I n t e r n a t i o n a l   T e l e c o m m u n i c a t i o n   U n i o n

ITU-T Technical Report

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(01 December 2021)

QSTR-USSD

Low resource requirement, quantum resistant, encryption of USSD messages for use in financial services

Technical Report ITU-T QSTR-USSD

Low resource requirement, quantum resistant, encryption of USSD messages for use in financial services

Summary

According to ITU-T QSTR-SS7-DFS "SS7 vulnerabilities and mitigation measures for digital financial services transactions" unstructured supplementary service data (USSD) is a main medium in which financial fraud is being committed. Due to its clear text form and lack of authentication, fraudsters can gain unlawful access to victim's accounts and transfer money out. The purpose of this Technical Report is to examine new technologies for encryption of USSD in an end-to-end manner and estimate its applicability for integration into existing USSD technology, suggesting new recommendation and signalling requirements for the integration of such technology into the existing reference architecture. This Technical Report focuses both on the core-network end and on the user equipment (UE) end, to recommend the appropriate and most secure location for such encryption technology to be implemented. Another aspect of this Technical Report is to examine the encryption under quantum computing attacks, and to set the standard for quantum resistant encryption in telecom.

Note

This is an informative ITU-T publication. Mandatory provisions, such as those found in ITU-T Recommendations, are outside the scope of this publication. This publication should only be referenced bibliographically in ITU-T Recommendations.

Keywords

Encryption, financial services, quantum, technical report, USSD

Change log

This document contains Version 1 of the ITU-T Technical Report on "Low resource requirement, quantum resistant, encryption of USSD messages for use in financial services" approved at the ITU-T Study Group 11 meeting held virtually, 1-10 December 2021.

Editor:	Assaf Klinger Vaulto Technologies Ltd. Israel	Tel: + Tel: +972502138133 E-mail: assaf.klinger@gmail.com

Table of Contents

Technical Report ITU-T QSTR-USSD

Low resource requirement, quantum resistant, encryption of USSD messages for use in financial services

1 Scope

This Technical Report is a result of ITU-T QSTR-SS7-DFS "SS7 vulnerabilities and mitigation measures for digital financial services transactions". ITU-T QSTR-SS7-DFS states that clear-text USSD is the most common medium of DFS financial transactions in the developing world, where there is no deployment 3G or 4G cellular infrastructure. A fact that leads to large scale financial fraud. This TR surveys the available and upcoming encryption technologies that can mitigate this risk, can be implemented OTT over existing 2G cellular infrastructure and require low computation resources which enable it to be deployed to UICC (uSIM) modules.

2 References

[1] ITU-T QSTR-SS7-DFS, *SS7 vulnerabilities and mitigation measures for digital financial services transactions*.

[2] 3GPP TS 23.090, *Unstructured Supplementary Service Data (USSD)*.

[3] 3GPP TS 22.030, *Man-Machine Interface (MMI) of the User Equipment (UE)*.

[4] 3GPP TS 31.102, *Characteristics of the Universal Subscriber Identity Module (USIM) application*.

[5] Peter W. Shor (1997), *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*.

[6] Grover L.K (1997), *Quantum mechanics helps in searching for a needle in a haystack*, Physical Review Letters 79, 325–328.

[7] Grover L.K (2003), *Terry Rudolph: How significant are the known collision and element distinctness quantum algorithms*? Quantum Information & Computation 4, 201-206. MR 2005c:81037.

[8] Grover L.K. (1996), *A fast quantum mechanical algorithm for database search*, Proceedings, 28th Annual ACM Symposium on the Theory of Computing, (May) p. 212.

[9] PQCRYPTO ICT-645622 final report: https://pqcrypto.eu.org/deliverables/d5.2-final.pdf

[10] "liboqs": an open source C library for quantum-safe cryptographic algorithms, https://github.com/open-quantum-safe/liboqs

[11] "CEX": open-source C library for quantum-safe cryptographic algorithms, https://github.com/Steppenwolfe65/CEX

[12] "libpqcrypto": ibpqcrypto is a new cryptographic software library produced by the PQCRYPTO project, https://ianix.com/pqcrypto/pqcrypto-deployment.html

[13] ISO/IEC 7816, *Identification cards – Integrated circuit cards*.

3 Definitions

3.1 Terms defined elsewhere

None.

3.2 Terms defined in this Technical Report

None.

4 Abbreviations and acronyms

This Technical Report uses the following abbreviations and acronyms:

AES	Advanced Encryption Standard
APDU	Application Protocol Data Unit
API	Application Programming Interface
BIKE	Bit Flipping Key Encapsulation
BTS	Base Transceiver Station
CNSA	Commercial National Security Algorithm
DFS	Digital Financial Services
DH	Diffie-Hellman
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
GSM	Global System for Mobile communications
GTP	GPRS Tunnelling Protocol
HLR & VLR	Home/Visitor Location Register
HQC	Hamming Quasi-Cyclic
ICC	Integrated Circuit Card
IE	Information Element
IMEI	International Mobile Equipment Identity
IMSI & TMSI	International Mobile Subscriber Identity
JCVM	Java Card Virtual Machine
JCRE	Java Card Runtime Environment
KEM	Key Encapsulation Mechanism
LMS	Leighton-Micali Signatures
MAP	Mobile Application Part
MCU	Micro Controller Unit
MMI	Man Machine Interface
MO-SMS	Mobile Originated SMS
MO-USSD	Mobile Originated USSD transaction
MS	Mobile Station
MSC	Mobile Switch Centre
MSISDN	Mobile Station International Subscriber Directory Number
MT-SMS	Mobile Terminated SMS
MT-USSD	Mobile Terminated USSD transaction

