of the outputs from a well-studied and trusted classical key agreement scheme and a separate and perhaps novel quantum-safe key agreement scheme. This might be a viewed as an interim step in the migration to using purely quantum-safe cryptography, or as a way of providing extra functionality or security. One such example for TLS is Quantum Safe Hybrid (QSH) [i.8] which proposes combining a standard classical TLS handshake with an NTRUEncrypt key transport. An initial implementation for TLS 1.2 reports a similar reduction in throughput of just 1.2x for the 128-bit security level.

NOTE: These figures are based on an implementation in WolfSSL using NTRUEncrypt parameter set eess439ep1 for 128 bits of classical security. In this case, typical timings give ECDHE-ECDSA-AES256-SHA384 with QSH taking 9,31 ms compared with 7,48 ms for the original classical TLS 1.2 ECDHE-ECDSA-AES256-SHA384 exchange.

Another example of a hybrid architecture is presented in [i.9].

5.2.4 Re-engineering

A more radical approach would be to re-engineer the infrastructure of the internet and use a systems engineering approach to mitigate performance issues and allow larger key sizes to be handled. One such proposal for TLS is [i.10] which envisions using session resumption techniques to minimize the transmission and storage of large public keys between peers on a network, together with using symmetric keys supplied by trusted servers to secure individual sessions. Clearly it would be major undertaking to migrate for the entire Internet to a new architecture such as this but this approach might be more suitable for smaller networks.

5.3 Discussion

5.3.1 Integration into the protocol stack

This first example already serves to highlight some important points for developers considering how to deploy quantum safe cryptography.

A considerable amount of work will be required to integrate quantum-safe cryptography into real-world systems. Hybrid key exchanges are not always allowed by network protocols or they may not fit into the bandwidth currently allocated for handshakes [i.11] and even the apparently straightforward "drop in replacement" approach would require new cipher suites and other infrastructure to be defined.

Quantum-safe primitives with very large public keys initially seem unsuitable for widespread deployment using the "drop in" approach as there are often restrictions on packet size, handshake size or other bandwidth issues. In clause 4.3.3 of the ETSI whitepaper introducing quantum-safe cryptography [i.1] TLS record fragment size was raised as a possible technical concern. This concern was also raised in the TLS 1.2 specification [i.2], where it was recommended that protocol implementations should fragment and reassemble the handshake protocol messages correctly. The specification further noted that the method of fragmentation and reassembly is an implementation detail, i.e. how this is done would be left up to the implementer, but may have an impact on its performance and interoperability. Going forward proper fragmentation and re-assembly of handshake messages will be crucial in ensuring the eventual migration to quantum-safe security.

TLS packet fragmentation works over TCP which is a synchronized connection oriented protocol. That is, TCP presents data as a stream to TLS. Likewise TLS presents data as a stream to applications (e.g. HTTP).The situation is drastically different with UDP since this is a packet oriented protocol. Applications over UDP (e.g. IKE or DTLS) need to supply their own fragmentation layer.

The available computational resources of emerging technologies is an important consideration: it is very different to perform cryptographic mathematics on a server-class machine than an embedded processor in a smartphone, heart-rate monitor or other resource-constrained device; see clause 7.5 for more discussion. The network architecture and the impact of any latency introduced by the processing of post quantum crypto primitives on the timeout requirements are other factors to be considered. Although the TLS protocol itself does not specify any timeout constraints, the computational cost of key pair generation, shared secret derivation, encryption/decryption, and/or signature generation and validation may affect the underlying protocol.

In cases where a cryptographic operation requires more time to complete than the configured connection timeout at the transport or application layer, the connection will close before the handshake is completed. Generating and transporting large public keys may become an important issue in low data-rate or low quality networks.

5.3.2 Handling large key sizes

Suppose a developer wanted to set up a "high security" TLS 1.2 network based on quantum safe primitives with very large parameters, such as a code-based key transport scheme and a hash-based signatures (see ETSI GR QSC 001 [i.12] for options here and information on parameter sizes).

The handshake component of TLS is the only one that is affected by the replacement of the classic public-key primitives with quantum-safe equivalents. It is responsible for algorithm negotiation, peer authentication, and symmetric key establishment. The entire handshake is completed in two message sequence round-trips, comprising four messages between the client and server [i.2]. In the case of classic cryptographic algorithms, handshake messages are typically always smaller than the maximum record size, and therefore do not require fragmentation. In the case of code-based key transport scheme and a hash-based signature these messages would dramatically increase in size. This may require message fragmentation by the Record Layer component, to accommodate messages larger than the maximum specified record size.

There are three Handshake component messages that are directly impacted by the quantum-safe primitives, due to the increased public key, signature and ciphertext sizes:

- *Certificate* message is sent by the server, and optionally the client. This message typically carries the entire certificate chain starting with the end entity certificate, followed by zero or more intermediate certificates, and optionally ending with the root certificate.
- *ServerKeyExchange* message is sent by the server. This message contains the server's public key-establishment key, along with the signature of that key.
- *ClientKeyExchange* message is sent by the client. In the case of key agreement, this message carries client's public key-agreement keys. In the case of key transport, it carries the client-generated pre-master secret encrypted with the server's public encryption key.

