Title*:	**QSC WI#3 - Cryptographic Primitives Suitability**
from **Source***:	CESG
Contact:	Michael Groves
input for **ISG***:	QSC
Contribution **For***:	Decision [ ] / Discussion [x] / Information [ ]
Submission date*:	23 December 2015
Meeting & Allocation:	**QSC#3**

1. Introduction

This Work Item aims to provide an overview of a number of real-world uses cases for the deployment of quantum-safe cryptography. Specifically, we present some examples of where cryptography is widely deployed today and investigate which things may need change to migrate to quantum-safe cryptography.

2. General discussion

Cryptography is already widely used and is rapidly becoming ubiquitous, appearing in everything from widely-used internet and mobile applications to emerging technologies such as IoT. The wide range of applications 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. We present some real-world use cases of where cryptography is deployed today and investigate how things may need change to migrate to quantum-safe cryptography.

In the sections below we give 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 work items.

Disclaimer: This draft survey document should not be treated as an official ETSI endorsement of any products or standards mentioned below. Nor is it our intention to prescribe how protocols defined and maintained by any other standards bodies should evolve. Our intention here is simply to provide some typical example use-cases for discussion.

3. Network security protocols (TLS)

3.1. Overview

Protocols such as IPsec, IKE, TLS, HTTP, SMTP and others are ubiquitous internet or application level protocols used to secure a host of modern communications applications including web browsing, e-mails, VPNs, VoIP, instant messaging, etc. Section 4 of the ETSI whitepaper [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 IETF, W3C or similar groups and it is not in the remit of ETSI QSC ISG to decide how these protocols should evolve. However, given the ubiquitous nature of these protocols, we do need to have some understanding of the compatibility of any ETSI recommended primitives with the wider commercial infrastructure.

The following sections focus on TLS as an important example of a real-world use case. We look at some specific proposals in the literature for ways to upgrade TLS to be quantum secure. The TLS [2], [3] protocol suite provides a cryptographic layer through which network application protocols such as 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 sections below.

3.2. TLS Cryptography

TLS version 1.2, defined in [2] and its intended upgrade, still in draft at [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 RSA, DH, DSA, ECDH and ECDSA and it is precisely these primitives that we want to upgrade to be quantum-safe. Since TLS so widely used, it is here that we will ideally want to deploy our best and most modern primitives to provide secure and efficient quantum-safe replacements for the current PKC protocols.

TLS also makes use of symmetric cryptography e.g. the block cipher AES for data encryption and the SHA hash algorithms 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, we will not discuss these further here but focus on the public-key primitives.

3.3. Quantum resistance

There have been three main approaches suggested so far for possible migration paths for upgrading TLS to incorporate quantum-safe primitives[1].

- 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. The most promising example of this approach is the Ring-LWE proposal [4]. An early proof of concept demonstration [5] integrated a version of this scheme into OpenSSL and compare 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 [6], [7] report greatly increased throughputs of 8x-20x over [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.

- A second proposal is to introduce hybrid schemes which derive an encryption key from some combination of the outputs from a classical key agreement scheme and a separate 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 QSH [8] which proposes combining a standard classical TLS handshake with an NTRUEncrypt key transport. An initial implementation[2] for TLS 1.2 reports a similar reduction in throughput of just 1.2x for the 128-bit security level.

[1] We do not discuss here the large amount of work required to define new cipher suites or upgrade any other infrastructure required to support quantum safe primitives.

[2] Zhenfei will provide a more detailed report separately, and I'll include a summary in the next version of this report. We simply note here that these figures are based on an implementation in WolfSSL using NTRUEncrypt parameter set

- 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 [9] which envisions using session resumption techniques to minimise 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.

Some general comments:

1) Both of the first two proposals [5], [8] are happy to use current (non quantum-safe) digital signatures such as RSA or ECDSA to provide forward security. This is example of the hybrid approach and can be viewed as a pragmatic way forward helping to ease the integration of quantum-safe cryptography into real-world systems.
2) Quantum-safe primitives with very large public keys initially seem unsuitable for widespread deployment using this “drop in” approach as there are often restrictions on packet size, handshake size or other bandwidth issues. However there may be isolated networks that do not have these restrictions.
3) Developers should not underestimate the amount of work required for the integration of quantum-safe cryptography into real-world systems. Even the apparently straightforward “drop in replacement” approach would require new cipher suites and other infrastructure to be defined.
4) Hybrid key exchanges are not always allowed by network protocols (e.g. IKE) or they may not fit into the bandwidth currently allocated for handshakes [10].
5) Many protocols including IPsec and TLS include options to support pre-shared keys, which eliminates the need for public key exchanges completely. However this is impractical for many real-world situations e.g. scalability. Other large systems can be configured to rely on pre-shared keys or work with key distribution centres include the ZigBee wireless mesh network for IoT applications [11], as well as the protocols HIMMO [12] and Kerberos.