NE	Network Element
NFC	Near Field Communication
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OTP	One Time Password
PKI	Public Key Infrastructure
PIN	Personal Identification Number
QSC	Quantum Safe Cryptography (a.k.a Post Quantum Cryptography)
SDCCH	Standalone Dedicated Control Channel
SIDH	Super Singular Diffie-Hellman Algorithm
SIM	Subscriber Identity Module
SMS	Short Messaging Service
SS7	Signalling System No. 7
SoC	System on Chip
STK	Sim Tool Kit
UE	User Equipment
UICC	Universal Integrated Circuit Card
USSD	Unstructured Supplementary Service Data
XMSS	extended Merkle Signature Scheme

5 Introduction

The world of digital financial services (DFS) is based mostly on telecom, since in most countries where DFS is popular, most of the end-users do not have reliable and accessible means to connect to the Internet due to poor 3G/4G deployment. This makes unstructured supplementary service data (USSD) communication channels the dominant communication channels in which the end-user communicates with the DFS provider. Moreover, using signalling system No. 7 (SS7) signalling attacks fraudsters masquerade themselves as the DFS provider to steal money from mobile accounts.

This document intends to survey new, quantum safe cryptography (QSC) encryption technologies that can safeguard USSD which use using quantum computing resistant cryptography and estimate the applicability of such new technology to the DFS use-case.

6 How does USSD work

Unstructured supplementary service data (USSD) is a capability built into the global system for mobile communications (GSM), much like the short message service (SMS). USSD differs from SMS since SMS uses a "store and forward" technique to deliver text messages while USSD information is sent directly from a sender's mobile handset to an application platform handling the USSD service. The USSD service can be located either in the sender's mobile network or in another connected network.

Another key difference from SMS is that USSD initiates a real-time "session" between the user equipment (UE) and the USSD application platform when the service is invoked, allowing data to be sent back and forth between the mobile user and the USSD application platform until the USSD service is completed. A USSD session can be invoked by either the UE or the USSD platform [2].

6.1 Network architecture for USSD signalling

According to [2], the network architecture of USSD services is described as shown in Figure 1.

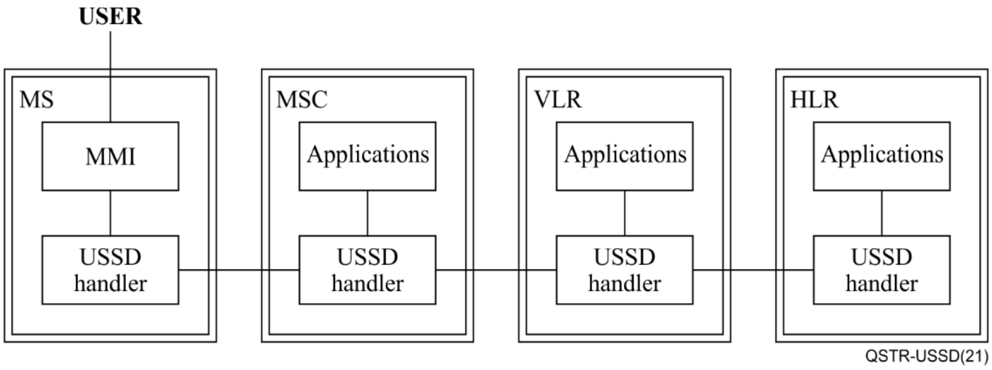

Figure 1 – USSD network architecture

The USSD session can be initiated by any one of the network elements (NEs) depicted in Figure 1, however the USSD initiation process is categorized either as a mobile originated USSD transaction (MO-USSD), if the session is originated by the MS, or as a mobile terminated USSD transaction (MT-USSD), if the session is initiated by any of the core NEs.

6.2 Signalling flow example of a MO-USSD session

In Figure 2 an example of a "balance query and top up" USSD session signalling flow is described. For more signalling flows of MO-USSD and MT-USSD please refer to [2].