Authentication: Although the private signing key may be very large, it has no impact on the TLS protocol because it is not transmitted, in whole or in part, during the handshake. The root public key size also does not produce any impact the TLS protocol, as its size is comparable to the current classic primitives' public key sizes. Signature size, on the other hand, will significantly increase the size of the certificates and the size of the signed digest in the key establishment messages. This means that the *Certificate* and *ServerKeyExchange* messages will grow in size, and this may require message fragmentation.

Key agreement: Fragmentation may also be a practical issue for code-based key agreement. On the client side, a generated pre-master secret will be encrypted with public key before it is sent to the server in the *ClientKeyExchange* message. If the message exceeds the record size limit if 16 kilo-octets it will be fragmented. Code-based keys are not only very large but they take a relatively long time to generate. In practice, hello messages larger than 1 kilo-octet will like to increase the connection error to an unacceptable level. This suggests that further investigation may be needed to understand fragmentation limits of the various implementations of TLS.

Forward security: High security TLS networks may also require forward secrecy, which is not typically an inherent characteristic of key transport schemes [i.12]. Cipher suites that do not provide forward secrecy are considered insecure and many have been disabled by commercial web browsers. (This may not be a problem if the network is isolated from the public Internet and uses bespoke software.) However it may sometimes be possible to add forward security in the following way: Generate a public-private key pair on the server side and send the signed public key to the client. After verifying the signature, send to the server the pre-master secret encrypted with server's public key. This behaviour can be specified in the ciphersuite description and is permitted by TLS. Note however that the generation of an ephemeral key pair will introduce some latency which in some cases trigger could time-outs.

5.3.3 Is quantum-safe authentication required today?

A common model for distinguishing between two types of quantum threat is to classify potential adversaries as either passive or active attackers. A *passive* attacker is capable of recording current network traffic and will break out the secret session key when quantum computers becomes available at some point in the future, while an *active* attacker is assumed to have access to a quantum computer and can therefore break the authentication and forge a certificate today. The communication is compromised even though the session key may itself be derived via a quantum-safe channel.

Although confidentiality and authentication are equally important in providing secure communications, under this model quantum-safe confidentiality is seen as a matter that needs to be addressed today, while quantum-safe authentication is not strictly needed until the day quantum computers arrive. Hence under this view there are two steps to achieve quantum safety: protecting current public key-based handshakes against passive attackers, and adding quantum-safe authentication, possibly at a later date. Both of the first two proposals [i.5] and [i.8] are content to retain using current (non quantum-safe) digital signatures such as RSA or ECDSA to provide authentication for TLS.

See clause 10 for more discussion on quantum safe authentication.

6 Offline services

6.1 Secure e-mail

While much of the public discussion so far with regards to the need for quantum safe cryptography has focused on interactive communications such as TLS or other network protocols, there has been rather less emphasis on off-line applications like secure e-mail, based for example on S/MIME [i.13]. Here it the receiver is not assumed to be on-line at the time of sending a message and so two-way, interactive key establishment protocols are not appropriate.

S/MIME is often used by governments and enterprises to ensure the security of e-mail communications which may contain details of strategic plans, health records, human resources information, military planning, or similar information which requires long term security.

The main difference that secure e-mail has over interactive protocols like TLS is the reliance on the fact that the receiving party already has a certificate, and hence, a public key in place to use for key establishment to achieve one-pass protocol. Protocols such as S/MIME use public key encryption or static key exchange for key establishment, with the static public key contained in a certificate to provide its authenticity. Key transport schemes, such as McEliece (see ETSI GR QSC 001 [i.12] for a survey of key transport options) are usually considered most appropriate to be used in offline scenarios and fit very well with the assumed functionality of typical secure e-mail applications. One practical drawback of many simple key transport schemes, is the loss of perfect forward secrecy, however it is often possible to overcome this in practical scenarios if the public keys are not too large and they can be generated efficiently. Other possibilities to consider are identity-based schemes, see clause 11.1.

6.2 Credentials for offline services

For many offline applications authentication via general-purpose signature algorithms such as lattice signatures or similar constructions for will be appropriate.

Hash-based signatures need to be used carefully to ensure that the consumption of state is monitored closely [i.14]; this implies a need for updating end-entity certificates on a periodic basis. One such model for updating credentials is already commonly used in the context of time-based expiration schemes for certificates, however things are much more challenging in applications where fixed hardware tokens like smart cards are used for credential storage. Policies for those hard tokens may not be flexible enough to incorporate usage-based credential updates.

For end-user certificates that exist in software, sometimes referred to as "soft certs", they typically have an expiration time associated with them that is defined within the certificate itself. When this time is close to expiring, the agent running on the client will typically automatically make a certificate request, PKCS10 in the X.509 world, to a certificate authority to request a new updated certificate, which when received is then placed in the updated certificate store. The old certificate is typically kept in order to facilitate decryption of old encrypted e-mail. So in the context of the use of hash based signatures, see clause 10.2.2, a similar request for new initialization data would be made once "enough" state had been used from the current private key. This method could be made automatic and seamless to the end user.