4. Internet of Things (DTLS)

4.1. Overview

Oscar will provide the draft text for this section. The paragraph below is a placeholder, extracted from [13].

DTLS is becoming the security standard to secure the IoT since it is required by many Machine to Machine standards such as OneM2M, OMA LWM2M, etc. However, with the advent of quantum computers most of the cipher suites of DTLS will become insecure. Furthermore, already today, it is very frequently discussed that DTLS and its cipher suites are too heavy for many IoT use cases. Thus, there is a need for a DTLS cipher suite that is post-quantum secure, efficient, scalable, and simple to use …

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

For instance, if we consider a home Lighting use case, lighting devices such as light bulbs or light switches are outfitted with a CPU and some communication means forming a wireless network. The system further uses a gateway connected to the Wifi wireless router. The user, Alice, Upon installation of the gateway and downloading an app for her smart phone can easily pair the app and the gateway. Afterwards, Alice just has to

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.

plug the lighting devices that are automatically added to his home network. The user can then use the smart phone app to create lighting settings, control the lights, or create schedules. The user can also enable more advanced features like geo-fencing so that the lights go off or on when she leaves or gets home or enable the remote control of his smart lighting system over the Internet.

We can also consider a Healthcare use case taking place in a hospital. Here, a new patient, Bob, is registered by nurse Alice. Alice retrieves Bob's electronic health record and takes a number of wireless sensors. Alice attaches the sensors, e.g., SpO2, breathing, or blood pressure to Bob’s body and associates them and a health monitoring device to his body sensor network. The health monitoring device will gather Bob's health information as collected from the sensors and securely forward the information to the Healthcare backend in charge of processing medical data and distributing the data to authorized Bob's doctors.

When talking about IoT we can consider many different communication protocols to realize above exemplary use cases. Each of those protocols fulfills different purposes. If it is about getting the smart objects connected, we have to consider protocols like Bluetooth, IEEE 802.15.4, WiFi, 6LoWPAN, ZigBee, LTE, LoRA, Threat, NTCIP, NFC, MQTT, LWM2M, ONEM2M.. . Some of those protocols have a star topology around a gateway that is connected to the Internet or a given core network. Other protocols for a network, typically some form of mesh, required for scalability or lower communication latency. Some of these protocols define only the physical and MAC layers, other protocols focus on the network layer (e.g., 6LoWPAN), other protocols rather focus on the transport and application layers (e.g., LWM2M) and others combine multiple protocols protocols (e.g., ZigBee).

From this description it is clear that there is not a clear winner or solution for IoT and it is not possible to provide an exhaustive description of all of them. Still, many of those protocols have some similarities. Many of them use some type of symmetric key approach for protecting the network or the communications between devices. One example is ZigBee in which a trust center handles a network key shared between the devices. Devices can also share a key with the trust center that can be used for network access or for distributing pairwise key between a pair of devices on demand. There are flavours of ZigBee in which a single key is pre-distributed to all devices and that key then is used to securely distribute a network key when an ad-hoc network is formed. These methods have inherent weaknesses and should only be applied in scenarios with a suitable threat model. In the last years there has been a further move towards public-key cryptography and many of the protocols try to incorporate primitives such as ECDH or ECDSA when feasible as constrained by resource limitations (computational, communication or energy) or operational requirements.

4.2. DTLS Also want: cryptography, quantum resitance cf. sec. 3.2, 3.3.

When talking about the transport layer, we observe that multiple standards rely on DTLS, the datagram version of TLS. This is because it is used to protect the CoAP protocol that was thought to run over UDP. CoAP requires the usage of DTLS with certificates, raw public-keys, pre-shared keys, or no security. However, all existing ciphersuites using certificates and raw public-keys will be broken if a quantum computer is built. Furthermore, cipher-suites based on public-key cryptography are relatively bulky and this can negatively impact to either the underlying network (e.g., features by lossy communication links and low bandwidth) or the business case of the IoT application (for instance, if an smart object communicating over a cellular link only needs to report a few bytes a day of application data but the setup of the secure channel requires several KBs this will have severe consequences for the business case).

Upgrading DTLS to make it quantum-safe can be done in several ways. One approach is to stop using existing public-key primitives and rely on PSK only. However, PSK only will not be a very scalable solution. Another approach is to introduce new cipher suites relying on QS PKC including both key agreement and digital signatures. Drawbacks here relate to the message sizes since communication has to go through relatively constrained and lossy wireless links or the energy consumption to send additional data. DTLS could also be architectured to work with a number of KDCs (only TTPs) so that when two devices are willing to communicate with each other, the KDC distributes a common key. The drawback of this approach refers to the additional communication and energy consumption overhead due to the key establishment requests and the drawback of relying on an only KDC. Another approach would be the integration of an ID-based Key Pre-distribution Scheme, such as HIMMO, with the DTLS-PSK mode so that any pair of devices can agree on a common pairwise key on the fly while inherently verifying their identities or credentials. The usage of an infrastructure of TTPs would also solve issues related to system privacy or a single point of failure. Challenge here is to have further open verification and improvements of ID-based KPSs.