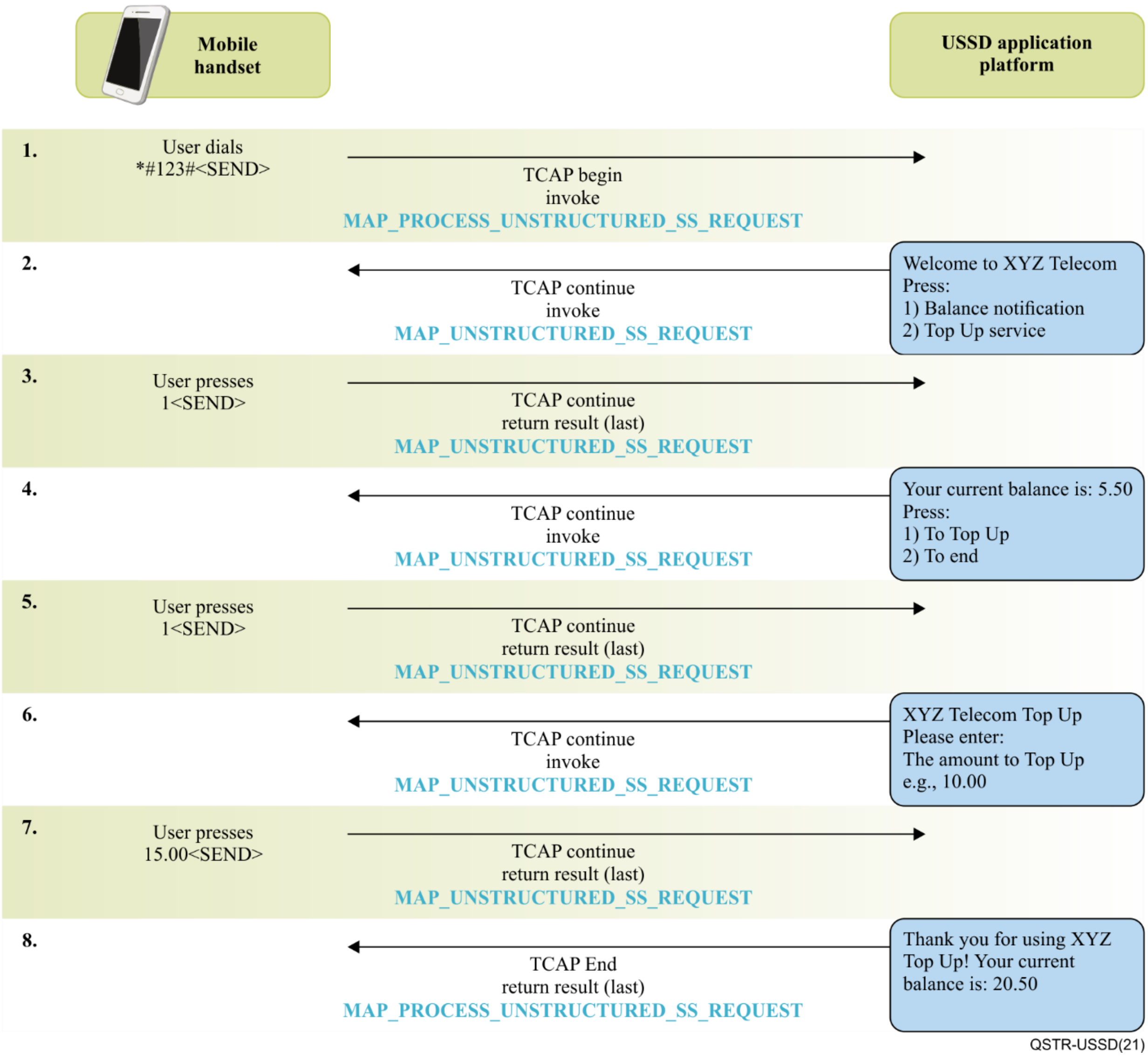

Figure 2 – Example of MO-USSD signalling flow

The information elements (IEs) and data in the USSD session is exchanged in clear text, as can be seen in the packet capture shown in Figure 3, this can be seen by inspecting the 'USSD string' field in the packet.

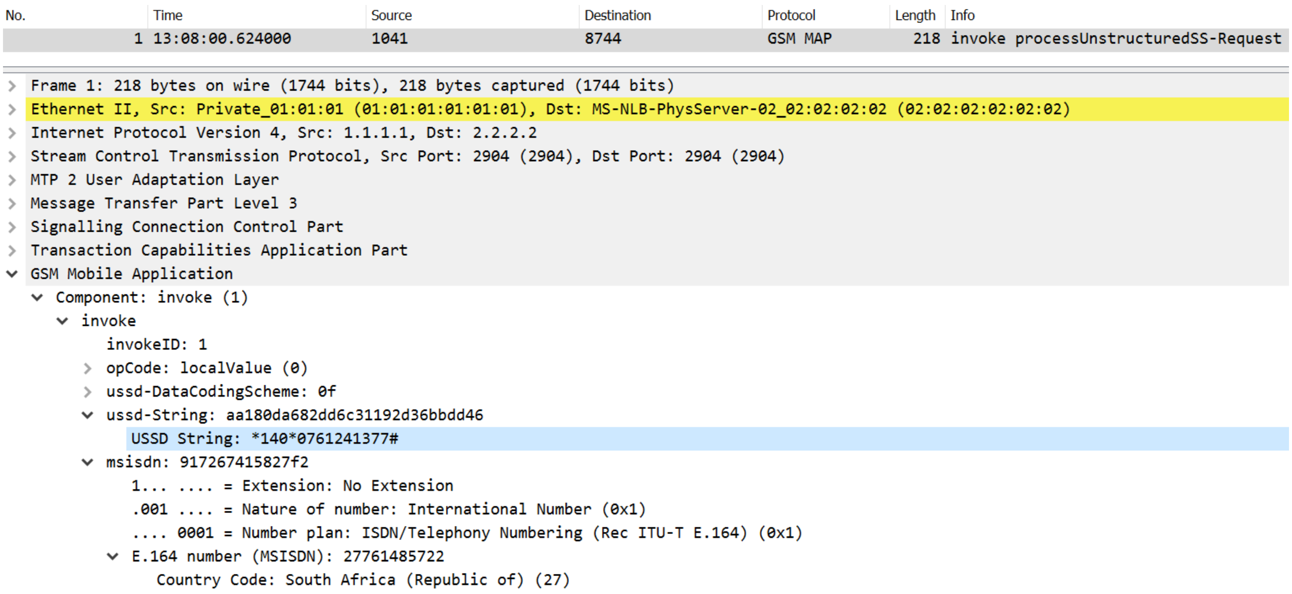

Figure 3 – Example of USSD signalling packet

6.3 USSD data rate

USSD is transmitted over a standalone dedicated control channel (SDCCH) which can hold a bandwidth of 0.8 kbit/s in 2G.