In the case of some types of credential storage, typically hardware orientated such as smart cards, the issuance of new credentials is often more manual in fashion and difficult to automate. A new card can be delivered or the end user can visit a card issuance office in order for new credentials to be placed on the card. Nevertheless the use of hardware for credential storage is a valid alternative to systems based on classical PKC, PKI and certificates, which may require major upgrades to support quantum safe cryptography see clause 5.3.1.

6.3 Discussion

Once again it seems clear that there will be much work required for developers to build fully quantum-safe systems. Not only will the core key agreement and authentication protocols need to be updated but also securing the supporting infrastructure and integrating protocols into wider systems will be just as important. Applications based on current PKC, PKI and certificates are widespread and will require major upgrades see clause 5.3.1 while the use of hardware-based credentials storage is a valid alternative which will still have a role to play.

7 Internet of Things

7.1 Introduction

The Internet of Things (IoT) refers to the increasing connectivity of "smart objects". This can be done in an isolated and ad-hoc manner or involve the connection of such objects to the Internet.

As an example, consider a home lighting use case, where devices such as light bulbs and switches are equipped with a processor and some communication component to form a local wireless network. The system further uses a gateway to the Internet connected to a WiFi™ wireless router. After the initial installation of the gateway and downloading an app to a smart phone, a user can easily pair the app with the gateway and add the lighting devices to the home network. The smart phone app can then control the lights, or create settings and schedules. The user could also enable more advanced features like geo-fencing so that the lights go off or on when someone leaves or enters the house, or enable the remote control of the smart lighting system over the Internet [i.15].

Next consider a healthcare use case within a hospital. Here, a new patient is registered by a nurse who retrieves previous electronic health records and connects a number of wireless sensors to help assess the condition e.g. to monitor breathing or blood pressure levels. A more powerful health monitoring device will gather the data collected from the various sensors and securely forward the information to the hospital's healthcare back end system which processes the medical data and distributes the results to nurses, doctors and other authorized recipients [i.16].

7.2 IoT cryptography

Given the vast number of possible IoT scenarios in domestic, retail, manufacturing, business, health, transport and other sectors, it is not surprising that many different solutions have been proposed and a wealth of different protocols have been used to connect smart objects together.

It seems clear that there will not be a single universal solution adopted for IoT networking and it is not possible for the present document to provide an exhaustive discussion of all the different options; nevertheless there are some general points that can be made.

Many IoT networks have a star topology around a central gateway or hub that is connected to the Internet (e.g. home lighting) or other core network (e.g. hospital Ethernet). Others employ some form of mesh networking to provide scalability, reduce power consumption or lower communication latency.

Symmetric key approaches are often used to secure the communications between the networked devices where a trust centre handles a network key shared between the devices. Devices can also share a key with the trust centre that can be used for network access or for distributing a pairwise key between a pair of devices on demand. Alternatively, a single key is pre-distributed to all devices and that key is then used to securely distribute a network key after an ad-hoc network is formed. These methods have some inherent weaknesses (e.g. lack of forward security) and are probably best suited to low-threat environments.

In the last few years there has been a move towards trying to deploy Public Key Cryptography (PKC) to improve the overall security of IoT networks and many of the protocols listed above try to incorporate PKC primitives such as ECDH or ECDSA when feasible. However it is a common and important feature of many IoT networks that deployments often have severe resource limitations. Cipher-suites based on PKC are relatively bulky and this can have a serious impact on the underlying network, which may have lossy communication links or low bandwidth. IoT applications may also be unsuitable for bulky PKC, for instance when a smart object only needs to report a few octets of application data but the setup of the secure channel requires several kilo-octets of PKC data. Hence for some IoT networks it may simply not be possible to deploy PKC, which typically has relatively large computational and communications overheads.

This is likely to be an increasingly important issue in the future since many of the quantum-safe primitives that are being proposed to replace current PKC for Internet-type applications have much larger computational or communication overheads than today [i.12]. Thus symmetric key approaches might be expected to remain important for many years to come.

7.3 Discussion

Resource limitations for emerging technologies are often in stark contrast to the resources assumed by network protocols such as TLS. Power and energy efficiency will be important considerations for IoT and 5G and there will be an increasing need to consider computational resources, bandwidth and latency in the selection and deployment of post quantum cryptography.

IETF RFC 7228 [i.18] contains a good discussion about power consumption for constrained devices and proposes a terminology. The introduction notes that:

The Internet Protocol Suite is increasingly used on small devices with severe constraints on power, memory, and processing resources, creating constrained-node networks. Small devices with limited computational, memory, and power resources, so-called "constrained devices" (often used as sensors/actuators, smart objects, or smart devices) can form a network, becoming "constrained nodes" in that network. Such a network may itself exhibit constraints, e.g. with unreliable or lossy channels, limited and unpredictable bandwidth, and a highly dynamic topology.

Constrained devices might be in charge of gathering information in diverse settings, including natural ecosystems, buildings, and factories, and sending the information to one or more server stations. They might also act on information, by performing some physical action, including displaying it. Constrained devices may work under severe resource constraints such as limited battery and computing power, little memory, and insufficient wireless bandwidth and ability to communicate; these constraints often exacerbate each other. Other entities on the network, e.g. a base station or controlling server, might have more computational and communication resources and could support the interaction between the constrained devices and applications in more traditional networks.