4.3. PSK, Kerberos and other non PKC methods

If Alice and Bob share a symmetric-key, then they can securely exchange encrypted messages and authenticate each other. Once Alice and Bob share a common key, they can apply it in an authentication protocol, derive a session key, or use such a key to send a message. One of the key challenges when using symmetric-keys refers to the key distribution and management approach. The security of the system relies on the secrecy of the symmetric-keys so that these keys need to be distributed in a secure way. If we assume a network with N parties and we assume the pre-distribution of a different pairwise key for each of parties (we call such a system a naïve key pre-distribution scheme), a total of N(N-1)/2 keys would be required to ensure secure pairwise communication links and each party would have to store N-1 keys. This fact poses, on the one hand, challenges regarding storage requirements. On the other hand, if at a later stage additional parties become part of the network, it requires the challenging process of updating the keying materials assigned to each party. As already outlined, each party has to store the keys associated to other parties. If further information about those parties is stored, then the information of those parties can be easily verified. While this works very easily for a reduced number of individuals, it does not scale for a large networks due to the associated storage needs.

Alternatively, a party can store and share just a single symmetric-key with a trusted server playing the role of either a Key Distribution Center or a Key Translation Center. Being a Key distribution center means that Alice will 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 center means that Alice will generate the key and Bob will be the one that "securely translates" with Bob's shared secret so that Bob knows the key that Alice is using to communicate with him. There are many protocols following such an approach:

- Kerberos – is a protocol that has its origins in an MIT's distributed authentication service. The 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 RFC 1510 in 1993 followed by many additional RFCs enhancing or updating it. Today it is widely used for many services.

- ZigBee Trust Center – ZigBee defines a protocol for wireless sensor networks. The wireless sensor network is managed by a network coordinator and the security by a trust center that plays the role of key distribution center. When a first sensor wishes to communication with a second one, the first sensor sends a request to the trust center 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.

Trusted centers are widely deployed. However, they have some drawbacks. The first one is that they have to be online representing a single point of failure. The second limitation is that they lead to lower performance in terms of communication overhead or latency. The final problem refers to the fact that they can monitor all the online communications. There have been a number of attempts to overcome these issues. One method is Blom's scheme, in which in its basic version, a trusted party computes a kxk symmetric matrix D in 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 (1984), Matsumoto and Imai 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. 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 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. The HIMMO scheme pursues the same goal, achieving a collusion resistant and efficient key pre-distribution scheme with properties that can also make it quantum-safe.

5. Deploy once (Satellite communications)

5.1. Satellite key management - background

The SAFEcrypto case studies document [14] discusses requirements for future satellite key management systems. Current systems are typically owned and operated by a single organisation and have relatively basic functionality and requirements for 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-organisation 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 section 3 above, so for the remainder of this section we 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.

5.2. Satellite key management – constraints

The basic requirement for this project is for a forward-secure authenticated key exchange meeting certain bandwidth constraints.

The satellite uplink and telemetry channel are both low data rate, presently around 10-64Kbit/s and 100Kbit/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 should be minimised.

The communications links are also characterised by very high latency (up to 240ms 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 minimising 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.

5.3. Summary

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 IoT use cases.

6. Another example protocol or technology

7. Conclusions

8. Bibliography

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

[2] T. Dierks and Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.2," IETF RFC 5246, 2008.

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

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

[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.

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

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

[8] J. M. Schank, W. Whyte and Z. Zhang, "Quantum-safe hybrid (QSH) ciphersuite for Transport Layer Security (TLS) version 1.3 (draft RFC)," draft-whyte-qsh-tls13-01, 20 September 2015.

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

[10] Z. Zheng, "Quantum safe hybrid handshake," ETIS QSC ISG meeting #2, 2015.

[11] Zigbee, "Zigbee alliance website," www.zigbee.org, 2015.

[12] O. Garcia-Morchon, D. Gomez-Perez, J. Gutierrez, R. Rietman, B. Schoenmakers and L. Tolhuizen, "HIMMO - A lightweight, fully collusion resistant key pre-distribution scheme," in *IACR ePrint 2014/698*, 2014.

[13] O. Garcia-Morchon, R. Rietman, S. Sharma, L. Tolhuizen and J. L. Torre-Arce, "DTLS-HIMMO: Efficiently Securing a Post-Quantum World with a Fully-Collusion Resistant KPS," IACR ePrint 2014/1008, 2014.

[14] A. Waller, A. Byrne, R. Griffin, S. La Porta, B. Ammar and D. Lund, "Case Study Specification and Requirements," SAFEcrypto D.91, 2015.

[15] C. Blanchard, "Security for the third generation (3G) mobile system," *Information Security Technical Report,* vol. 5, no. 3, pp. 55-65, 2000.