7 Examples of exploiting USSD vulnerabilities on to commit DFS fraud

7.1 Account takeover

In this example, a fraudster uses USSD to takeover an account that does not belong to him. To perform this attack, the fraudster first needs to spoof his victim's phone number and dial the USSD code (this can be done by over the air interception). Once the fraudster initiates the USSD session with the digital financial services (DFS) provider spoofing the victim's phone number they can change the personal identification number (PIN) code and add another phone number to the account. Once done, the fraudster performs another USSD session, this time with the new phone number they added and use the new PIN to login to the account and transfer the money out.

7.2 Social engineering of sensitive credentials using USSD

Unstructured supplementary service data (USSD) is used for, online banking and other financially sensitive applications. Due to the high level of assumed trust by the users (when receiving USSD messages), the simplest attack to execute and scale an attack is using USSD to send a fraudulent message to the user spoofing the identity of the financial service provider, luring the user to divulge sensitive information such as account number and PIN code. For example, to phish these credentials, the attacker sends a phishing USSD message as shown in Figure 4.

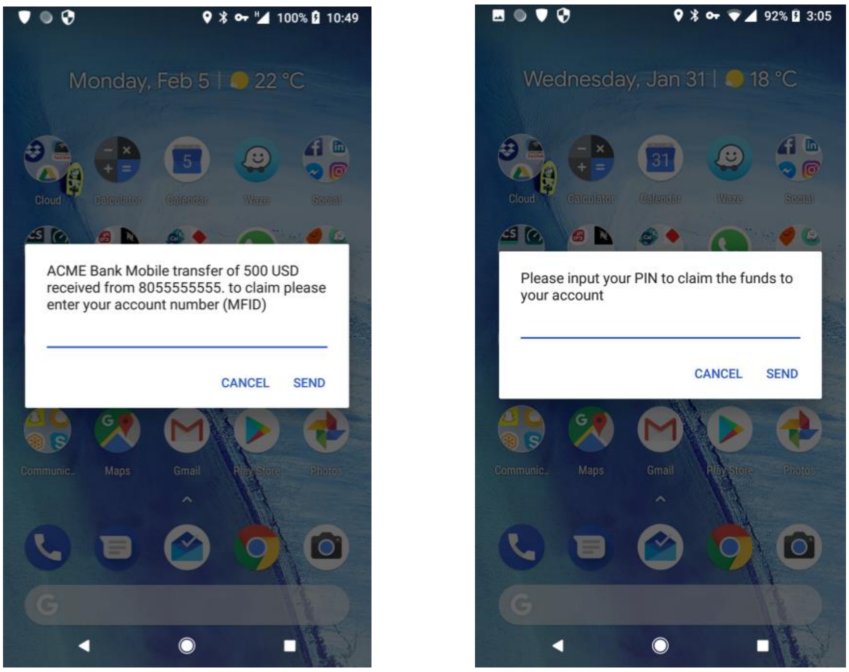

Figure 4 – Using USSD to socially engineer the user

Since there is no identification in the USSD message, and the user is used to having these messages from the network, trust is achieved, and the user divulges their account number and PIN. From there on, the attacker logs into the account and transfers the funds out.

8 Quantum safe cryptography (QSC)

Quantum safe cryptography refers to cryptographic algorithms that are thought to be secure against an attack by a quantum computer. The problem with currently popular asymmetric cryptographic algorithms, is that their security relies on one of three hard mathematical problems: the integer factorization problem, the discrete logarithm problem, or the elliptic-curve discrete logarithm problem. All these problems can be easily solved by a sufficiently powerful quantum computer running Shor's algorithm [5]. Even though current, publicly known, experimental quantum computers lack processing power to break any real cryptographic algorithm, many cryptographers are designing new algorithms to prepare for a time when quantum computing becomes a threat.

8.1 Approaches to quantum safe cryptography

There are two categorical approaches researchers take when developing quantum resistant cryptography.

The first is developing new algorithms and trap door functions that have inherent resiliency to the computation advantages of quantum computers for asymmetric ciphers.

The second is to double the key-space of current symmetric algorithms, since Grover's algorithm [6] and [7] proved that quantum computers reduce the primage resistance of popular hash functions using n bits input to $2^{\frac{n}{2}}$ (and it has the same impact on the key search [8]), thus doubling the key size can effectively enable block and other symmetric ciphers to retain their current security level, provided other security parameters are not affected and the construction adapts to the key size increment.

In this clause we will survey both approaches and try to compare the different solutions via a common criterion which is the applicability to USSD encryption.

8.2 New algorithms for post-quantum cryptography

There are five families of new algorithms for post-quantum cryptography:

1) **Lattice-based cryptography**: constructions of cryptographic primitives that involve lattices, either in the construction itself or in the security proof. Many lattice-based constructions are considered to be secure under the assumption that certain well-studied computational lattice problems cannot be solved efficiently by both classical and quantum computers.

2) **Multivariate cryptography**: construction of asymmetric cryptographic primitives based on multivariate polynomials over a finite field F. In certain cases, those polynomials could be defined over both a ground and an extension field, in which case the polynomials have the degree of two. It is commonly admitted that multivariate cryptography turned out to be more successful as an approach to build signature schemes primarily because multivariate schemes provide the shortest signature among post-quantum algorithms, however this family of algorithms produce large sized keys which may negate the advantage of the small signatures.