With this in mind, here are some things to consider when deploying post quantum cryptography for IoT or other constrained devices:

- Many current internet protocols will be unsuitable for low power environments such as loT and mobile. Bulky PKC protocols will have a disproportionate effect on power consumption.
- Clearly quantum-safe primitives with efficient key generation and small public keys are likely to be preferred for bandwidth-constrained applications. However it will often not be possible to support additional functionality such as forward security or frequent parameter updates.
- All parties in a network want to derive key material in a simple and mutually understood way. It will often not be possible for protocols to be able to negotiate cipher suites or choices of parameters.
- Transmission of public keys can sometimes be eliminated in low bandwidth applications by key pre-distribution mechanisms. To establish a shared key a device only need some relatively small transmit reconciliation data and a pointer to its public key.
- Applications such as V2X require ultra-low latency. This could also be provided by pre-generated and pre-distributed keys or by sending public keys at the earliest moment.

8 Satellite communications

8.1 Requirements

The SAFECrypto case studies document [i.19] discusses requirements for future satellite key management systems. Current systems are typically owned and operated by a single organization and have relatively basic functionality and requirements for symmetric key material. One aim of the SAFECrypto project is to develop much larger and more flexible systems to support the ever growing market for satellite-based services and the increasingly complex requirements for multinational, multi-organization missions and shared infrastructure.

In high-level terms, the document envisions that a typical future satellite control network will comprise of an operational control centre that issues commands to the satellite and receives telemetry data back; one or more ground stations that actually sends the commands and collects the telemetry information and the payload data gathered and sent back by the satellite; end users (consumers of the satellite data); and possibly other auxiliary nodes such as data centres that apply some filtering and processing of the raw satellite data before it is sent on to the end users. The ground-based connections will usually be secured by the usual commercial solutions such as VPNs based on IPSec, TLS, etc. which were discussed in clause 5 above, and so the remainder of this clause will focus just on the key management requirements arising due to communications with the satellites.

The document identifies that cryptographic protection will be *essential* to protect the command and control instructions sent up to the satellite (uplink), and the telemetry channel and the payload data (downlink) sent back to the ground. There may also potentially be requirements for end-to-end security between satellites and end users, or even between satellites in a future "network of space-based entities." Additional requirements are noted for perfect forward security and protection of the satellites against key compromise on the ground. The authors conclude that

- In all cases, it can be assumed that the use of public key cryptography is restricted to authenticated key establishment. This may require public key encryption/decryption and signing/verification to be done on the satellites.
- Due to the longevity of satellites and associated infrastructure, and the difficulty of changing anything after the launch, any public key solution needs to be secure for a long period of time. It is thus an ideal case study for the use of post-quantum cryptographic solutions.

8.2 Constraints

The basic requirement for this project is for a forward-secure authenticated key exchange meeting certain bandwidth constraints and is robust enough to work over a sometimes unreliable channel.

The satellite uplink and telemetry channel are both low data rate, presently around 10-64 Kbit/s and 100 Kbit/s respectively. The downlink is high data rate, perhaps 2Gb/s, however this is mostly taken up with raw payload data. So bandwidth is a precious resource and this will place some limitations on the size available for the public key exchanges. Another implication is that the number of round-trip communications required for the authenticated key establishment is minimized.

The communications links are also characterized by very high latency (up to 240 ms each way for geostationary satellites). This means that for key establishment the time to transmit data will dominate the execution time of the protocol i.e. very high speed is not a requirement for the cryptography.

No special requirements are noted for protection against side channel attacks. These would probably be very difficult to mount in practice since both the satellites and the ground stations would be difficult to access, although timing attacks and to some extent fault based attacks remain as theoretical possibilities.

The physical inaccessibility of the satellite means that any master keys provisioned at launch cannot be replaced. Therefore minimizing the risk of exposure of these keys either directly or through cryptanalysis is deemed to be a critical factor in the selection of the cryptographic schemes.

8.3 Discussion

Although this is a somewhat niche application this scenario nicely illustrates some of the requirements and solution for quantum safe cryptography. In particular the "deploy once" constraint means that long-life cryptography incorporating quantum resistance has been noted a requirement at the outset of the project. There are also some interesting bandwidth constraints. Both of these considerations are similar to some of the IoT use cases.

9 Key Distribution Centres

9.1 Introduction

If Alice and Bob share a secret symmetric key they can securely exchange encrypted messages and authenticate each other. One of the main challenges when using symmetric keys is secure key distribution and management. The security of the entire system rests on the secrecy of the symmetric keys and so these need to be generated and distributed in a secure way.

The most naïve key pre-distribution scheme for a network of N parties would require distributing a total of $N(N\text{-}1)/2$ keys to secure all the pairwise communication links. Each party would have to store N-1 keys and when additional parties join the network the keying materials assigned to each party will need to be updated. While this can be made to work satisfactorily for a small number of parties it does not scale up well and is considered an unsuitable solution for large networks.

An alternative model is for parties on the network to store and share just a single symmetric-key with a trusted server playing the role of either a Key Distribution Centre (KDC) or a Key Translation Centre (KTC) [i.20]. Being a key distribution centre means that Alice can send a request to the KDC stating the need for a shared key with Bob and then the KDC will provide Alice and Bob with such a key. Being a key translation centre means that Alice will generate the key and Bob will be the one that "securely translates" using his shared secret, meaning that that Bob knows the key that Alice is using to communicate with him.

9.2 Examples

9.2.1 Kerberos®

Kerberos® [i.21] is a protocol that has its origins in a distributed authentication service developed at MIT. Basic Kerberos involves a client, a server, and trusted server. Client and Server do not share a secret, while the trusted server shares a secret with each of them. The main goal of the server is to verify the identity of the client. The whole protocol enables mutual authentication between client and server and the establishment of a common key between client and server. Kerberos was first standardized as IETF RFC 1510 [i.22] in 1993 followed by many additional RFCs enhancing or updating it. Today it is widely used for many services.

NOTE: Kerberos® is an example of a suitable product available commercially. This information is given for the convenience of users of the present document and does not constitute an endorsement by ETSI of this product.

9.2.2 ZigBee® Trust Centre

ZigBee® [i.17] defines a protocol for wireless sensor networks. The wireless sensor network is managed by a network coordinator and the security by a trust centre that plays the role of key distribution centre. When a first sensor wishes to communicate with a second one, the first sensor sends a request to the trust centre that then distributes a symmetric key to both of them. This symmetric key is used for mutual authentication and the derivation of a session key.

NOTE: ZigBee® is an example of a suitable product available commercially. This information is given for the convenience of users of the present document and does not constitute an endorsement by ETSI of this product.

9.2.3 Datagram Transport Layer Security (DTLS)

Many IoT protocols rely on DTLS [i.3], the datagram version of TLS which is used to protect the Constrained Application Protocol (CoAP) [i.23]. Secure CoAP requires the use of DTLS with PKC certificates and public-keys, or pre-shared keys [i.24]. Upgrading DTLS to support new quantum safe PKC cipher suites for key agreement and digital signatures is possible, however as noted in clause 7.3, these may not be suitable for some constrained IoT applications, and relying on pre-shared keys (PSK) only is not a very scalable solution. DTLS could be engineered to work with KDC that generate and distribute keys to devices that want to communicate.

A recent interesting proposal is the integration of an ID-based Key Pre-distribution Scheme HIMMO with the DTLS-PSK mode [i.25] so that any pair of devices can agree on a common pairwise key on the fly while inherently verifying their identities or credentials.

9.3 Discussion

Trusted centres such as KDCs and KTCs are widely deployed, however, they have some drawbacks. The first one is that they can lead to lower performance in terms of communications overhead and latency. The second limitation is that they have to be online, which means that availability on demand is a potential issue. The final problem refers to the fact that in principle (but often not in practice) they can monitor all the online communications between parties on their network. The use of an infrastructure of multiple KDCs or KTCs could mitigate issues related to the second and third problems, single points of failure and system privacy.

There have been a number of attempts to overcome these issues. One method is Blom's scheme [i.26], which in its basic version, a trusted party computes a k x k symmetric matrix D over a given finite field. Each party i has associated an identity vector D_i of length k and obtains from the trusted party a secret keying material $U_i = (D_i\ G)^t$. When two parties i and j wish to communicate with each other, they can exchange their identity vectors D_i and D_j and compute a common symmetric-key $K_{ij} = D_j\ U_i = D_i\ U_i = K_{ji}$. This can be done without the intervention of the trusted party overcoming many drawbacks of the above naïve key pre-distribution schemes and online systems based on an online trusted server. However, the problem with Blom's scheme is that if an attacker compromises the secret U_i vectors of k or more parties, the attacker can re-compute the secret matrix D and break the whole system in a very simple way.

After Blom's scheme, Matsumoto and Imai [i.27] generalized the concept of key pre-distribution schemes in 1987 introducing concepts for the verification of information or the usage of multiple trusted servers. In 1992, Blundo et al. [i.28] proposed another scheme with similar properties to Blom's scheme but based on polynomials. The search for efficient key pre-distribution schemes increased in the 00's due to the advent of wireless sensor networks with plenty of key pre-distribution schemes such as randomized ones. In 2007 a scheme based on perturbation polynomials [i.29] was presented aim at creating a collusion resistant and efficient scheme; however it was broken by a couple of years afterwards by Albrecht et al. [i.30]. The HIMMO scheme [i.31] pursued the same goal of achieving a collusion resistant and efficient identity-based key pre-distribution scheme enabling the direct generation of pairwise keys; at the time of writing HIMMO is still under academic assessment and there are few, if any, alternative efficient solutions available.

10 Authentication

10.1 Introduction

This clause will look in a little more detail at some archetypal use cases and proposed solutions for quantum safe authentication, where this is required (see discussion in clause 5.3.3). The overall situation is similar to that of key establishment in that there are a diverse set of requirements and it seems likely that no single quantum safe authentication solution will fit them all.