3) **Hash-based cryptography**: constructions of cryptographic primitives based on the security of hash functions. So far, hash-based cryptography is limited to digital signatures schemes such as the Merkle signature scheme. Hash-based signature schemes combine a one-time signature scheme with a Merkle tree structure. Since a one-time signature scheme key can only sign a single message securely, it is practical to combine many such keys within a single, larger structure. A Merkle tree structure is used to this end. In this hierarchical data structure, a hash function and concatenation are used repeatedly to compute tree nodes. In 10/2020, the US National Institute of Standards and Technology (NIST) published a recommendation for stateful hash-based signature schemes [b-NIST SP 800-208] based on the extended Merkle signature scheme (XMSS) and Leighton-Micali signatures (LMS).

4) **Code-based cryptography**: cryptographic systems which rely on error-correcting codes, such as the McEliece and Niederreiter encryption algorithms and the related Courtois, Finiasz and Sendrier signature scheme. The Post Quantum Cryptography Study Group sponsored by the European Commission has recommended the McEliece public key encryption system as a candidate for long term protection against attacks by quantum computers. Classic McEliece is also a finalist (3rd round) in NIST-PQC and two other code based key encapsulation mechanisms (KEMs) have also made it to the third round of NIST-PQC as alternates, namely bit flipping key encapsulation (BIKE) and Hamming quasi-cyclic (HQC).

5) **Super-singular elliptic curve isogeny cryptography**: cryptographic system that relies on the properties of super-singular elliptic curves and super-singular isogeny graphs to create a Diffie-Hellman replacement with forward secrecy. This cryptographic system uses the well-studied mathematics of super-singular elliptic curves to create a Diffie-Hellman like key exchange that can serve as a straightforward quantum computing resistant replacement for the Diffie-Hellman and elliptic curve Diffie–Hellman key exchange methods.

The key issue with the new algorithms is that **the primary research is not yet complete and until today no applicable implementation of these algorithms exists**, which provides full coverage for application security, i.e., encryption, digital signature, and trap-door functions. Several national standardization bodies conduct independent programs for post-quantum cryptography, for example NIST started such a program in 2016, the program is at its 3rd stage, and currently there are only 3-5 candidates left in each category (key exchange, digital signature and encryption/key establishment) draft standards are expected to emerge in 2022, more information can be found in [b-NIST-PQC]. Another program is the EU commission's PQCRYPTO [9] program that ran from 2015 to 2018, which was not able to reach any kind of standardization of new algorithms and states in its final report *"It is very clear, that the road to standardization of post-quantum cryptography is* *a long one. Therefore, project partners interested in standardization of post-quantum cryptography need to continue their efforts beyond the formal termination of PQCRYPTO"*

Some of the current active implementation projects for post-quantum cryptography are "liboqs" [10] and CEX [11].

8.3 Symmetric algorithms

Unlike the new algorithms described in the previous clause, quantum-resilience for symmetric encryption can be achieved by extending the key length of traditional encryption, and hashing algorithms. Table 1 contains the available symmetric encryption minimum recommendation for the post-quantum era:

Table 1 – Available symmetric encryption for the post-quantum era

Mechanism	Algorithm	Recommended by
Hash function	SHA-2	[b-NIST IR 8105], [b-ITU-T X.1197]
Confidentiality	AES256	[b-NIST IR 8105], [b-ITU-T X.1197]

8.4 Asymmetric algorithms

Asymmetric algorithms based on the difficulty of factoring or solving discrete logarithms are considered to be quantum-broken, thus they need to be replaced by new algorithms. Table 2 contains the NIST-PQC 3rd round candidates for asymmetric cipher suites for the post-quantum era:

Table 2 – Candidates for asymmetric cipher suites for the post-quantum era

Mechanism	Algorithm	Recommended by
Digital signatures	Crystals (Dilithium), Falcon, Rainbow	Still under review, none are recommended thus far
Public key encryption / KEMs	Classic McEliece, CRYSTALS-KYBER, NTRU & SABER	Still under review, none are recommended thus far

8.5 Available post-quantum software packages

Please note, that with regards to all packages listed below, they implement the specification of the algorithms as they were known at the time these packages were developed and posted publicly. **This document does not intend to present any of these implementations as standard or complying to standard, they are available as-is.** In addition all the libraries listed below are not production ready, and its contributors specify that their library is meant to help with research and prototyping and recommended not to be used in production environments.

8.5.1 CEX

AES-based AHX and RHX in CEX [11] provide a host of symmetric and asymmetric quantum-safe algorithms. It is backwards-compatible with AES CPU optimizations/co-processors.

This library is being built in two stages; the symmetric cryptography, which consists of ciphers, hash functions, MACs, RNGs, TRNGs, etc, preliminary work has been completed as of version v1.0. That work is still evolving however, as improvements and additions to the symmetric cryptography will continue throughout the library's lifetime. The second half is the addition of asymmetric cryptography, with a strong focus on post-quantum security. This work is well under way, and this release contains the NTRU (NTRU Prime), RingLWE (New Hope), ModuleLWE

(Kyber) and McEliece (Niederreiter) asymmetric ciphers, as well as the Dilithium, XMSS, Rainbow, and SPHINCS+ signature schemes.