10.2 Requirements and use cases

10.2.1 Authenticating Internet-based applications

A secure Internet is fundamental in today's connected world; compromises to the authentication used in the communication links between citizen, banks, traders, or healthcare providers would have a huge impact on individuals and on the global economy. Let us consider again the example of TLS discussed in clause 5, which is a typical example of peer to peer authentication between a pair of entities which are assumed to be online. The most straightforward upgrade would be to switch to quantum safe drop in replacements for the ECDSA or RSA signatures in widespread use today. The present document will refer to these types of signatures as *general purpose* signatures. They have a public-private key pair and produce a different signature each time they are called via the use of random values in their generation or padding schemes respectively. Importantly, these do not require any auxiliary key material or periodic resets. To be considered practical, any proposed quantum safe replacement should have key and signature sizes which are not too large and have very efficient sign and verify times to support potentially high volumes of rapid online transactions. See ETSI GR QSC 001 [i.12] for more information on the various options being developed.

10.2.2 Offline file Authentication

The second important example use case is that of authenticating offline data ("data at rest"). Files with critical information such as healthcare, legal or governmental data need to remain authentic for long periods of time. Failure to provide adequate authentication here could lead to the corruption or forgery of data. Other offline data such as software, firmware or anti-virus updates usually do not have such long lifetimes but are also critical to maintaining the functionality and secure operation of offline systems, products and applications.

General purpose quantum-safe replacements for RSA and ECDSA are of course suitable here too. However since updates to offline data are typically much less frequent compared with online applications, and requirements on speed and bandwidth are similarly less stringent, alternative quantum-safe proposals based on Hash tree signature schemes have been proposed as potential alternatives to general purpose signatures for offline applications. Recent proposals for hash tree schemes [i.12] and [i.32] are relatively practical schemes, albeit with much larger key sizes then general purpose signatures. The most important practical difference is that hash tree schemes require some auxiliary data to initialize the tree and can only sign a fixed number of files before needing to be re-seeded. Hash trees are still relatively new and untested in real-world deployments. See [i.14] and the "New Challenges" discussion in [i.32] for some practical concerns over integration of hash tree signatures into applications and libraries and security concerns over "statefulness". At the present time hash tree signatures seem best suited to controlled environments such as root or firmware signing rather than general purpose applications.

10.2.3 Authenticating broadcast communications

The third example is the authentication of broadcast communications with high safety impact. A clear use case is V2X in which cars and transport infrastructure broadcast data that needs to be verified by surrounding entities. The Connected Vehicle security standard [i.33] recommends a signature that is sent up to ten times per second and a public key certificate up to twice per second with a total security overhead of up to 1 100 octets per second on average. At high traffic volumes the total bandwidth available between vehicles is down to around 1 500 octets per second and the total security overhead runs at around 500 octets per second, leaving about 1 000 octets per second for actual transmission data. In practice this means that a suitable quantum safe replacement for the current scheme signature would have to be no longer than twice the current security overhead.

There are currently no good quantum safe drop-in replacement PKC schemes that immediately meet this requirement (even ignoring the performance and latency requirements), however short signature schemes based on HFE [i.12] seem most promising. It seems likely that some combination of short signatures together with key pre-distribution will be required to keep down the security overhead.

10.3 Symmetric solutions

There are also symmetric (non-PKC) methods available for authentication; in particular Message Authentication Codes (MACs) are out of scope for the present document but give efficient authentication schemes; existing standards for MACs include [i.34] and [i.35].

In addition to the practical drawbacks associated with symmetric keys (clause 8), symmetric authentication schemes do not usually give the source-authentication and non-repudiation provided by PKC digital signatures. Hence mixed symmetric and PKC schemes are sometimes integrated into protocols [i.21], relying on a KDC for the verification of identities.

Symmetric-only solutions are usually more suited to one-to-one communication settings while one-to-many uses cases can often only be realized with asymmetric-methods. In a one-to-one setting, both parties can authenticate to each other if they share the same key e.g. through the distribution of PSKs from a KDC. In a one-to-many setting, authentication can be done by means of asymmetric-keys to show source authentication.

10.4 Discussion

The examples given here already serve to demonstrate diversity of requirements and potential solutions: online vs offline modes; network topology (peer-to-peer or broadcast); symmetric or public key. Depending on the operational requirements of the use case under consideration, the implementer will need to decide on a good balance between security and performance needs. Applications as diverse as V2X and TLS enforce strict timing, bandwidth or state limitations on their protocols. In many of these cases, efficiency considerations are the overriding factor and a practical solution with a robust security design may be preferred over a solution with stronger theoretical security but worse performance. General purpose quantum-safe signatures should suit most applications but there will be situations where alternatives such as hash trees and short signature schemes may be more appropriate.

11 Exotic functionality

11.1 Identity-based encryption (IBE)

Public safety communications require secure one-to-one, group and broadcast calls. Such systems need to support several thousand users, who are members of the emergency services needing to communicate securely during an incident, e.g. police officers during a terrorist attack. Symmetric keys need to be established in real time, so that a secure channel is available immediately when requested. They may also need to be frequently re-keyed as participants join and leave a conversation.

Identity-Based Encryption (IBE), where a user's public key is derived from their identity, is appropriate here because it requires fewer exchanges and less infrastructure than the traditional PKI approach to network protocols described in clause 5. It is being considered as a use case for group calls in COTS public safety communications by the SAFECrypto project.

To agree a shared symmetric secret with Bob, a user Alice would traditionally need to obtain Bob's public key, either from a central repository or sent directly by Bob. This public key would need to be authenticated with a digital signature from a trusted authority. This is likely to require several time-consuming exchanges. However in an IBE system, Alice automatically knows an authenticated public key for Bob, because it is his identity in the system, e.g. a telephone number or email address.

In practice, IBE is used for key transport as part of an authenticated key exchange, to establish a shared symmetric key between two or more users. Alice initiates an exchange by encrypting a symmetric key and sending it to Bob. Bob verifies the message via a digital signature that ties the message to Alice's identity, or by sending Alice an IBE message encrypted under her identity. They then communicate using the shared symmetric key.

IBE is deployed in a number of commercial products. All current instantiations are based on elliptic curve cryptography. Encryption and decryption of a 192-bit shared symmetric secret might typically take a few tens of milliseconds, and the corresponding ciphertext might by a few kilobytes long. However elliptic curve cryptography is not quantum-safe, and so a shared symmetric secret encrypted with IBE today is vulnerable to a future adversary with access to a quantum computer.

Several possible IBE schemes have been proposed, e.g. [i.36], [i.37] and [i.38] which might be considered as drop-in replacements in their own right, or as prototypes for a replacement. They all include security analyses, and some reference proof-of-concept implementations. For the most practical of these, encryption and decryption is reported to be several times faster than comparable elliptic curve IBE systems, and encrypted symmetric keys a few times larger.

Much of the discussion in clause 5 on how to manage a quantum-safe transition is relevant to IBE. Current elliptic curve-based IBE schemes could be replaced entirely with quantum-safe primitives, or alternatively with *hybrid* schemes with both classical and quantum-safe security. However, unlike more traditional protocols, there is less scope for transitioning IBE authentication and confidentiality functions at different times, because they are tightly bound into a single primitive. A hybrid approach may therefore be appropriate for IBE in a situation where one of authentication and confidentiality is considered ready for deployment, but the other is not.

Hierarchical IBE (HIBE) is a variant on IBE. It may be suitable for providing scalable secure separation of data similar to ABE and FHE, but is more mature than either of these technologies. Several proposals for quantum-safe IBE already support HIBE functionality [i.38] and [i.39].

In HIBE, a Key Management Server (KMS) is able to delegate limited authority to a sub-KMS. HIBE offers fine-grained security that is easier to scale than IBE but is more practical than current ABE or FHE proposals, as it allows encryption that can be decrypted only by users under a specific delegated KMS. HIBE may also allow more efficient keying. Rather than contacting the central KMS for keying or re-keying, a user can contact their local delegated KMS, which may be more accessible and have fewer users to support.

For example, in a public safety communications system, the central KMS might sit within a national interior ministry, and delegate authority to regional emergency services. This would allow messages to be encrypted so that they can only be decrypted by emergency service responders in particular regions or organizational units. User keying could be done locally, rather than requiring every user to periodically revert to the interior ministry.

11.2 Attribute-based encryption (ABE) and fully homomorphic encryption (FHE)

There are many use cases where security separation is in the access to data:

1) In the public safety communications considered by SAFECrypto [i.19], there may be a requirement to restrict secure channels to participants from a geographical location, organizational unity, or - for classified communications - with an appropriate security clearance.

2) SAFECrypto is also considering secure analysis of municipal data, for research into e.g. public health or economic or social trends. If conducted insecurely, such research could compromise the privacy of individuals or the security of municipal services. Different levels of access may be necessary for e.g. municipality staff, academic and industry researchers, and cloud providers hosting data.

3) The HORIZON 2020 HEAT [i.40] project is considering the processing of data from satellites. In this model, organizations share access to satellites and their infrastructure, which reduces costs. The data gathered by the satellites may however need to be kept confidential, due to commercial or national sensitivities.

4) HEAT is also considering the smart grid for electricity distribution. Aggregate consumption data is critical for grid management, as the use of incorrect data could lead to a blackout. Disaggregated and secure consumption data, on the other hand, is necessary for billing purposes, but these can also reveal personal information such as a consumer's age or health.

In small networks, these requirements may be satisfied by carefully managing which data are encrypted to which individuals, but this approach may be unwieldy and error-prone at scale. Attribute-based encryption (ABE) or fully homomorphic encryption (FHE) may be appropriate instead.

In ABE, a user's public key captures a number of attributes, and data can be encrypted so that only users with a specific set of attributes are able to decrypt it. As with IBE, many historical ABE proposals have been based on elliptic curves, which are not quantum safe. There are several quantum-safe proposals for ABE schemes, e.g. [i.41] and [i.42].

FHE allows computation on encrypted data, and so enables analysis without compromising the confidentiality of individual data points. It is a comparatively recent innovation, and the main proposals such as [i.43] and [i.44] are lattice-based and so considered quantum-safe.

Neither of these approaches is as mature as more traditional crypto functionality, or even IBE, and they use particularly large parameters. For example, FHE ciphertexts in the HEAT project may be several megabytes. Computation is therefore likely to be performed using specialist high-end computing equipment. Whether or not this is a problem will depend on the use case: for example it may be plausible to envision such a system in a secure processing centre for satellite data, but less plausible for an embedded device like a smart meter.