8.5.2 liboqs

Liboqs [10] provides many NIST PQC candidates, including KEMs such as BIKE, McElisse, NTRU, SABER and others, and digital signature algorithms (DSAs) such as CRYSTALS-Dilitium, Falcon, Picnic Rainbow and others. Liboqs contains more PQC algorithms than CEX, but contains no symmetric ciphers that are designed to be quantum-resistant.

8.5.3 libpqcrypto

libpqcrypto [12] is a new cryptographic software library produced by the PQCRYPTO project, that includes software for 77 cryptographic systems (50 signature systems and 27 encryption systems) from 19 of the 22 PQCRYPTO submissions

9 The uSIM as a computation platform for post-quantum crypto

9.1 SIM Card – background

- ISO/IEC 7816 Smart card i.e. universal integrated circuit card (UICC)
- Same as the UICC banking cards, digital id / passport, or any other card with a "chip" – further reading: [b-Smart-Card]
- The subscriber identity module (SIM) is an application of a UICC
- The UICC consists of a CPU, RAM, E^2PROM and I/O circuits
- Modern UICC cards also have wireless near field communication (NFC) interface
- UICCs run Java card runtime environment (JCRE) operating system with applications (named "applets") running as Java card virtual machines (JCVMs) on the JCRE – further reading [b-Java-Card]

9.2 USIM software and hardware description

Table 3 describes USIM software and hardware.

Table 3 – USIM software and hardware

Application	2G – SIM	3G – SIM+uSIM	4G – SIM+uSIM+iSIM
Smart card type	ICC	UICC	UICC/eUICC
CPU	8 bit MCU	16 bit MCU	32 bit SoC
Storage (E^2PROM)	Up to 32 Kbyte	Up to 128 KByte	Up to 256 Kbyte
Interface	Electrical	Electrical	Electrical/NFC
# of identities	1	2	multiple
Burning	Physical	Physical + OTA	Physical + OTA
Cryptography	A5/1, A5/2	A5/1, A5/2, A5/3, Kasumi, Milenage	A5/1, A5/2, A5/3, Kasumi, Milenage, AES128, PKI

In order to support QSC, the (U)SIM shall provide storage for a 256-bit root key, as an enabler for 256-bit, lightweight post-quantum symmetric cryptography algorithms. The 3G/4G cards are able to support such a key.

9.3 USIM file system and applet structure

9.3.1 ICC card file system example

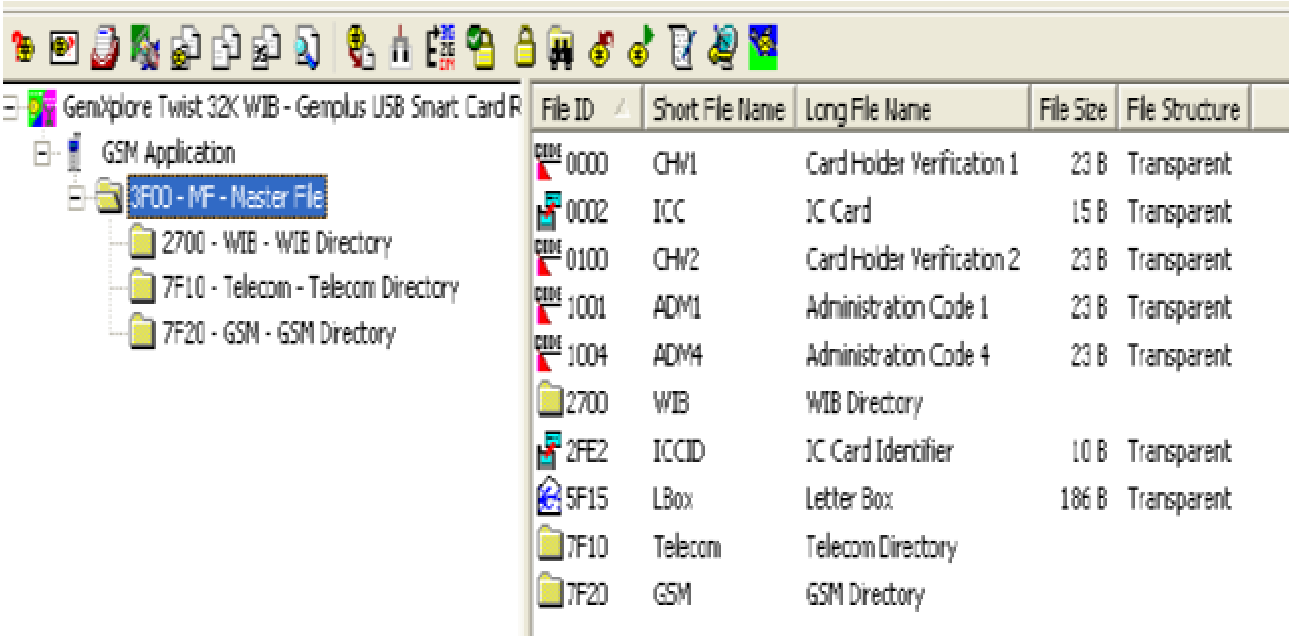

Figure 5 – ICC card file system

Figure 5 shows an example of an integrated circuit card (ICC) card file system.