11.3 Discussion

The protocols discussed here promise some very attractive functionality but are still very much the subject of research and development; it is probably too early to give definitive guidance.

12 Conclusions

A considerable amount of work will be required to integrate quantum-safe cryptography into real-world systems. Clause 5 discussed in detail some of the ways that current network protocols could be upgraded to support larger parameter sizes. New cipher suites and other infrastructure will need to be defined to support drop in replacements or hybrid schemes and practical issues such as handling packet fragmentation and time outs will need to become more robust. Consideration also needs to be given for how to support backwards compatibility with legacy systems when the time comes to introduce quantum safe cryptography.

Clause 10 discussed options for upgrading authentication mechanisms. For online, network-based applications the consensus at this time is to continue with classical ECDSA and RSA signatures until the options for quantum safe authentication are better understood and practical quantum safe drop-in replacements are identified. Hash trees signatures seem better suited for offline applications in controlled environments such as certificate or firmware signing rather than general purpose applications. HFE schemes may provide options for applications that require very short signatures.

Next generation technologies will require a greater diversity of security solutions and place more demands on protocols. Both IoT and 5G will put the focus more on efficiency than ever before; not just on speed and bandwidth but also on latency and power consumption. Symmetric schemes will continue to have a place particularly in mobile and IoT networks and these will require key distribution solutions, as discussed in clause 9. Exotic functionality such as IBE, ABE and FHE discussed in clause 11 may also have a part to play, particularly to support public safety applications and to provide privacy for data analytics.

When assessing potential security solutions it will also be important to consider the value and lifetime of the data to be protected, in order to select an appropriate security solution. Some data will be considered relatively low-value and/or ephemeral (e.g. IoT data for controlling household lighting), while other data will be considered high value and/or long-lived (e.g. PKI certificates). A high-level summary of the various examples discussed in this paper is given in annex A.

With so many complex and far reaching decisions to be made it is important that developers do not rush to select and deploy quantum safe solutions too quickly. There is currently no consensus in academia on exactly which quantum safe schemes are the best. Standards bodies such as ETSI and NIST have only recently begun to study in detail which quantum safe primitives to recommend for key establishment and authentication [i.12] and [i.45].

These processes will take several years to complete and one view is that hybrid solutions, where the security solution relies on both the traditional security of classical RSA/ECC *and* a quantum-safe scheme that is not fully scrutinized or standardized may still be better than solely relying on RSA/ECC and waiting years for all the standardization efforts to finish. However a counter view is that attempts to incorporate hybrid cryptography into existing protocol stacks may introduce unnecessary complexity.

Perhaps the best advice for now is for developers to have crypto agility and migration options in mind to make the transition to quantum safe cryptography as straightforward as possible once specific schemes are eventually recommended by standards bodies.

Annex A:
Summary table

Table A.1

Use case	Comment
Online network security protocols (TLS)	Most commercial systems will require both good security and efficiency due to the variety and volume of data carried. Clause 5 noted that integration of quantum safe cryptography into the existing protocol stack would be a major task.
Offline services (S/MIME)	Performance requirements are typically less stringent than for online services. Clause 6 noted that it would be important to upgrade the supporting infrastructure (PKI) as well as the protocol (S/MIME).
IoT	Clause 7 noted that many IoT networks will have severe resource constraints. It may be difficult to support computationally heavy security protocols, however many use cases will have relatively low value, ephemeral data (e.g. household lighting) and hence lower security solutions may be acceptable.
Satellite	Clause 8 noted that some similarities with IoT: the security solution had to be "deploy once" and there were some restrictions on available bandwidth.
KDC / PSK	Can provide good, computationally efficient solution for small low-power networks. PSK does not scale well for large numbers of end nodes and requires some additional infrastructure or key pre-distribution system.
Online authentication	Clause 5 noted that the current consensus is that existing, non quantum-resistant signatures such as RSA and ECDSA should still provide sufficient security for most commercial online applications. It will take academia and standards bodies several more years to identify and approve suitable quantum-safe drop-in replacements for RSA and ECDSA.
Offline authentication	This is seen as an important early use for quantum-safe cryptography, due to the value and lifetime of some of the data to be signed. Clause 10 noted that hash trees are viewed as a good potential solution here, although there are some practical issues around statefulness that need to be considered.
Broadcast authentication	Broadcast Authentication: Some applications such as media broadcasting may require authentication of source embedded in systems such as those for Digital Rights Management (to protect the content). Broadcast Authorization: Some applications such as V2X require high efficiency solutions. The data is important for safety, however it is ephemeral in nature and is intended to give assurance to the receiver that the transmitter is a legitimate V2X device of the type claimed prior to processing the data.
IBE/ABE/FHE	Schemes such as these are likely to become increasingly important for public safety and enterprise applications. Security and efficiency requirements for IBE and ABE are likely to be similar to those for standard online applications; FHE is currently impractical for most practical applications but is the subject of academic research to improve the efficiency.

History

Document history		
V1.1.1	February 2017	Publication