9.3.2 ICC card telecom data file

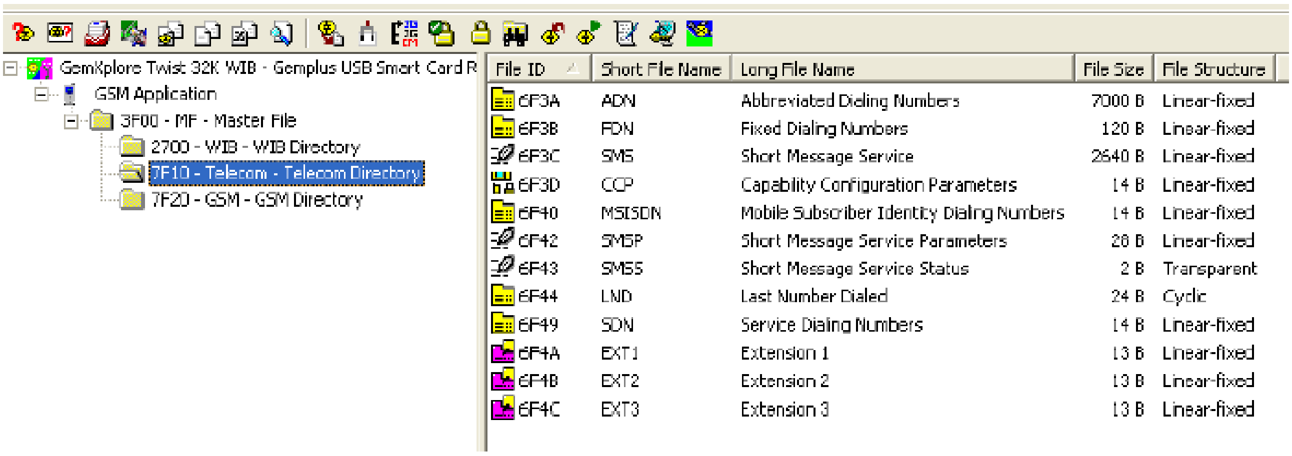

Figure 6 – ICC card telecom data file

Figure 6 shows an example of an ICC card telecom data file.

9.3.3 ICC card GSM data file

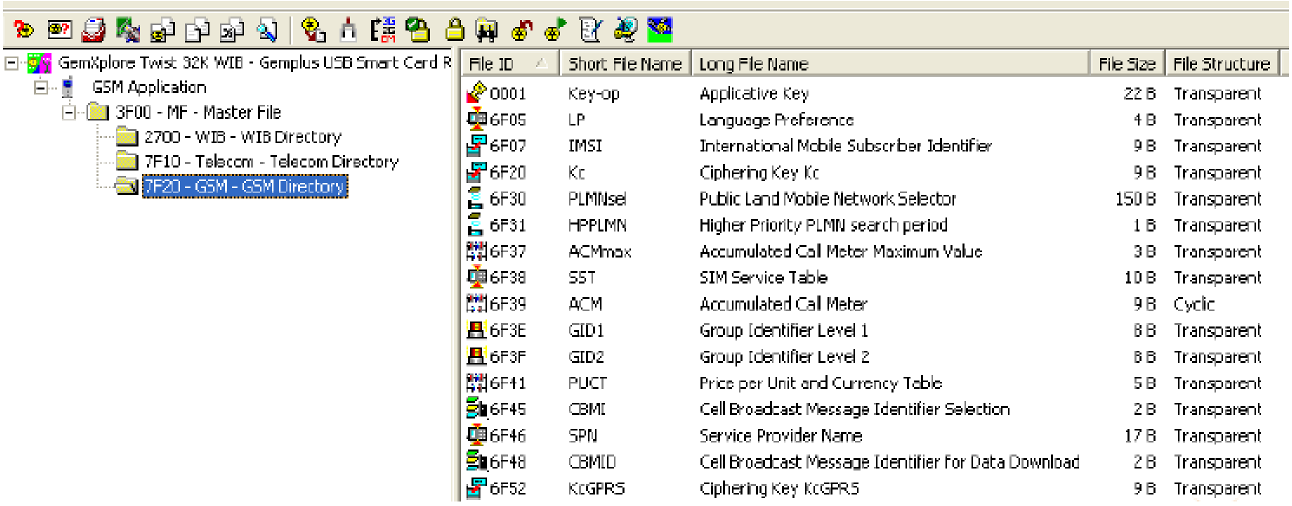

Figure 7 – ICC card GSM data file

Figure 7 shows an example of an ICC card GSM data file.

9.3.4 SIM card UICC example

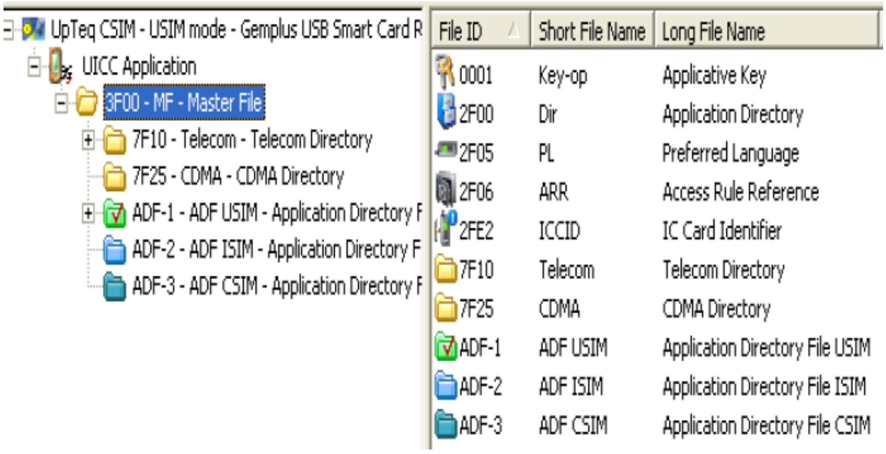

Figure 8 – SIM card UICC

Figure 8 shows an example of a SIM card UICC.

9.4 USIM resources applicability to post-quantum cryptography algorithms

The USIM engine is the JCRE which runs JCVMs (Applets), the host (the UE) communicates with each applet using a simple application programming interface (API) based on files. The Applets are written in Java and the API messages between the host and the applet are called application protocol data units (APDUs). Figure 9 details the interface of the host to the applet.

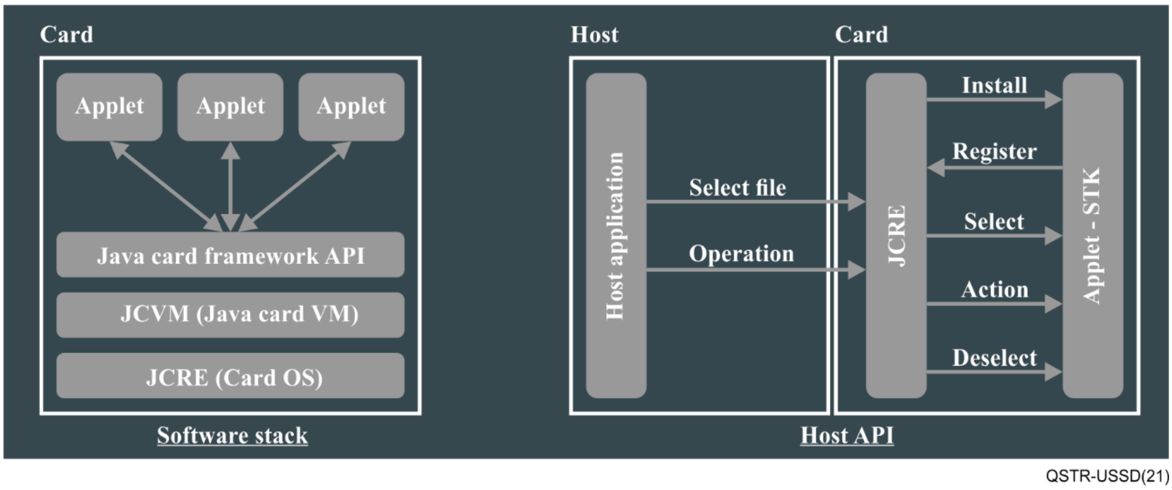

Figure 9 – Interface of the host to the applet

In order to support QSC, a (U)SIM applet needs to be written to the card, hardware acceleration for at least advanced encryption standard (AES), including AES-256, and preferably also for HKDF expand using SHA256 will contribute greatly to the performance of the card but it is not mandatory as the UEs themselves are powerful enough to support it.

10 Applicability matrix between UICC platform and post-quantum crypto

At this stage, since PQC for asymmetric cryptography is not yet finalized nor standardized, this report will focus on the applicability of application of post-quantum symmetric cyphers, also since there is benchmark information on the execution of these ciphers with MCUs and CPUs found in UICCs.

Since different UICC vendors use different MCU/CPU we will use a reference model and architecture MCU and CPU to test the applicability:

MCU reference model and architecture – ATmega 8 bit AVR running at 16 MHz clock. Benchmarking data taken from [b-NIST-LWC]

CPU reference model and architecture – ARM Cortex M0+ (32 bit) running at 48 MHz clock. Benchmarking data taken from [b-Crypto-Benchmark]

Action	MCU performance	CPU performance
AES256-GCM Encrypt	6 milliseconds → 40 kbps	0.2 milliseconds → 1.1 Mbps
AES256-GCM Decrypt	6 milliseconds → 50 kbps	0.2 milliseconds → 1.1 Mbps
SHA-2 (256 bit)	5 milliseconds → 200 Hash/s	0.4 milliseconds → 2261 Hash/s

As can be seen from the table above, **quantum safe symmetric cyphers can run on simple UICCs (even old ones holding only a SIM app) with ample performance to secure USSD transactions**, which require only 0.8 kbps, using lightweight implementations of symmetric ciphers.

To provide the broadest compatibility, new (U)SIMs should be deployed for legacy devices as mentioned in the above clauses 9.1 and 9.2. Those may enable post-quantum secure mobile payment on such devices.

Bibliography

[b-ITU-T X.1197] Recommendation ITU-T X.1197 (2012), *Guidelines on criteria for selecting cryptographic algorithms for IPTV service and content protection*.

[b-Crypto-Benchmark] https://www.bearssl.org/speed.html
https://www.wolfssl.com/docs/benchmarks/

[b-Java-Card] *JavaCardOS Wikipedia*.
https://www.javacardos.com/wiki/start

[b-NIST IR 8105] NISTIR 8105 (2016), *Report on Post-Quantum Cryptography*.

[b-NIST-LWC] https://www.wolfssl.com/docs/benchmarks/
https://core.ac.uk/download/pdf/186473296.pdf

[b-NIST-PQC] *Post-Quantum Cryptography*.
https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions

[b-NIST SP 800-208] *Recommendation for Stateful Hash-Based Signature Schemes*.
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-208.pdf

[b-Smart-Card] *Smart card*.
https://www.wikiwand.com/en/Smart_card